MuCAR: A Greedy Multi-flow-based Coding-Aware Routing in Wireless Networks

Jing Chen*, Quan Yuan†, Ruiying Du*, and Jie Wu‡
*State Key Laboratory of Software Engineering, Computer School, Wuhan University, Wuhan, China
†Department of Math and Computer Science, University of Texas-Permian Basin, Odessa, TX, USA
‡Department of Computer and Information Sciences, Temple University, Philadelphia, PA, USA
Email: *{chenjing, duruiying}@whu.edu.cn, †yuan_q@utpb.edu, ‡jiewu@temple.edu,

***Abstract*—It has been proved that network coding can optimize routing in wireless networks. Thus, in deterministic routing, coding is considered as an important factor for route selection. While the existing deterministic routing solutions detect paths with coding opportunities based on the two-flow coding, little attention has been drawn to the multi-flow situation. Coding multiple flows *directly*, however, can improve the coding benefit when multiple flows intersect at coding nodes. In this paper, we analyze the challenges of the multi-flow coding, and propose a Greedy Multi-flow-based Coding-aware Routing (MuCAR) protocol in wireless networks. The main idea is to define the decoding policy and the coding condition in the multi-flow environment, and code the multiple intersecting flows in a greedy way. Meanwhile, we discuss the interference issue in the multi-flow coding and its solution. We show that MuCAR can induce competitive performance in terms of increased coding benefit and decreased delay, which is verified by extensive simulations.**

***Index Terms*—Network coding, multi-flow, routing.**

I. Introduction

Network coding [1] leverages the inherent broadcast characteristic of wireless channels, to augment a network's capacity. Instead of solely forwarding packets, intermediate nodes encode the packets from different flows into one for transmission, known as *inter-flow* coding. Those packets would be further recovered at destinations. Recently, such a technique has been utilized in routing protocols for wireless networks to improve network throughput [2]–[4].

One of the applications of network coding is in *deterministic routing*, where the route between a given pair of nodes is determined before packet delivery. Specifically, coding opportunities are evaluated on candidate routes, and routes with more coding opportunities are picked by source nodes for data transmission, known as *deterministic coding-aware routing*. Even though extra information is required to predetermine the next hop, deterministic coding-aware routing schemes have the advantage of controllable performance. Based on whether that extra information is collected periodically, those schemes can be further classified into two categories: *proactive* and *reactive*. Proactive protocols [5]–[7] periodically monitor link connectivity, neighbors' information, and flow rates et. al to estimate the availability and the coding opportunity of a path for route selection. In contrast, in the reactive protocols [4], [8], [9], routes are established only upon requests, without periodically collecting information. Existing reactive protocols

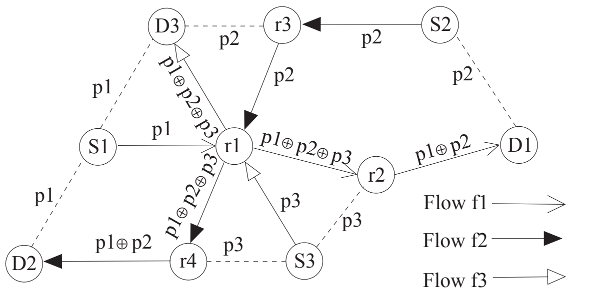


Fig. 1. Decoding at intermediate nodes example in a multi-flow network

improve COPE [10], the first practical network coding system for wireless networks, by detecting network coding opportunities over a multi-hop region, instead of the two-hop region in COPE. Since a wider region is covered for discovering coding opportunities, they exhibit an appealing routing performance.

However, when detecting paths with coding opportunities, most of the existing deterministic reactive routing protocols [4], [8] focus on two-flow coding, while multi-flow coding is rarely discussed. Such insufficient discussion may impair the *coding benefit*, which depends on not only the number of coding opportunities but also the number of the coding flows. For example, in Figure 1, there are initially two flows, f_1 ($S_1 \to r_1 \to r_2 \to D_1$), f_2 ($S_2 \to r_3 \to r_1 \to r_4 \to D_2$), which intersect at node r_1. Based on the two-flow coding methods, packet p_1 from flow f_1, and p_2 from flow f_2 can get coded at node r_1 as $p_1 \oplus p_2$. However, if there is a new flow f_3 ($S_3 \to r_1 \to D_3$) whose packet is p_3, intersecting with two other flows at node r_1, the existing two-flow coding methods cannot *directly* code those three flows together. We observe that by allowing node r_1 to encode packets p_1, p_2, p_3 into $p_1 \oplus p_2 \oplus p_3$ directly, it can improve the coding benefit. Also, since r_2 overhears p_3 from S_3, and D_1 overhears p_2 from S_2, p_1 gets successfully recovered at D_1. Similarly, nodes D_2 and D_3 can obtain their interested native packets, respectively.

In this paper, we propose a Greedy Multi-flow-based Coding-Aware Routing (MuCAR) protocol to improve coding benefit in deterministic reactive routing, where multiple flows are *directly* encoded in a greedy way when they satisfy our coding condition, and the encoded packets are decoded through the collaboration of multiple decoding nodes.

Our work introduces several key challenges to be solved. First of all, a coding opportunity is identified by determining whether the encoded packet can be successfully decoded. Decoding in the multi-flow situation involves the cooperation of multiple decoding nodes, while in the two-flow coding, decoding is conducted at a single node. In other words, a novel decoding policy is required to define the coding condition in the multi-flow environment. Secondly, multi-flow coding may change nodes' forwarding behaviors, which can crash the sufficiency of the existing coding condition in the two-flow coding, defined as *multi-flow interference* in this paper. Thus, simply extending the coding condition in the two-flow to the multi-flow situation does not work. Thirdly, the multi-flow coding should not decrease coding opportunities compared with the two-flow coding, especially considering that the coding condition in the multi-flow situation is more strict. Therefore, MuCAR has to be backward compatible to the two-flow coding in the worst case, which indicates that the number of coding opportunities in the two-flow coding is its lower limit. Finally, multi-flow indicates flow rate difference. As a practical coding system, both of the real-time and adaptive requirements must be considered simultaneously.

