Computer Networks 105 (2016) 194–206

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Distributed Greedy Coding-aware Deterministic Routing for multi-flow in wireless networks

Jing Chen[a], Kun He[a,*], Quan Yuan[b], Ruiying Du[a], Lina Wang[a], Jie Wu[c]

[a] State Key Laboratory of Software Engineering, Computer School, Wuhan University, Wuhan, China
[b] Department of Math and Computer Science, University of Texas-Permian Basin, Odessa, TX, USA
[c] Department of Computer and Information Sciences, Temple University, Philadelphia, PA, USA

A R T I C L E I N F O

Article history:
Received 4 January 2016
Revised 10 May 2016
Accepted 31 May 2016
Available online 11 June 2016

Keywords:
Network coding
Multi-flow
Reliability
Routing

A B S T R A C T

As one of the compelling performance improvement techniques, network coding is widely used for designing routing protocols in wireless networks. Specifically, in deterministic routing, coding benefit is viewed as an important factor for distributed route selection. However, most of the existing deterministic routing protocols only detect two-flow coding opportunities in the route discovering phase, but the multi-flow scenario is not researched sufficiently. It is obvious that multi-flow coding can improve the coding benefit in complex network environments. In this paper, we analyze the challenges of the multi-flow coding, and propose a Distributed Greedy Coding-aware Deterministic Routing (DGCDR) for multi-flow in wireless networks. To increase the potential coding opportunities, a decoding policy and a coding condition are defined in the multi-flow environment, which exploit the coding benefit of multiple intersecting flows in a greedy way. Meanwhile, considering the interference introduced by multi-flow coding, we design an extra confirmation process in our protocol. Furthermore, to enhance the flexibility of packet delivery and coding, we propose a greedy aggregation mechanism and a greedy coding algorithm. From the simulation results, we can find that DGCDR can induce a competitive performance in terms of increased coding benefit, decreased delay, larger throughput, and smaller queue size.

1. Introduction

Wireless Network Coding (NC) [1–4], which exploits the broadcast characteristic of the wireless medium to augment the capacity of the network, highlights a novel direction in routing to improve network throughput [5–8]. Unlike the traditional forwarding mode, intermediate nodes can initially encode packets from different flows into a set of fewer packets, and then forward those packets which would be further decoded at destinations, this is known as *inter-flow* coding [9]. To optimize the transmission efficiency, researchers introduce such a technique into the routing protocol and it is called *coding-aware routing* [10–13].

The main issue addressed in the coding-aware routing schemes is how to obtain more coding opportunities in routing, leading to higher transmission efficiency [14–16]. Relevant existing work can be classified 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 save transmissions. Khreishah et al. [17] proposed a distributed opportunistic routing based on network coding, by formulating the problem with arbitrary channel conditions as a convex optimization problem, and presenting an optimal back-pressure algorithm on that. CodePipe [18] 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 improvements in throughput, energy-efficiency and fairness. Different from opportunistic routing, deterministic routing determines particular paths based on coding opportunities before packet delivery [13]. That means, the source node evaluates the number of coding opportunities on each candidate route, and selects the route with more coding opportunities to transmit packets. Obviously, those schemes have the advantage of controllable performance, even if some extra information is needed to calculate potential paths.

Based on the methods used for collecting extra information, those deterministic coding-aware schemes can be further subdivided into two classes: *proactive* and *reactive*. Proactive protocols [19–21] maintain a constantly updated topology understanding to estimate the availability and the coding opportunity of a path for route selection. Sengupta [19] et al. proposed CA-PATH-CODE,

* Corresponding author.
E-mail addresses: chenjing@whu.edu.cn (J. Chen), milloglobe@gmail.com (K. He).

http://dx.doi.org/10.1016/j.comnet.2016.05.027
1389-1286/

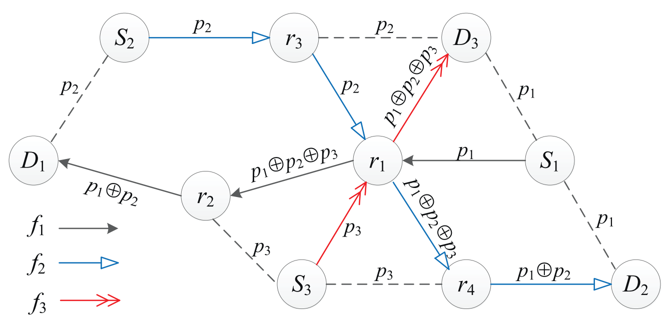


Fig. 1. Example of decoding at intermediate nodes in a multi-flow network.

a XOR-based coding-aware routing, based on the COPE [22] approach, which leveraged the coding opportunities in the two-hop range. HyCare in [20] exploited the Expected Time of the Overall Transmission (ETOX) as the link-state information, to find possible network coding opportunities in routing. [21] 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 that regarding neighbors and flow rates, to estimate coding opportunities. In contrast, reactive protocols establish paths only upon request [23], and therefore they usually require fewer resources. Researchers in [13] 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. [24] 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 when detecting paths with coding opportunities, without discussing the multi-flow coding sufficiently. Bin Guo et al. [25] presented a general discussion on the coding condition, but they did not consider the multi-flow interference and other implementation details. Such an 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 Fig. 1, initially there are 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 $p_1 \oplus p_2$. But, if there is a new flow f_3 ($S_3 \to r_1 \to D_3$), whose packet is p_3, intersecting other two 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 DGCDR to improve the coding benefit in 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.

1.1. Challenges

Our work introduces several key challenges needing to be solved. First of all, multi-flow coding may change nodes' forwarding behaviors, which can crash the sufficiency of the existing coding condition, defined as *multi-flow interference* in this paper. Thus, the evaluation of coding opportunities cannot only depend on the topological relationship as in the two-flow environment. Secondly, a coding opportunity is identified by whether the encoded packet can successfully get 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. 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, DGCDR has to be backward compatible with the two-flow coding in the worst case, which makes the number of coding opportunities in the two-flow coding be its lower limit. Finally, in reality, the multi-flow environment introduces flow rate differences. As a practical coding system, both real-time and adaptive requirements should be considered simultaneously.