Our main contributions are: a) a greedy decoding policy to regulate when and how to decode packets cooperatively; b) a coding condition to identify coding nodes in the multi-flow environment; c) a scheme to sense and avoid the *multi-flow interference* in the process of route discovery; d) a greedy aggregation mechanism to maximally code the qualified flows together; e) a greedy encoding and decoding algorithm to reduce the transmission delay.

The reminder of the paper is organized as below. We review the related work in Section II. In Section III, we discuss the decoding policy and the coding condition in MuCAR, and the detailed implementation is proposed in Section IV. Section V evaluates the performance. The paper is concluded in Section VI.

II. Related Work

Network coding highlights a novel direction in routing to improve network throughput. This section reviews the related work that inspired our work in the literature.

The main issue addressed in coding-aware routing schemes is how to obtain more coding benefit in routing, which means saving more transmissions. Relevant existing work can be divided into two main categories: *opportunistic routing*, and *deterministic routing*. In the former category, each node rebroadcasts packets to its neighbors with a given forwarding probability, where network coding is employed to improve transmission efficiency. Khreishah et al. [11] designed a distributed opportunistic routing algorithm based on coding, by formulating the problem with arbitrary channel conditions as a convex optimization problem, and presenting an optimal back-pressure algorithm on that. CodePipe [12] is a reliable multicast protocol proposed in lossy wireless networks. By employing an LP-based opportunistic routing structure, opportunistic feeding, fast batch moving and inter-batch coding, the work offered improvement in throughput, energy-efficiency and fairness.

On the other hand, deterministic routing predetermines particular nodes that forward packets [4] with some extra information. Considering whether that extra information is periodically collected, it can be further categorized into two subcategories: *proactive* and *reactive*. Proactive protocols periodically monitor peer connectivity to ensure the availability of a path. Sengupta [5] et al. proposed CA-PATH-CODE, a XOR-based coding-aware routing, based on the COPE [10] approach, which leveraged the coding opportunities in a two-hop range. HyCare in [6] exploited the Expected Time of Overall Transmission (ETOX) as the link-state information, to find possible network coding opportunities in routing. [7] presented a Link State MultiPath (LSMP) protocol that utilized network coding and link-state shortest path routing. Such proactive schemes usually consume extra resources to periodically collect some information, such as neighbors and flow rates, to estimate coding benefit. In contrast, reactive protocols establish paths only upon request, and therefore they usually require fewer resources. Researchers in [4] presented Distributed Coding-Aware Routing, which is a reactive XOR-ed routing scheme. Generalized coding conditions (GCCs) were defined to discover paths with potential coding opportunities, which eliminated the two-hop coding limitation in COPE. Jing Chen et al. [8] proposed a Connected Dominating Set (CDS)-based and Flow-oriented Coding-aware Routing (CFCR) scheme. The scheme selected the appropriate coding nodes from the connected dominating set to discover coding opportunities. However, most of the existing reactive protocols only consider two-flow coding, without discussing the multi-flow coding sufficiently, which may degrade coding benefit. Bin Guo et al. [9] presented a general discussion on the coding condition. However, they did not consider the multi-flow interference and other implementation details. For that, we propose a Greedy Multi-flow-based Coding-Aware Routing (MuCAR), which systematically investigates the decoding policy, the coding condition, and the encoding and decoding algorithm in the multi-flow situation, to further improve the performance.

III. MuCAR Components

A. System Model

The system model used in this paper is that in a multi-hop wireless network; a group of nodes are involved in moving data packets from the source nodes to the destination nodes. To reduce the transmission number, a coding node generates and broadcasts the newly coded packets, which are the XOR combinations [10] of the earlier received native packets $p_1, p_2, ..., p_n$ from multiple flows $f_1, f_2, ..., f_n$, when they are passing through that node. Note that the rates of flows may vary in the network. In other words, we focus on the *inter-flow* coding in this paper, as opposed to the *intraflow* coding [13]. Intermediate nodes can decode received coded packets cooperatively if sufficient information is acquired through overhearing. Once the intended destination receives the native packet extracted from the coded packet, the message delivery is finished, as shown in Figure 1. Also, due to the dynamic nature of the wireless networks, the quality of a link between any two nodes may change unpredictably. However, to simplify analysis, we assume that links are asymmetric between different nodes.

B. Greedy Decoding Policy

Previous works, such as DCAR [4], CFCR [8], have similar limitations in utilizing network coding for routing. First, they only consider two intersecting flows, but evade the mutual interference among multiple flows. Such limitations may impair coding benefit in the network. Besides, they focus on finding one node for decoding to define coding conditions, which is not practical in the multi-flow case. For example, as we mentioned in Figure 1, to have D_1 receive p_1, we need the collaboration of nodes r_2 and D_1 to decode packet $p_1 \oplus p_2 \oplus p_3$ from node r_1, since r_2 overhears p_3, and D_1 overhears p_2.

In our design, intermediate nodes are encouraged to decode the received coded packets at the earliest possible moment, based on the following greedy decoding policy. We let f indicate a data flow, $a \in f$ denote a node belonging to the route of flow f, $r_k(k > 0)$ represent the intermediate nodes on the route, and use $N(a)$ as the single-hop neighbor set of node a. Assuming $F(a, f)$ denotes the forward nodes set of node a on the route of flow f, and $B(a, f)$ indicates the backward nodes set of node a on the route of flow f, the greedy decoding policy is defined as below.

Definition 1. *(Greedy decoding policy). For the n native packets $p_1, p_2, ..., p_n$ which respectively come from the flows $f_1, f_2, ..., f_n$, node c generates the coded packet $p_1 \oplus p_2 ... \oplus p_n$. If $r_k \in F(c, f_i)$ ($1 \leq i \leq n$) can be aware of the native packet p_j of flow f_j($1 \leq j \leq n, j \neq i$), r_k partially decodes the coded packet by removing p_j from it.*

For example, in Figure 1, when the coded packet $p_1 \oplus p_2 \oplus p_3$ arrives at r_2 from coding node r_1, r_2 will decode it to $p_1 \oplus p_2$ once it overhears packet p_3 from S_3, and forwards $p_1 \oplus p_2$ to D_1. Then D_1 recovers p_1 through overhearing p_2 from S_2 to finish the delivery. Note that coding can only reduce the traffic load on the intersection node of flows. Once the coded packets have passed through the intersection nodes such as r_1, the coded form becomes meaningless for the forward nodes in the flow. Thus, the best choice is to decode them by the forward nodes in the flow at the earliest possible moment. Also, usually the neighbors' set of intermediate nodes is different from that of the source and the destination. Hence, the involvement of those intermediate forward nodes can introduce more overhearing, and increase coding opportunities.