1.2. Contributions

The main contributions of our paper are summarized as follows:

- In contrast to previous coding-aware routings [13,24,25], which claim that the coding condition is sufficient or even sufficient and necessary, to the best of our knowledge, this is a first work to prove that only a necessary coding condition can be achieved by analyzing the topological relationship of nodes in a multi-flow environment;
- To identify the real coding opportunities in potential coding nodes found by the necessary condition, we propose a scheme to sense and avoid the *multi-flow interference* in the process of route discovery;
- Different from the previous two-flow protocols, which require destination nodes to decode packets, we propose a greedy decoding policy to regulate when and how to decode packets in multi-flow scenes, cooperatively;
- To ensure the backward compatibility of our routing protocol, we design a greedy aggregation mechanism to maximally code the qualified flows together, which, in the worst case, is backward compatible to the two-flow coding;
- We exploit a greedy encoding and decoding algorithm to reduce the transmission delay, and it can automatically match the different rates of flows.

Compared with our conference version [26], we make improvements in the three aspects as follows. First, we provide the details of protocol implementation to help readers to understand our protocol clearly. Secondly, we add the route maintenance process to enhance the compatibility of our protocol in dynamic wireless networks. Thirdly, we rearrange our simulations and supply the detailed comparisons in the aspects of algorithm characteristics, packet loss ratio, mobility, and average flow rate, respectively.

1.3. Paper organization

The reminder of the paper is organized as follows. The problem statement is presented in Section 2. In Section 3, we discuss the coding condition and the decoding policy in DGCDR. The routing metric and the detailed protocol construction are described in Sections 4 and 5, respectively. Section 6 evaluates the performance. The paper is concluded in Section 7.

2. Problem statement

2.1. 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 [22] 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 [14]. Intermediate nodes can decode received coded packets cooperatively if sufficient information is acquired by overhearing. Once the intended destination receives the native packet extracted from the coded packet, the message delivery is finished, as shown in Fig. 1. Also, due to the dynamic nature of the wireless networks, the quality of a link between any two nodes may change unpredictably.

2.2. Design goal

The main aim of our routing protocol is to obtain as much coding benefit as possible in routing for multi-flow wireless networks. We refine the design goals as follows.

- DGCDR is a distributed routing protocol, which means that each node estimates the coding opportunity, and selects the suitable route by itself, based on the information from its neighbours and the feedback from intermediate nodes in candidate routes.
- DGCDR has strong flexibility in handling various environments, such as different numbers of flows, different flow rates, and multiple coding nodes in a route.
- DGCDR has good realizability, which requires not only theoretical analysis, but also the implementation of specific considerations.

3. Decoding policy and coding condition

3.1. Greedy decoding policy

Previous works, such as DCAR [13], CFCR [24], 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 Fig. 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. If we let f indicate a data flow, $a \in f$ denotes a node belonging to the route of flow f, $r_k(k > 0)$ represents the intermediate nodes on the route, and we use $N(a)$ as the single-hop neighbor set of node a. Assuming that $F(a, f)$ 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 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 $r_k \in F(c, f_i)$ $(1 \le i \le n)$ can be aware of the native packet p_j of flow $f_j(1 \le j \le n, j \ne i)$, r_k partially decodes the coded packet by removing p_j from it.

For example, in Fig. 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.

3.2. Necessary coding condition and multi-flow interference

In coding-aware routing, nodes must independently be evaluated regarding whether they satisfy the coding condition to conduct coding. Previous works solve the issue mainly in the situation that only two flows intersect at a node. Here, we propose the multi-flow coding condition as in Definition 2, 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 $t \in N(q)$, $(1 \le i, j \le n, i \ne j)$

Theorem 3.1. *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 $p_i (1 \le i \le n)$ can be extracted, *if and only if* all other packets $p_j (1 \le j \le n, j \ne i)$ 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 $p_j (1 \le j \le n, j \ne i)$ by one or more steps.

First, let us consider one flow, $f_i (1 \le i \le n)$. If $c \lhd p_i \oplus p_j$ indicates that node c generates a coded packet based on packets 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. We have

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

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

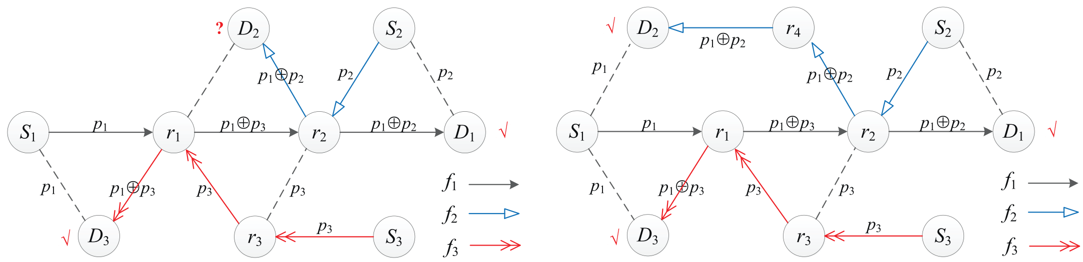
$$R(f_i) = 1 (1 \le i \le n) \Leftrightarrow C(c) = 1 \tag{2}$$

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

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

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

Case 1: As $c \lhd p_1 \oplus p_2 \ldots \oplus p_n$ is known, we analyze the relationship between Eqs. 1 and 3. Obviously, if the forward nodes of



(a) Coding with multi-flow interference (b) Coding without multi-flow interference

Fig. 2. Coding opportunity and multi-flow interference.

flow f_i can overhear all the other native packets $p_j (j \neq i)$, at least one of those nodes must be within one-hop scope of other flows' backward nodes. Hence, $(\{p_j | p_j \in Pack_n \wedge j \neq i\} \subset Pack(F(c, f_i))) \Rightarrow$ Eq. 3. If $C(c) = 1$, from Eqs. 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 Fig. 2(a), there are two flows, f_1 and f_2 in the network. At some time, flow f_3 starts. According to Definition 2, flows f_1 and f_3 satisfy the network coding condition in the view of node r_1, and flows 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 flows 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 in finding potential coding nodes in the routing process. For example, in Fig. 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 5, 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 neighbors of S_1; D_1 is the neighbor of S_2 and r_1 is a potential coding node. Note that a potential coding node may not be the coding node, since the coding condition is insufficient. The other 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 be successfully decoded.

As we have mentioned, in Fig. 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 Fig. 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 forwarding 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 5.

Table 1
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		

4. Routing metric

The greedy decoding policy and the coding condition introduce more path coding benefits. Aside from coding benefits, 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 Fig. 2(b), between the source node S_1 and destination D_1, multiple routes exist, such as $S_1 \to r_1 \to r_2 \to D_1$, $S_1 \to D_2 \to r_4 \to r_2 \to S_2 \to D_1$, $S_1 \to D_3 \to r_1 \to r_2 \to D_1$ etc. Obviously, the first route has the shortest length. However, other routes may be better if they have more coding benefits or better link qualities. We intend to design the routing metric that can synthesize those three factors comprehensively.

4.1. 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 1. $R = \{\gamma(f_k), 1 \leq k \leq n(\theta_j)\}$ 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.

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

Accordingly, the benefit of route P_i is,

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

4.2. Influence of link quality

Due to the possible packet loss, the quality of each link $q(l_x)$ can affect the transmission performance, while $l_x(\ 1 \le x \le h(P_i)\ )$ 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. $E[\delta(l_x)]$ is equal to the sum of $Pb(\delta(l_x) > y)$ while y changes from zero to infinity. We have,

$$E[\delta(l_x)] = \sum_{y=0}^{+\infty} Pb(\delta(l_x) > y) \tag{7}$$

Based on Eq. 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 $q(l_x)(1 \le x \le h(P_i))$ on route P_i in the RREP process of routing introduced in Section 5.1.2. Then, it can calculate the extra increased transmission count $Ex(P_i)$ of route P_i.

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

4.3. Routing metric definition

By quantifying the coding benefit and link quality, we can define our 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 DGCDR defined below:

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

Obviously, a smaller expected transmission count and a larger coding benefit produces a smaller DGCDR 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 DGCDR 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 5.1, the source node can acquire sufficient information to estimate the expected transmission count, multi-flow interference, and coding benefit of a path.

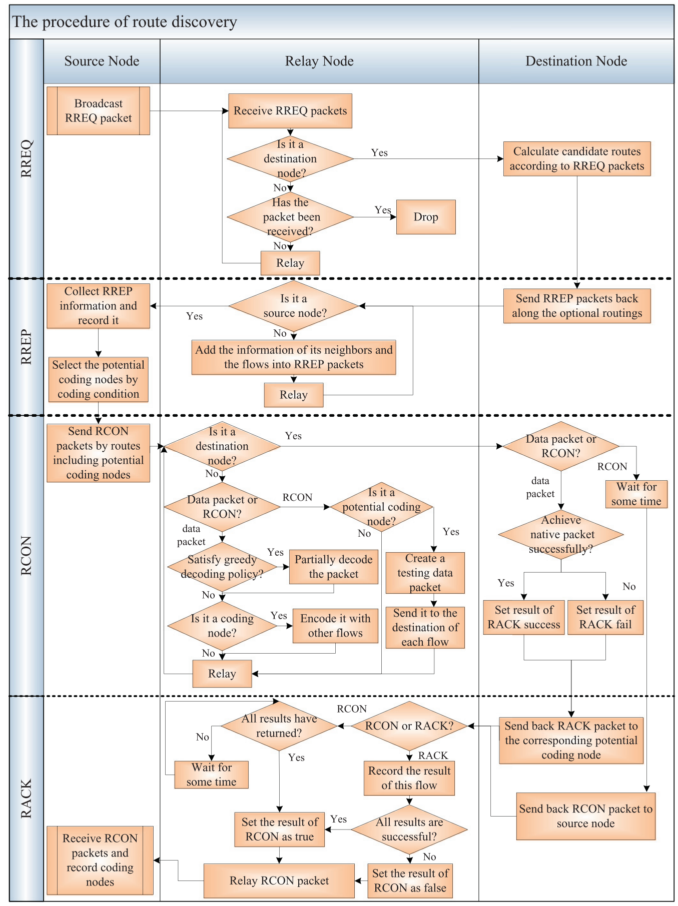


Fig. 3. The procedure of DGCDR.

5. Routing protocol construction

The DGCDR routing protocol consists of the following components: route discovery, route selection, and route maintenance. Besides, we exploit a two-flow compatible mechanism and discuss the greedy data transmission algorithm.

5.1. Route discovery

The routing discovery procedure has four steps as shown in Fig. 3, which involves the source node, the destination node and the relay nodes of the route. The main aim of this procedure is to collect enough information, such as coding benefit, link quality, and hops, as a supplement to decision-making.

5.1.1. RREQ (Routing REQuest)

At first, the source node broadcasts RREQ packets to its neighbors. Then, relay nodes estimate whether they already have the received RREQ packets. If they do, received RREQ packets should be dropped. Otherwise, packets are forwarded. After the destination node has received RREQ packets from different relay nodes in some period, it can calculate some candidate routes for transmission. Referring to DSR, we design the format of the RREQ packet header as shown in Fig. 4. The field 'Type' indicates the type of packet, such as RREQ, RREP and so on. The field 'Length' denotes the length of the packet, and 'RREQ_ID' denotes a unique identity of RREQ packet. The field 'Destination_IP' and 'Source_IP', are the IP addresses of the destination and source nodes, respectively, as their names imply. At the end of the packet, there are n fields 'Relay_IP', which sequentially record IP addresses of the relay nodes.