C. Necessary Coding Condition and Multi-flow Interference

In coding-aware routing, nodes must independently be evaluated regarding whether they satisfy the coding conditions necessary to conduct coding. Previous works solve the issue mainly in terms of the situation in which only two flows intersect at a node. Here, we propose the multi-flow coding condition as in Definition 2, based on our greedy decoding policy, to evaluate whether a node is a potential coding node.

Definition 2. *(Coding condition). For n flows f_1, f_2, ..., f_n intersecting at node c, if any two flows f_i and f_j satisfy the following condition, the node c can be a potential coding node:*

- *There exists node $q \in B(c, f_i)$ and node $t \in F(c, f_j)$, such that $q = t$ or $q \in N(t)$ or $t \in N(q)$, ($1 \leq i, j \leq n$, $i \neq j$)*

Theorem III.1. *The coding condition in Definition 2 is only a necessary condition of greedy coding awareness.*

Proof: The goal of the destinations in data flows is to obtain their interested native packets from the corresponding sources, respectively. Additionally, based on the basic coding theory, we know that considering that node c codes n native packets $p_1, p_2, ..., p_n$ into the coded packet $p_1 \oplus p_2 ... \oplus p_n$, packet $p_i(1 \leq i \leq n)$ can be extracted, *if and only if* all other packets $p_j(1 \leq j \leq n, j \neq i)$ are known. Therefore, it requires that the forward nodes of the coding node c in that flow can extract the native packet p_i by overhearing other native packets $p_j(1 \leq j \leq n, j \neq i)$ by one or more steps.

First, let us consider one flow, $f_i(1 \leq i \leq n)$. If $c \lhd p_i \oplus p_j$ indicates that node c generates a coded packet based on packet p_i and p_j, $Pack_n$ denotes the set of n native packets, $Pack(F(c, f_i))$ represents the set of packets which can be overheard by the nodes in $F(c, f_i)$, and $R(f_i) = 1$ indicates that the destination node of flow f_i can obtain the native packet p_i successfully. Thus, we have

$$[(c \lhd p_1 \oplus p_2 ... \oplus p_n) \wedge (\{p_j | p_j \in Pack_n \wedge j \neq i\} \subset Pack(F(c, f_i)))] \Leftrightarrow R(f_i) = 1 \quad (1)$$

Then, use $C(c) = 1$, denoting that node c can be a coding node, which requires that the destination node of each flow can obtain its own native packet. Obviously,

$$R(f_i) = 1 (1 \leq i \leq n) \Leftrightarrow C(c) = 1 \quad (2)$$

From Definition 2, the coding condition consists of two parts. The first one is $c \lhd p_1 \oplus p_2 ... \oplus p_n$. The second one is defined as follows.

$$\{q | q \in B(c, f_j) \wedge j \in [1, n] \wedge j \neq i\} \subset (F(c, f_i) \cup N(F(c, f_i))) \quad (3)$$

Since Equation 2 follows the basic coding theory, we only need to verify whether we can deduce Equation 1 with the known coding conditions.

Case 1: As $c \lhd p_1 \oplus p_2 ... \oplus p_n$ is known, we analyze the relationship between Equation 1 and 3. Obviously, if the forward nodes of flow f_i can overhear all the other native packets $p_j (j \neq i)$, at least one of those nodes must be within a one-hop scope of other flows' backward nodes. Hence, $(\{p_j | p_j \in Pack_n \wedge j \neq i\} \subset Pack(F(c, f_i))) \Rightarrow$ Equation 3. If $C(c) = 1$, from Equation 1 and 2, we can get the coding conditions. Thus, the necessary condition is approved.

Case 2: The coding condition does not guarantee that the decoding nodes can obtain all the necessary native packets for decoding. For example, as shown in Figure 2(a), there are two flows, f_1 and f_2, in the network. At some time, flow f_3 starts. According to the Definition 2, flow f_1 and f_3 satisfy the network coding condition in the view of node r_1, and flow f_2 and f_3 satisfy the network coding condition in the view of node r_2. However, after node r_1 codes p_1 and p_3 into $p_1 \oplus p_3$, and broadcasts it, node D_2 can only overhear $p_1 \oplus p_3$, rather than the required p_1 to decode $p_1 \oplus p_2$. As a result, node D_2 cannot obtain p_2, and r_2 should not be a coding node, even though flow f_2 and f_3 satisfy the network coding condition at node r_2. Hence, the sufficient condition cannot be met. □

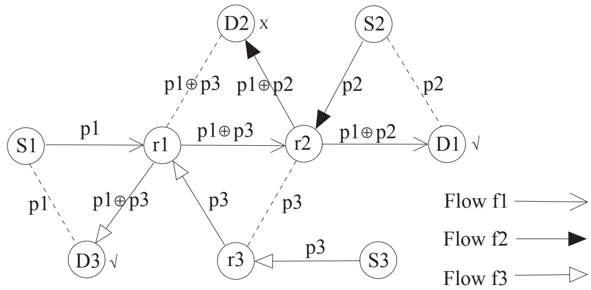
There are two findings about our coding condition. One is that, even though the condition defined in Definition 2 is necessary but not sufficient, it is still very useful to assist source nodes to find potential coding nodes in the routing process. For example, in Figure 1, flows f_1, f_2 intersect at node r_1 and flow f_3 initiates. The source node S_3 can estimate whether r_1 can still be a potential coding node. Through the routing process in section IV, S_3 can get some topology information, $F(r_1, f_1) = r_2, D_1$, $B(r_1, f_1) = S_1$, $F(r_1, f_2) = r_4, D_2$, $B(r_1, f_2) = r_3, S_2$, $F(r_1, f_3) = D_3$, $B(r_1, f_3) = S_3$. Because r_2 and r_4 are the neighbors of S_3, D_2 and D_3 are the neighbor of S_1; D_1 is the neighbor of S_2, r_1 is a potential coding node. Note that a potential coding node may not be the coding one, since the coding condition is insufficient. The other finding is that it is the multi-flow interference defined below that makes our coding condition lack sufficiency.