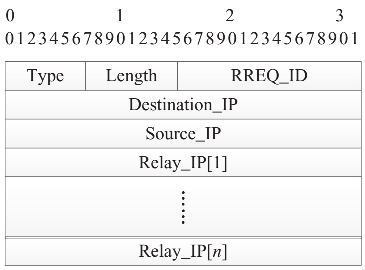


Fig. 4. RREQ header.

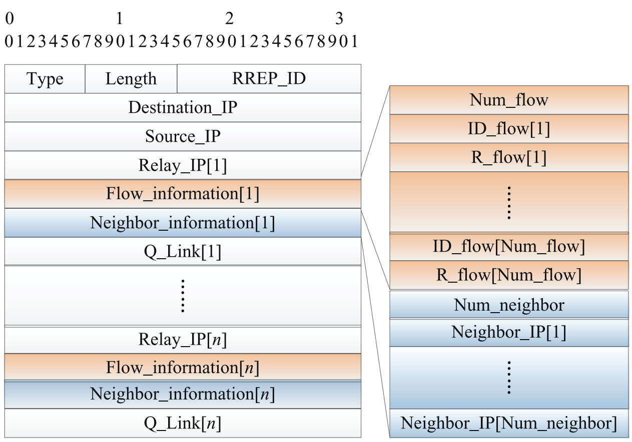


Fig. 5. RREP packet header.

Note that the n denoting the hop limitation of the route is configurable. In other words, if the length of a route exceeds n, it will not be a candidate route.

5.1.2. RREP (Routing REPly)

The destination node sends RREP packets back 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. The structure of the RREP packet header is shown in Fig. 5. Compared to RREQ packets, behind each 'Relay_IP', it has some extra blocks consisting of three additional fields 'Flow_information', 'Neighbor_information' and 'Q_Link'. The first field describes the information of flows intersecting at this relay node. It includes the number of flows, the identity and rate of each flow which are denoted by fields 'Num_flow', 'ID_flow' and 'R_flow', individually. Neighbors' information, including their number and IP address, is stored in the second field 'Neighbor_information'. The third field 'Q_Link' indicates the quality of links between this node and its next hop in routing.

When the source node receives RREP packets, it records the related routing information into a local table. For example, Table 1 presents the routing information of flow f_1 in Fig. 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 3.2. Moreover, the table also stores the flow states and flow rates. For example, S_1 has only one flow 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 4 for route selection. Furthermore, the qualities of links

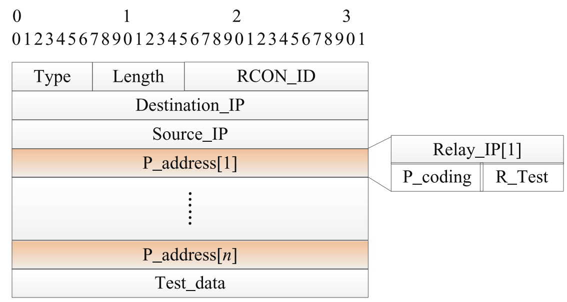


Fig. 6. RCON packet header.

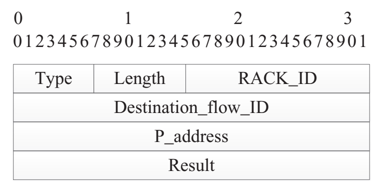


Fig. 7. RACK packet header.

$S_1 \rightarrow r_1$, $r_1 \rightarrow r_2$, $r_2 \rightarrow D_1$ are represented as q_1, q_2, q_3 for computing the routing metric too. Each flow has its own local table.

5.1.3. 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 to see whether or not 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 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.

The format of an RCON packet is shown in Fig. 6. Different from RREQ and RREP packets, in the RCON packet, 'Relay_IP' is replaced by an extended field 'P_address' which contains two other fields 'P_coding' and 'R_TEST'. Both of them are Boolean type. The first one indicates whether the relay node is a potential coding node, while the latter one indicates the result of testing. Besides, at the end of RCON, the 'Test_data' field contains the data used to be coded with other flows through the potential coding node.

5.1.4. RACK (Routing ACKnowledge)

This step can be divided into two stages. The first stage is that in which 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 is estimating whether that potential coding node can be a real one. If the feedback from those different destinations is all positive, the potential coding node is confirmed as a real one. Fig. 7 presents the packet format of RACK. The field 'Destination_flow_ID' indicates the identity of testing flow, such as f_2, f_3. The field 'P_address' denotes the IP address of the potential coding node which launches testing. The field 'Result' indicates the testing result.

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 returning the testing results of different flows. These results are the basis of determining the coding nodes in RCON.

5.2. Route selection

When the source node finishes the routing discovery, route selection begins. This is completed in that the source node computes the DGCDR metric value of each candidate route, and selects the best one following the principles below.

- If all DGCDR values are different, the path with the smallest one is the best choice for data delivery.
- If some paths have the same smallest DGCDR value, the link quality is the highest priority factor. First, we must guarantee that packets can reach the destination.
- The trends of path length and coding benefit are the same, and both of them intend to be smaller. In other words, if paths P_i and P_j have the same DGCDR and Ex value, the path with the smaller path length has a smaller coding benefit. In this situation, the path also has fewer relay nodes, computation and communication costs.

Note that with the DGCDR 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. The routing selection algorithm is shown in Algorithm 1.

Algorithm 1: Route selection algorithm.

Input: P, t, routing information table
Output: The optimization routing P_m

$P_m = Null$; $DGCDR_m = \infty$; $j = 0$;
for $i = 1; i++; i \leqslant t$ **do**
  $DGCDR(P_i) = h(P_i) - \beta(P_i) + Ex(P_i)$;
  if $DGCDR(P_i) < DGCDR_m$ **then**
    $j = i$;
    $P_m = P_i$;
    $DGCDR_m = DGCDR(P_i)$;
  else
    if $DGCDR(P_i) == DGCDR_m \&\& Ex(P_i) < Ex(P_j)$ **then**
      $j = i$;
      $P_m = P_i$;
    else
      if $DGCDR(P_i) == DGCDR_m \&\&$ $Ex(P_i) == Ex(P_j) \&\& h(P_i) < h(P_j)$ **then**
        $j = i$;
        $P_m = P_i$;
      end
    end
  end
end
return P_m;

0 1 2 3 4 5 6 7	8 9 0 1 2 3 4 5	6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
Type	Length	RERR_ID
Reporter_IP		
Rep_Des_IP		
Flow_ID		
Failnode_IP		

Fig. 8. RRER packet header.

5.3. Route maintenance

Due to the dynamic nature of wireless networks, links and flows may change unpredictably. In our scheme, the procedure of routing maintenance should be able to cope with environment change caused by different factors.

In the first case where links fail, the source node should be notified by the RERR (Routing ERRor) packet whose structure is shown in Fig. 8. The field ‘Reporter_IP’ is the node address in routing that finds the link failure. The field ‘Rep_Des_IP’ is the source node routing address which is the reporting destination. The field ‘Flow_ID’ and ‘Failnode_IP’ indicate the identity of flow and the address of the problem node, respectively. From the ‘Flow_ID’, relay nodes can find the corresponding path and backward ‘Failnode_IP’ to node ‘Rep_Des_IP’. After the RERR packet arrives at the source node, the procedures of routing discovery and selection will be launched again.

In the other case where a flow exits, the source node will send a RCON packet with null ‘testdata’ to each coding node in the current path. In other words, those coding nodes will check their coding opportunity again. Because they still satisfy the necessary coding condition, there is a big probability that the result is positive. If it is true, nothing needs to be changed. Otherwise, the source node begins the procedures of routing discovery and selection.

5.4. 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 conditions required for being 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 the n intersection flowing just once, we repeat the evaluation while 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 potential coding node. Then it records the involved flows and put 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 5.1. Under this mechanism, our scheme can maximally code multiple flows together. In the worst case, it degenerates into a two-flow coding-aware routing.

5.5. Data transmission

To make it more practical, the implementation of the coding algorithm should be considered. Fig. 9 shows the linked list which is stored locally in each node. The header of the linked list includes two fields ‘Node_ID’ and ‘Packet_Num’, indicating the identity of the node and the number of packets, individually. The items within the linked list have five parts. ‘Packet i’ is the index of the packet, and the maximum value of the index is equal to the value

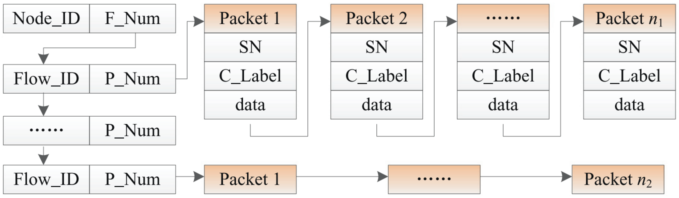


Fig. 9. The list structure of packets.

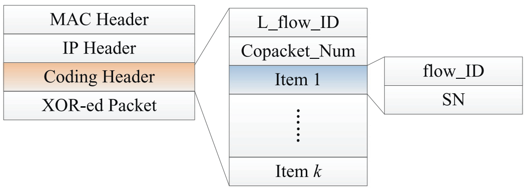


Fig. 10. The structure of a coded data packet.

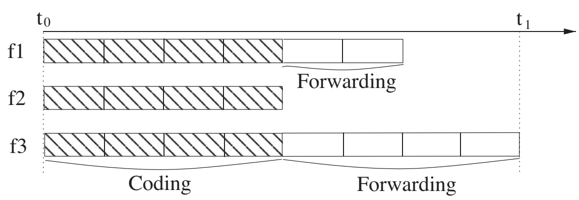


Fig. 11. Coding on flows with different rates at coding node c.

of 'Packet_Num'. 'flow_ID' and 'SN' are the identity of flow and the sequence number of the packet, respectively. 'C_Label' is an identification used to display whether that packet has been used to be coded with the packets from the other flows. 'data' stores the content of the packet. The linked list is the basis of the coding algorithm.

Fig. 10 presents the structure of a coded data packet. As a normal data packet, it also has MAC, IP headers and data portions denoted as 'XOR-ed Packet'. Besides, it has an extended part called 'Coding header' which contains the information for coding. 'L_flow_ID' denotes the identity of the flow, and 'Copacket_Num' indicates the total number of the coded packets. 'Item' presents the detailed information of the packet. It consists of 'flow_ID' and 'SN' fields, which are the same as the fields shown in Fig. 9.

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 Fig. 11, coding node c is encoding packets from the flows f_1, f_2, and f_3 with Algorithm 2. Considering each flow has a different rate, the received packet numbers of flows in time window $[t_0, t_1]$ are different at coding node c, where f_1 has 6 packets, f_2 has 4 packets, and f_3 has 8 packets that have arrived. According to Algorithm 2, 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 flows arriving to encode. Instead, they just encode whatever is available at the moment, which can simply reduce the delay.