Definition 3. *(Multi-flow interference). For n flows f_1, f_2, ..., f_n intersecting at node c, a new flow f_{n+1} initiates. If the coding behavior of flow f_{n+1} eliminates the transmission of the native packet p_i at nodes in $B(c, f_i)(1 \leq i \leq n)$, some packets may not get decoded successfully.*

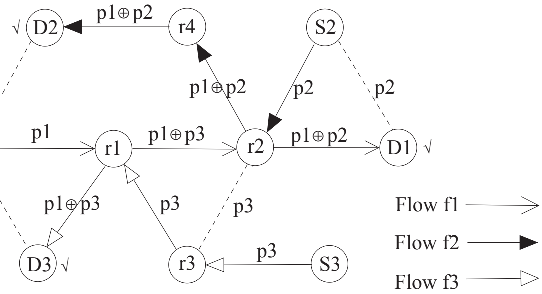
As we have mentioned, in Figure 2(a), the new flow f_3 changes the behavior of node r_1 who was transmitting p_1. Specifically, since flow f_1 and f_3 satisfy the coding condition at node r_1, r_1 generates $p_1 \oplus p_3$ and broadcasts it. That change eliminates the transmission of p_1 at node r_1, and simply makes D_2 unable to decode $p_1 \oplus p_2$ from r_2. In other words, flow f_3 induces the multi-flow interference issue. But it is worth noting that multi-flow interference does not exist in Figure 2(b). The reason is that D_2 can overhear p_1 directly. In the greedy decoding policy, the multi-flow interference occurs only if none of the forward nodes can overhear the native packet p_i from flow f_i. To solve this interference issue, even with the potential coding nodes identified based on our coding condition, source nodes still have to confirm coding opportunities by extra unicast, as introduced in section IV.

D. Routing Metric

The greedy decoding policy and the coding condition are to introduce more coding benefit on paths. Aside from coding benefit, other factors such as link quality and path length should be considered to evaluate a specific route, especially when there exist multiple candidate routes between a pair of given nodes. For example, in Figure 2(b), between the source node S_1 and destination D_1, multiple routes exist, such as $S_1 \rightarrow r_1 \rightarrow r_2 \rightarrow D_1$, $S_1 \rightarrow D_2 \rightarrow r_4 \rightarrow r_2 \rightarrow S_2 \rightarrow D_1$, $S_1 \rightarrow D_3 \rightarrow r_1 \rightarrow r_2 \rightarrow D_1$ et. al. Obviously, the first route has the shortest length. However, other routes may be better if they have more coding benefit or better link qualities. We intend to design the routing metric that can synthesize those three factors comprehensively.



(a) Coding with multi-flow interference



(b) Coding without multi-flow interference

Fig. 2. Coding opportunity and multi-flow interference

1) Coding benefit: We begin with measuring the coding benefit brought by the coding opportunities. Let $P = P_i, 1 \leq i \leq t$ denote the candidate route set of the new flow, while t represents the number of candidate routes. For route P_i, $\beta(P_i)$ indicates its coding benefit. $h(P_i)$ represents the hop number of route P_i between the source and destination node. $\theta_j(1 \leq j \leq m)$ denotes the jth coding node, where m is the number of coding nodes on route P_i. For route P_i, the number of flows through the coding node θ_j is denoted by $n(\theta_j)$, which can be computed from the routing information introduced in Table I. $R = \{\gamma(f_k), 1 \leq k \leq n(\theta_j)\}$ represents the rate set of flows intersecting at coding node θ_j. $\gamma_{min}(\theta_j)$ denotes the minimum rate in set R, which is $\underset{1 \leq k \leq n(\theta_j)}{min} \gamma(f_k)$.

As we know, network coding is a technology transmitting multiple packets using broadcast to improve performance. For example, one transmission can be saved if two packets are coded. Similarly, n transmissions can be saved, if $n+1$ packets are coded at a coding node. For the coding node θ_j on route P_i, it can save $n(\theta_j)$-1 transmissions. Considering that different flows may have different rates, we calculate the coding benefit based on the minimum rate of flows intersecting at the same coding node. The benefit of coding node θ_j is defined below.

$$\beta(\theta_j) = \frac{\gamma_{min}(\theta_j)}{\sum_{1\leq k\leq n(\theta_j)} \gamma(f_k)}(n(\theta_j) - 1) \tag{4}$$

Accordingly, the benefit of route P_i is,

$$\beta(P_i) = \sum_{1\leq j\leq m} \beta(\theta_j) \tag{5}$$

2) Influence of link quality: Due to the possible packet loss, the quality of each link $q(l_x)$ can affect the transmission performance, while $l_x(\ 1 \leq x \leq h(P_i) \ )$ denotes the xth link on route P_i. In other words, it represents the success transmission ratio of this link, which is determined by the transmission count $\delta(l_x)$. Their relationship is defined as follow.

$$q(l_x) = \frac{1}{\delta(l_x)} \tag{6}$$

In practice, we use the expectation of transmission count, $E[\delta(l_x)]$, to measure the link quality. Let $Pb(\delta(l_x) > y)$ be the probability that link l_x needs more than y transmissions to deliver a packet. We have,

$$\begin{aligned}E[\delta(l_x)] &= \sum_{y=0}^{+\infty} Pb(\delta(l_x) > y)\\ &= Pb(\delta(l_x) > 0) - Pb(\delta(l_x) > 1) +\\ &2Pb(\delta(l_x) > 1) - 2Pb(\delta(l_x) > 2) +\\ &3Pb(\delta(l_x) > 2) - 3Pb(\delta(l_x) > 3) ...\\ &= \sum_{y=1}^{+\infty} y * [Pb(\delta(l_x) > y - 1) - Pb(\delta(l_x) > y)]\\ &= \sum_{y=1}^{+\infty} y * Pb(\delta(l_x) = y)\end{aligned} \tag{7}$$