Algorithm 2: Encoding algorithm.

```
Input: Packet[Packet_Num] //native packet queue
Output: XOR_Packet, CoPacket_Num
k = 1;
XOR_Packet = Null;
CoPacket_Num = 0;
for i = 1; i++; i ⩽ Packet_Num do
    m = 0;
    for j = 1; j++; j ⩽ k do
        if Packet[i].flow_ID = = item[j].flow_ID then
            break;
        end
        m++;
    end
    if m == k then
        k++;Packet[i].C_label = True;
        item[k].flow_ID = Packet[i].flow_ID;
        item[k].SN = Packet[i].SN;
        XOR_Packet = XOR_Packet ⊕ Packet[i].data;
    end
end
CoPacket_Num = k;
return XOR_Packet, CoPacket_Num;
```

Algorithm 3: Decoding algorithm.

```
Input: XOR_Packet, CoParket_Num
Output: ParXOR_Packet, ParPacket_Num
ParXOR_Packet = Null;
ParPacket_Num = 0;
if CoPacket_Num > 1 then
    for i = 1; i++; i ⩽ CoPacket_Num do
        for j = 1; j++; j ⩽ Packet_Num do
            if Packet[j].flow_ID == item[i].flow_ID &&
            Packet[j].SN == item[i].SN then
                CoPacket_Num;
                XOR_Packet ⊕= Packet[j].data;
                Delete item[i] from the coded packet;
            end
        end
    end
end
ParPacket_Num = CoPacket_Num;
ParXOR_Packet = XOR_Packet;
return ParXOR_Packet, ParPacket_Num;
```

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 3. In this way, we guarantee that coded packets are decoded at the earliest possible moment.

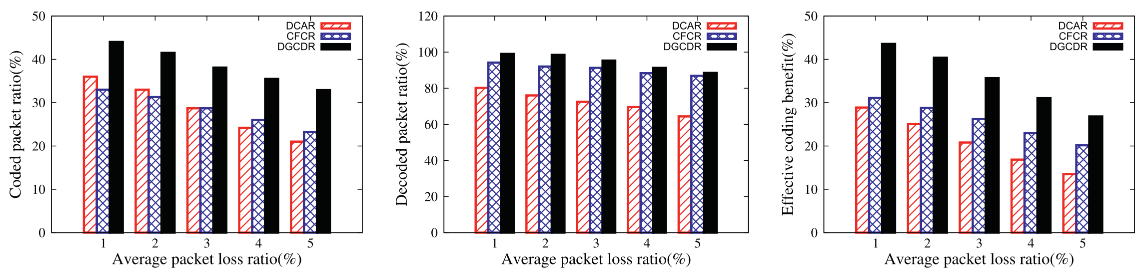
6. Performance analysis

6.1. Theoretical analysis

From the Section 5, we can find that the most complex component in our routing protocol is routing discovery procedure, which contains four steps: Routing REQuest (RREQ), Routing REPly (RREP), Routing CONfirm (RCON), and Routing ACKnowledge (RACK). To analyze the complexity of this procedure, we assume that the average number of candidate routes is ϕ and the average number of crossing flows in a candidate route is ζ. In steps RREQ and RREP, source nodes broadcast routing requests, and destination nodes return the

Table 2
Algorithm comparisons.

	Coding method	Coding type	Forwarding method	Routing type	Mobility	Computation approach
COAB [18]	XOR	Inter-flow	Deterministic	Back-bone	No	Centralized
HyCare [20]	XOR	Inter-flow	Deterministic	Proactive	Yes	Distributed
UNIV [17]	RLNC	Intra-flow	Deterministic	Proactive	No	Distributed
CodePipe[27]	RLNC	Inter/Intra-flow	Probabilistic	opportunistic	No	Centralized
LSMP [21]	RLNC	Intra-flow	Deterministic	Proactive	Yes	Distributed
NUM [14]	RLNC	Intra-flow	Probabilistic	Opportunistic	No	Distributed
CA-CODE [19]	XOR	Inter-flow	Deterministic	Proactive	No	Centralized
DCAR [13]	XOR	Inter-flow	Deterministic	Reactive	Yes	Distributed
CFCR [24]	XOR	Inter-flow	Deterministic	Reactive	Yes	Distributed
DGCDR	XOR	Inter-flow	Deterministic	Reactive	Yes	Distributed



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

Fig. 12. Effective Coding Benefit with different PLRs.

reply packets along candidate routes. Thus, the time complexity is O(n). In steps RCON and RACK, source nodes launch confirm process to determine whether the potential coding nodes are available. Each potential coding node should interact with correlative flows. Thus, the time complexity is O($\phi \times \zeta$).

6.2. Simulation environment

To analyze the influence of various factors, we adopt ns2, which is a discrete event simulator targeted at networking research, to evaluate the performance of DCAR [13], CFCR [24], and DGCDR. As in Table 2, though many state-of-the-art coding-aware routing protocols have been proposed, only these three protocols are the distributed deterministic reactive coding-aware routing schemes based on inter-flow coding. Note that RLNC denotes the random linear network coding method in the table. In our simulation, if there is no special explanation, the parameters are set as follows. In the simulation area of 1000 × 1000 m^2, there exist 40 nodes whose transmission range is 200 m and the MAC protocol is IEEE802.11. And there also exist 6 UDP/CBR flows intersecting randomly with different source and destination nodes, where the packet size is 1000B and the average flow rate is 100 kbps. In addition, the packet loss ratio of a link is 1%. Additionally, the mobility of nodes adopts the random direction model. We analyze the performance under three factors, including packet loss ratio, mobility, and flow rate.

6.3. Results and analysis

6.3.1. Packet loss ratio

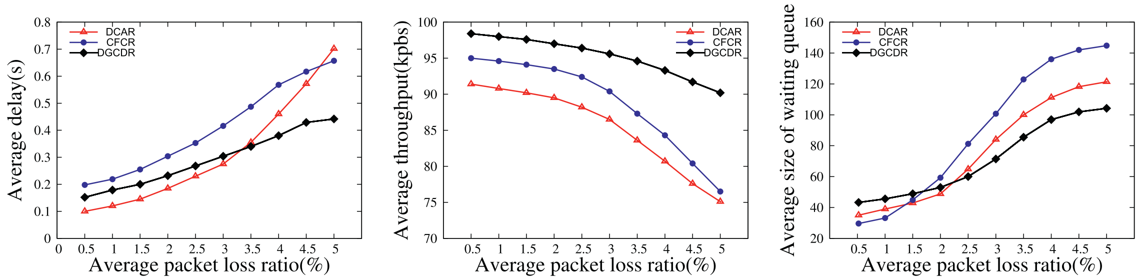
The influence of packet loss ratio is presented from the following aspects: coding benefit, delay, throughput, and queue size.