Based on Equation 7, each node can estimate its expected transmission count, and calculate the success transmission ratio of the corresponding hop. The source node can achieve all $q(l_x)(1 \leq x \leq h(P_i))$ on route P_i in the RREP process of routing introduced in section IV-A2. Then, it can calculate the extra increased transmission count $Ex(P_i)$ of route P_i.

$$Ex(P_i) = \sum_{1\leq x\leq h(P_i)} (\frac{1}{q(l_x)} - 1) = \sum_{1\leq x\leq h(P_i)} \frac{1}{q(l_x)} - h(P_i) \tag{8}$$

3) Routing metric definition: By quantifying the coding benefit and link quality, we can define the routing metric. To simplify computing, we treat the transmission number as the hop number. As we mentioned, the routing metric is determined by the hop number, the decreased transmissions of coding benefit, and the increased transmissions of link quality. As a result, we have the metric of MuCAR defined as below.

$$\begin{aligned}MuCAR(P_i) &= h(P_i) - \beta(P_i) + Ex(P_i)\\ &= \sum_{1\leq x\leq h(P_i)} \frac{1}{q(l_x)} - \frac{\gamma_{min}(\theta_j)}{\sum_{1\leq k\leq n(\theta_j)} \gamma(f_k)}(n(\theta_j) - 1)\end{aligned} \tag{9}$$

Obviously, a smaller expected transmission count and a larger coding benefit produces a smaller MuCAR metric value, which indicates lower consumption of network resources in routing, and better routing performance. Compared with other metrics of the existing coding-aware routing schemes, our metric has the following characteristics: a) instead of considering coding benefit solely, the MuCAR metric comprehensively reflects the factors of coding benefit, link quality and path length, which are translated into a single form; b) The metric adapts well to different rates of flows intersecting at the coding node. The benefit of the coding node is calculated with the minimum rate of the flows; c) Our metric can be calculated in a distributed way. After the routing discovery in section IV-A, the source node can acquire sufficient information to estimate the expected transmission count, multi-flow interference and coding benefit of a path.

IV. Routing Implementation

The MuCAR routing protocol includes the following components: route discovery and route selection. Also, we discuss a two-flow compatible mechanism and describe the encoding and decoding algorithms.

A. Route Discovery

We illuminate the routing discovery procedure in Figure 3, which involves the source node, the destination node and the relay nodes of the flow, and consists of four steps below.

1) RREQ (Routing REQuest): Initially, the source node broadcasts RREQ packets. Then, the relay nodes estimate whether they should forward the RREQ packets. After the destination node has received RREQ packets from different relay nodes in some period, it can calculate some candidate routes for transmission. RREQ packets record the IP addresses of the source node, the destination node, and each relay node on the route in the traversing order. Also, each route has the maximum hop limitation, which is configurable. If a route has the exceeded hops, it will not be considered as a candidate route.

2) RREP (Routing REPly): The destination node sends back RREP packets to the source node via relay nodes of candidate routes. Each relay node adds its flows and neighbors' information into RREP packets, and forwards the packets through unicast. Specifically, RREP packets include the number of flows intersecting at relay nodes, and their rates. In addition, the neighbors' information contains one-hop neighbors' IPs of that relay node. Finally, it stores the link quality between that relay node and its next hop.

When the source node receives RREP packets, it records the related routing information into a local table. For example, Table I represents the routing information of flow f_1 in Figure 2(b). The routing of f_1 involves three hops, and four nodes including source node S_1, destination node D_1, and relay nodes r_1, r_2. The situations regarding each node's neighbors are the basis for evaluating the coding condition introduced in section III-C. Moreover, the table also stores the flow states and flow rates. For example, S_1 has only one flow

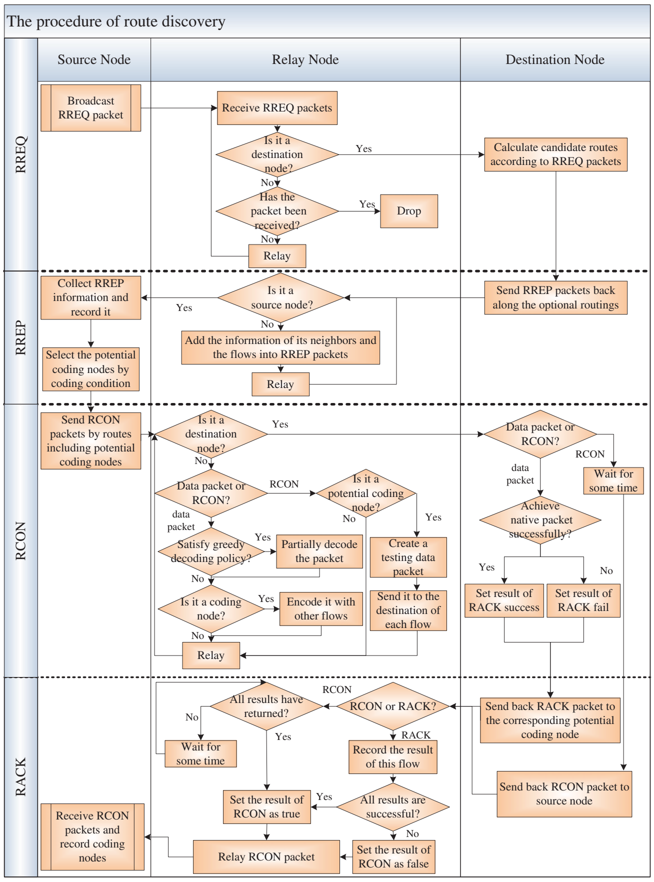


Fig. 3. The procedure of MuCAR

with the flow rate FR_{f1}: r_1 has two flows f_1, f_3 with the flow rate FR_{f1}, and FR_{f3} individually; r_2 has two flows f_1, f_2 and with the respective flow rates FR_{f1}, and FR_{f2}. The information of flow state can help to test and avoid multi-flow interference in the process of RCON (Routing CONfirm) and RACK (Routing ACKnowledge), as well as be used to calculate the routing metric described in section III-D for route selection. Furthermore, the qualities of links $S_1 \to r_1$, $r_1 \to r_2$, $r_2 \to D_1$ are represented as q_1, q_2, q_3 for computing the routing metric too. Each flow has its own local table.