To analyze the *coding benefit*, we first investigate the *coded packets ratio* and the *decoded packets ratio*, respectively. The coded packets ratio is the ratio of the number of coded packets to the number of all transmitted packets. Fig. 12(a) presents the comparison of the coded packets ratio of the three protocols with different *Average Packet Loss Ratio* (APLR), where APLR is the average ratio of the lost packets in the wireless link transmission. For example, APLR = 2% means the average number of the lost packets is 2 when 100 packets are transmitted over a wireless link. We see that as the wireless link quality degrades and the packet loss ratio grows, the coded packet ratio decreases, but the advantage of DGCDR becomes obvious. The coded packets ratio of DGCDR exceeds that of DCAR 18.2% and CFCR 25% with 1% APLR. When APLR is increased to 5%, the gaps of the coded ratio turn into 36.2% and 29.4%, individually. The reason is that neither DCAR and CFCR considers the wireless link quality. The coded packets ratio reflects the approximate quantity of the coded packets in the network transmission. However, due to the multi-flow interference, not all coded packets can be successfully decoded.

Fig. 12(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 node number varies in the network. Different from DCAR, both CFCR and DGCDR 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 DGCDR. Furthermore, compared to CFCR, DGCDR 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.

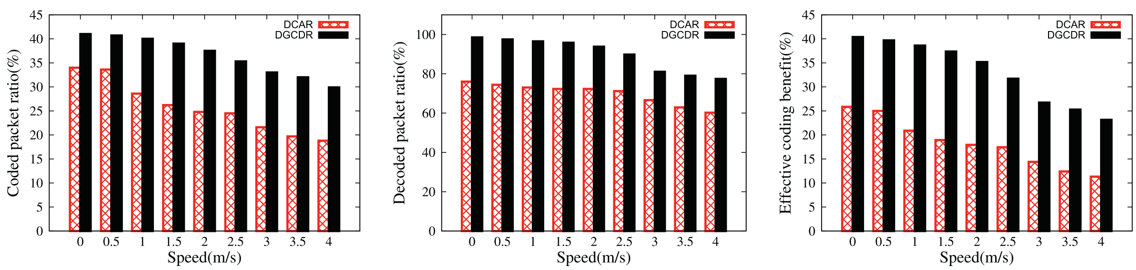
The coding benefit is determined by the product of the coded packets ratio and the decoded packets ratio. Fig. 12(c) presents the effective coding benefit of the three protocols under different APLRs.

The *average delay*, which is the end-to-end packet transmission delay between the source and the destination with various APLR, is presented in Fig. 13(a). We see that while the APLR is ascending, the average delay of the three protocols rises. When the APLR is small, due to the extra process in routing discovery, CFCR and DGCDR have larger average delay than DCAR. Also, because DGCDR selects the high quality link for transmission, it has lower average delays than CFCR. When the APLR is over 3.4%, with the coding confirmation and link quality consideration, DGCDR has the least delay.



(a) Average delay (b) Average throughput (c) Average queue size

Fig. 13. Delay, throughput, queue size with different PLRs.



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

Fig. 14. Effective coding benefit with different speeds.

The *average throughput*, which reflects the average rate of the messages delivered over a random route, is exhibited under different APLR in Fig. 13(b). We can see that when the APLR is low, DGCDR has a slight advantage. When the packet loss ratio grows, DGCDR has a larger superiority compared to the other two protocols. The phenomenon is also due to the consideration of 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 APLR turns higher, the average throughput of CFCR becomes the lowest.

The *queue size*, which indicates the number of waiting packets, is presented in Fig. 13(c). When APLR is at a low level, DGCDR has the biggest queue size due to the requirements of coding opportunity overhearing and multi-flow interference avoidance. Because the high quality wireless link can decrease the number of re-transmissions, the queue size of DGCDR becomes the smallest, when APLR ascends.

From the above analysis, we can see: a) As APLR grows, DGCDR has the largest effective coding benefit and average throughput, and the smallest average delay and queue size; b) The packet loss ratio impacts network performance at different levels, thus, the link quality should be considered in the routing protocol.

6.3.2. Mobility

Similar to packet loss ratio, we analyze the impact of mobility from the four aspects: coding benefit, delay, throughput, and queue size. However, since CFCR mainly focuses on routing in wireless mesh networks, and does not consider mobility sufficiently, we only compare the performance of DCAR and DGCDR in this part.

Figs. 14 (a), 14(b), and 14(c) respectively exhibit the variation trend of coded packet ratio, decoded packet ratio, and coding benefit, with different speeds. We can find that while the speed ascends, the values of these three parameters descend. Due to the greedy decoding policy, DGCDR can find more coding opportunities and has a smaller descending range of coded packet ratio than DCAR in Fig. 14(a). On the other hand, since multiple participating nodes introduce greater failure risk, DGCDR has a slightly larger descending range of decoded packet ratio than DCAR in Fig. 14(b), even if DGCDR can avoid multi-flow interference effectively. Synthesizing the above two parameters, as shown in Fig. 14(c), the effective coding benefit of DGCDR is larger but declines a bit faster than that of DCAR. It means that DGCDR has better network performance than DCAR in low mobility environments.

The average delay is presented with various speeds in Fig. 15(b). While the speed grows, the average delay becomes higher. On account of an extra routing confirmation process for avoiding multi-flow interference, DGCDR has a higher start point in the curve of average delay. With the speed increasing, due to the choice of high quality wireless links, the average delay of DGCDR turns lower than that of DCAR from about 0.4 m/s.

The average throughput decreases while the speed increases in Fig. 15(a). We can find that DGCDR always has a higher average throughput than DCAR, and the gap between them becomes larger when speed grows. The reason is that DGCDR gains more effective coding benefit by the greedy decoding policy, the greedy aggregation mechanism, and the greedy coding decoding algorithm.