3) RCON (Routing CONfirm): The main task of this step is to test potential coding nodes. If those potential coding nodes can pass the tests in RCON and RACK, they will be considered as real coding nodes. Source nodes select candidate routes which may have several potential coding nodes, and send RCON packets along those routes. After receiving the RCON packet, the intermediate node checks itself whether it is a potential coding node. If yes, it encodes a test data with the history data from other flows, and sends the encoded data to each flow's destination. Meanwhile, the intermediate node still forwards data packets for testing and transmitting. If the intermediate node satisfies the greedy decoding policy, it decodes the data packets partially or completely. If it already

TABLE I
THE LOCAL TABLE OF ROUTING INFORMATION

Flow	f_1			
Hops	3			
Nodes on route	S_1	r_1	r_2	D_1
Coding nodes	-	T	T	-
Neighbors	D_2, D_3	r_3, r_4, D_2, D_3	r_3, r_4, S_2, S_3	S_2, S_3
Flow state	f_1	f_1, f_3	f_1, f_2	f_1
Flow rate	FR_{f1}	FR_{f1}, FR_{f3}	FR_{f1}, FR_{f2}	FR_{f1}
Link quality	q_1, q_2, q_3			

has been confirmed as a real coding node in other flows, it encodes the relevant flow packets and forwards the coded one.

When the RCON packet reaches the destination of a flow, it goes back along the same route after a short delay. On its return, it records the testing results of each intermediate node.

4) RACK (Routing ACKnowledge): This step can be divided into two stages. The first one is that the destination nodes of the testing flows notify the results of the decoding test to the corresponding potential coding nodes who launch the test. The other stage is estimating whether that potential coding node can be a real one. If the feedbacks from those different destinations are all positive, the potential coding node is confirmed as a real one. If network status changes, such due to a new flow joining, or an old flow quitting, the coding nodes should be detected again. Below is the summary of the above four steps.

- RREQ and RREP collect candidate route information, including the hops of the route, and the neighbors and flow information of each relay node on the route.
- RCON notifies the potential coding nodes on a route to check whether multi-flow interference exists. Through the feedback of RCON, the potential interference is avoided, and coding nodes are confirmed.
- RACK plays an important role in sending back the testing results of different flows. These results are the basis of determining the coding nodes in RCON.

B. Route Selection

When the source node finishes the routing discovery, route selection begins, where the source node computes the MuCAR metric value of each candidate route, and selects the best one following the principles below.

- The source node chooses the route with the smallest MuCAR metric value for data delivery.
- If the smallest MuCAR metric value is occupied by multiple routes, the link quality is the highest priority for route selection, since we need to guarantee that packets can reach the destination first.
- If two routes have the same MuCAR metric value and link quality, the source node picks the route with the smaller path length, since the shorter route has less delay, and lower computation cost.

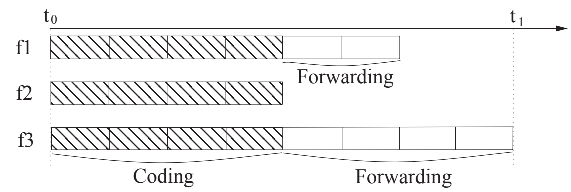


Fig. 4. Coding on flows with different rates at coding node c

Note that with the MuCAR metric, our route selection can achieve a tradeoff among the coding benefit, link quality and path length, instead of simply picking the routes with the largest coding benefit for data delivery.

C. Greedy Aggregation

When there are n flows intersecting at node c, it is possible that not all of the n flows are qualified for coding together. There may only exist $m(m < n)$ flows satisfying the coding condition to get encoded together. For that, we present the following solution. After the source node receives RREP packets, and records the related routing information into a local table, it identifies the potential coding nodes based on the necessary coding condition. Instead of evaluating the coding opportunity of n intersection flows just once, we repeat the evaluation by decreasing n progressively when the evaluation test result is false, until n is equal to 2. If the result becomes true, the source node labels the node as a potential coding node. Then it records the involved flows and puts these flows' information into the header of the RCON packet. When the potential node receives the RCON packet, it can test the interference following the instruction in section IV-A. Under this mechanism, our scheme can maximally code multiple flows together. In the worst case, it degenerates into a two-flow coding-aware routing.

D. Data Transmission

In our opinion, two issues urgently need to be addressed in the data transmission. First of all, because it is unrealistic that all intersecting flows have the same rates, our algorithm should consider processing flows with different flow rates. Secondly, to follow the greedy decoding policy, we must make sure that coded packets are decoded at the earliest possible moment.

The coding operation includes encoding and decoding. To solve the first issue, we design the encoding algorithm to XOR packets from different flows based on the smallest rate of flows. As shown in Figure 4, coding node c is encoding packets from the flows f_1, f_2, and f_3 with Algorithm 1. Considering each flow has a different rate, the number of packets received per flow in time window $[t_0, t_1]$ is different at coding node c, where f_1 has 6 packets, f_2 has 4 packets, and f_3 has 8 packets arrived. According to Algorithm 1, only 4 packets from each flow will be coded at c. 2 packets from f_1 and 4 packets from f_3 will be directly forwarded by node c. As a result, packets of the slowest flow will be fully encoded, and part of the packets from the other faster flows are relayed directly. With such a scheme, coding nodes do not need to wait for the packets of all arriving flows to encode. Instead, they just encode whatever is available at the moment, which can simply reduce the delay.

Algorithm 1 Encoding Algorithm

Input: Packet[n] //Local native packets queue
Output: min, XOR_Packet_Queue, CoPacket_Num_Queue //XOR_Packet_Queue denotes the generated coded packets queue, and CoPacket_Num indicates the queue of the number of native packets involved in packets coding

Statistic min as the minimum packet number of all flows in this generation;
for each round in the min **do**
  Select the first packet in queue from each flow;
  XOR these packets;
  Push the coded packet into XOR_Packet_Queue
  Push the number of involved data packets into CoPacket_Num_Queue;
  Remove those data packets from each flow;
return (XOR_Packet_Queue, CoPacket_Num_Queue, min);

Regarding the second issue, if a relay node overhears some native packets of a coded packet, it can partially or completely decode the coded packet with Algorithm 2. In this way, we guarantee that coded packets are decoded at the earliest possible moment.

Algorithm 2 Decoding Algorithm

Input: XOR_Packet, CoPacket_Num, Packet[n] //Local native packet queue
Output: XOR_Packet, Packet_Num

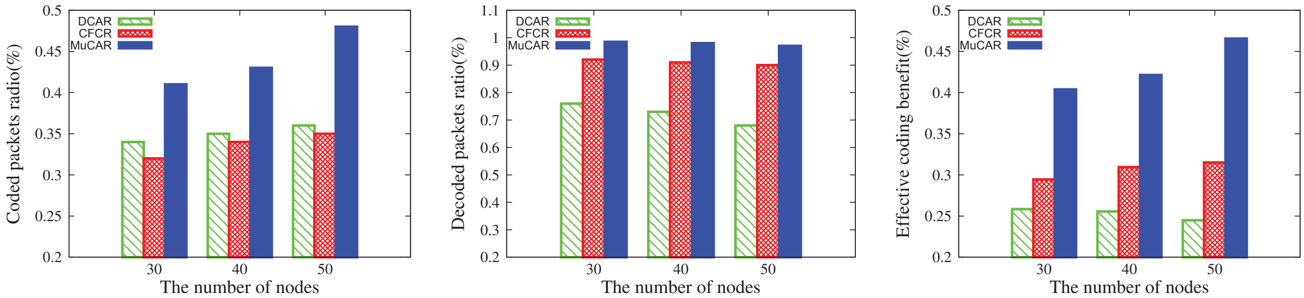
if (CoPacket_Num $\geq$ 1) **then**
  for each packet i in the coded packet **do**
    for each packet j in the local native packet queue **do**
      if (packet i and j is the same packet) **then**
        CoPacket_Num = CoPacket_Num -1;
        XOR packet j with XOR_Packet;
return (XOR_Packet, CoPacket_Num);

V. Simulation and Analysis

We compare the performance of MuCAR with two state-of-the-art protocols, DCAR [4] and CFCR [8], which are both distributed deterministic reactive coding-aware routing schemes, and use XOR coding methods. We implement the three protocols on ns2, which is widely used in network research. To avoid extreme cases, if there is no special explanation, the parameters are set as Table II. Results are averaged over 10 randomly generated network examples. The performance comparison is based on the following perspectives.

A. Effective Coding Benefit

To present the coding benefit, we first analyze the *coded packets ratio* and the *decoded packets ratio*, respectively.

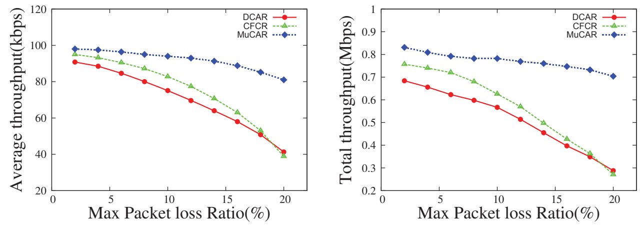


(a) Coded packets ratio (b) Decoded packets ratio (c) Effective Coding Benefit

Fig. 5. Effective Coding Benefit Comparison

TABLE II
THE PARAMETERS OF SIMULATION

Simulation Parameter	Value
MAC protocol	IEEE802.11
Data flow type	UDP/CBR
Packets size	1000B
Flow rate	100kbps
Packet loss ratio	2%
Number of nodes	30
Number of flows	8
Transmission range	250m
Area	1500m * 1500m



(a) Average Throughput (b) Total Throughput

Fig. 6. Throughput Evaluation under Different MPLR

The coded packets ratio is the ratio of the number of coded packets to the number of all transmitted packets. Figure 5(a) presents the comparison of the coded packets ratio of the three protocols with different numbers of nodes in the network. We see that as the number of nodes increases, the advantage of MuCAR becomes obvious. The coded packets ratio of MuCAR exceeds that of DCAR 7% and CFCR 9% in the 30 nodes scenario. When the number of nodes is increased to 50, the gaps of coded ratio turn into 12% and 13%, individually. This is because with the greedy aggregation mechanism, MuCAR can encode the intersecting flows maximally. As a result, it obtains more coding benefit. Meanwhile, the coded packet ratios of the all three protocols ascend when the number of nodes grows. The reason is that more nodes can bring more opportunities for overhearing.

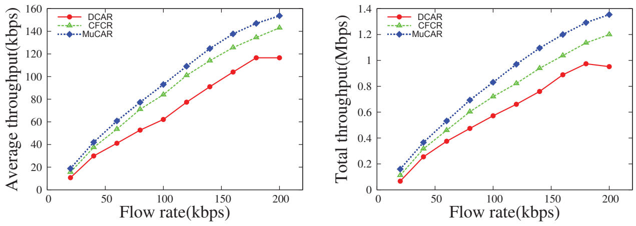
The coded packets ratio reflects the approximate quantity of coded packets in the network transmission. However, due to the multi-flow interference, not all coded packets can be successfully decoded. Figure 5(b) shows the decoded packets ratio, the ratio of the number of decoded packets to the number of coded packets, for the three protocols where the nodes number varies in the network. Different from DCAR, both CFCR and MuCAR have stable decoded packet ratios, even as the number of nodes changes. This phenomenon indicates that higher node density does not generate more multi-flow interference with CFCR and MuCAR. Furthermore, compared to CFCR, MuCAR has a better decoded packet ratio because it has more decoding opportunities at intermediate nodes. Due to the packet loss ratio in wireless channels, the decoded packets ratio of multiple nodes is greater than that of a single node.

Coding benefit is determined by the product of the coded packets ratio and the decoded packets ratio. Figure 5(c) presents the effective coding benefit of the three protocols under different node densities, where we can see: a) no matter the node density, MuCAR has the largest effective coding benefit; b) in CFCR and MuCAR, within some limit, higher node densities can result in a larger coding benefit.

B. Throughput

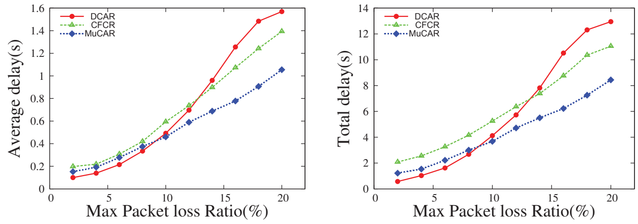
We then evaluate the throughput of the three protocols with the following metrics: a) *Average Throughput*: the average rate of the messages delivered over a random route, which reflects the throughput performance on a single route; b) *Total Throughput*: the sum of the data rates that are delivered to all terminals in the network, which presents the benefit of the whole network. Specifically, we would like to investigate how those two throughput metrics behave under the different *Max Packet Loss Ratio* and *Flow rate*, where Max Packet Loss Ratio (MPLR) is the maximum ratio of the lost packets on the wireless link transmission. For example, MPLR = 10% means the number of the lost packets does not exceed 10 when 100 packets are transmitted over a wireless link. Besides, flow rate is defined as the amount of packets which pass through a specified flow per unit time.

Figure 6(a) and 6(b) exhibit the average and total throughput under different MPLR, respectively. We can see that when the MPLR is low, MuCAR has a slight advantage. When



(a) Average Throughput (b) Total Throughput

Fig. 7. Throughput Evaluation under Different Flow Rate



(a) Average Delay (b) Total Delay

Fig. 8. Delay Evaluation under Different MPLR

the wireless link quality degrades, and the packet loss ratio grows, MuCAR has a larger superiority compared to the other two protocols. The reason is that neither DCAR nor CFCR considers the wireless link quality. In addition, because CFCR is inclined to converge flows to some backbone nodes, the link quality of those nodes may severely affect the throughput. Thus, when the MPLR turns higher, the average and total throughput of CFCR becomes the lowest.

Besides MPLR, we also consider the impact of the flow rate. Figure 7 shows that both the average and total throughput ascend as the flow rate increases. Obviously, MuCAR has the best performance. It is not merely due to the increased coding benefit, but also because of MuCAR's greedy coding and forwarding mechanism. In MuCAR, coding nodes do not require that all the native packets from different flows arrive for encoding. Instead, it simply encodes the available packets, or forwards them if it cannot acquire the related packets for coding. Hence, when the flow rate reaches 200 kbps, compared with DCAR and CFCR, MuCAR has the improved average throughput and total throughput.

C. Delay

Figure 8 represents the average and total end-to-end packet transmission delays between the source and the destination with MPLR varying. Similar to the throughput analysis, the *average delay* is to show the effectiveness of three protocols on a single random route, and the *total delay* is to present the influence of performance for the whole network. We see that while the MPLR is ascending, the average and total delays of the three protocols rises. When the MPLR is small, due to the extra process in routing discovery, CFCR and MuCAR have larger average and total delays than DCAR. Also, because MuCAR selects the high quality link for transmission, it has lower average and total delays than CFCR. When the MPLR is over 12%, with the coding confirmation and link quality consideration, MuCAR has the least delay.

VI. Conclusion

In this paper, we proposed a Greedy Multi-flow-based Coding-Aware Routing (MuCAR) protocol, to improve the routing performance of deterministic routing in wireless networks. The main idea is that by defining the decoding policy and the coding condition in the multi-flow environment, we directly code the multiple intersecting flows in a greedy way to improve the coding benefit. Also, we analyze the multi-flow interference issue, and describe the implementation of MuCAR, including route discovery, route selection, and the algorithms of encoding and decoding.

Acknowledgment

This research was supported in part by the Key Laboratory of Aerospace Information Security and Trusted Computing Ministry of Education, the NSF grants CNS 1460971, CNS 1439672, CNS 1301774, ECCS 1231461, ECCS 1128209, and CNS 1138963, the National Natural Science Foundation of China under Grant No. 61272451, 61173154, 61232002, 61373169, 61332019, and the Major State Basic Research Development Program of China under Grant No. 2014CB340600. The corresponding author is Ruiying Du.

References

[1] R. Ahlswede, N. Cai, S. R. Li, and R. W. Yeung, "Network information flow," *IEEE Transactions on Information Theory*, 2000.

[2] J. Wang, J. Wang, K. Lu, Y. Qian, B. Xiao, and N. Gu, "Optimal design of linear network coding for information theoretically secure unicast," in *Proc. of IEEE INFOCOM*, 2011.

[3] J. Chen, T. Li, and R. Du, "Efficient reliable opportunistic network coding based on hybrid flow in wireless network," in *China Communications*, 2011.

[4] J. Le, J. C. S. Lui, and D. M. Chiu, "Dcar: Distributed coding-aware routing in wireless networks," *IEEE Transactions on Mobile Computing*, 2010.

[5] S. Sengupta, S. Rayanchu, and S. Banerjee, "Network coding-aware routing in wireless networks," in *IEEE/ACM Transactions on Networking*, 2010.

[6] C. Yang and D. P. Agrawal, "Hycare: Hybrid coding-aware routing with etox metric in multi-hop wireless networks," in *Proc. of IEEE MASS*, 2013.

[7] S. Y. Oh, B. Shen, and M. Gerla, "Network coding over a manet proactive link state routing protocol and tdma scheduling," in *Proc. of MILCOM*, 2012.

[8] 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*, 2014.

[9] B. Guo, H. K. 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*, 2011.

[10] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard, and J. Crowcroft, "Xors in the air: Practical wireless network coding," in *Proc. of ACM SIGCOMM*, 2006.

[11] A. Khreishah, I. M. Khalil, and J. Wu, "Universal opportunistic routing scheme using network coding," in *Proc. of IEEE SECON*, 2012.

[12] S. Wang, G. Tan, Y. Liu, H. Jiang, and T. He, "Coding opportunity aware backbone metrics for broadcast in wireless networks," in *Proc. of IEEE INFOCOM*, 2013.

[13] B. Radunovic, C. Gkantsidis, P. Key, and P. Rodriguez, "Toward practical opportunistic routing with intra-session network coding for mesh networks," in *IEEE/ACM Transactions on Networking*, 2009.