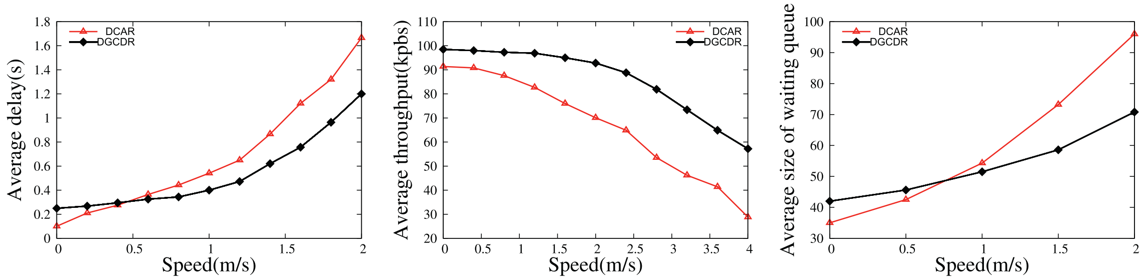
The queue size becomes bigger as the speed rises in Fig. 15(c). When speed is low, DGCDR has more waiting packets in queue for avoiding multi-flow interference. Due to the greedy aggregation mechanism, the queue size of DGCDR becomes lower than that of DCAR, when speed ascends.

By the previous analysis, we can find: a) As speed rises, DGCDR has a larger effective coding benefit and average throughput, and smaller average delay and queue size; b) The novel designs in our routing protocol, e.g., greedy decoding policy, greedy aggregation mechanism, and greed coding algorithm, can weaken the undesirable effect of mobility on network performance.

6.3.3. Average flow rate

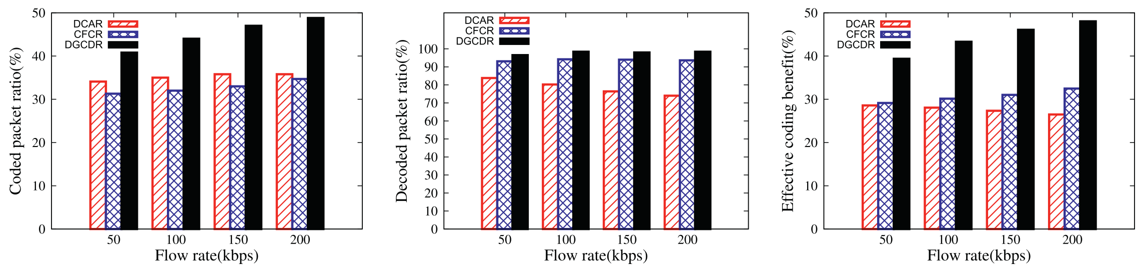
In this part, we focus on the influence of average flow rate from three aspects: average delay, average throughput, and queue size.

Figs. 16 (a), 16(b), and 16(c) respectively present coded packet ratio, decoded packet ratio, and coding benefit, with different average flow rate. We can observe that when the average flow rate increases, the values of those three parameters are stable in DCAR and CFCR. In DGCDR, the coded packet ratio and the coding ben-



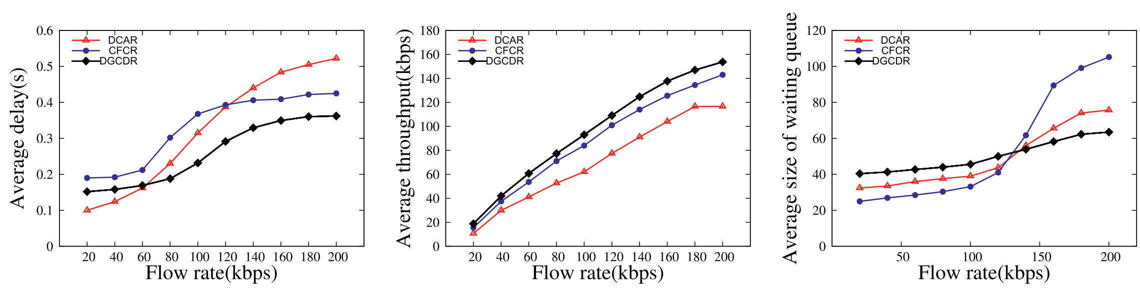
(a) Average delay (b) Average throughput (c) Average queue size

Fig. 15. Delay, throughput, queue size with different speeds.



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

Fig. 16. Effective coding benefit with different flow rates.



(a) Average delay (b) Average throughput (c) Average queue size

Fig. 17. Delay, throughput, queue size with different flow rates.

efit rise, and the decoded packet ratio remains stable. The reason is that our greedy aggregation mechanism allows native packets to be sent without having to wait for coding. When the average flow rate ascends, native packets are easier to meet the matched packets to be coded together.

The average delay is reflected with different average flow rate in Fig. 17(a). When the average flow rate grows, the average delay becomes higher. Due to the extra routing process for avoiding multi-flow interference, CFCR and DGCDR have a higher start point. Compared with CFCR, due to the greedy aggregation mechanism, the average delay of DGCDR is lower. With the average flow rate increasing, due to the multi-flow interference, the average delay of DCAR becomes higher than that of CFCR and DGCDR.

The average throughput rises while the average flow rate grows in Fig. 17(b). We can find that DGCDR always has the highest average throughput. Compared with DCAR and CFCR, the gaps become larger when average flow rate ascends. This is because DGCDR obtains more effective coding benefit by the greedy decoding policy, the greedy aggregation mechanism, and the greedy coding decoding algorithm.

The queue size grows with average flow rate increasing in Fig. 17(c). When average flow rate is low, DGCDR has the biggest number of waiting packets in queue to confirm the multi-flow interference. Due to the greedy aggregation mechanism and greedy coding and decoding algorithm, the queue size of DGCDR becomes the smallest, when average flow rate ascends.

By the previous analysis, we can find: a) As average flow rate grows, DGCDR has the largest effective coding benefit and average throughput, and smallest average delay and queue size; b) The confirmation process for avoiding multi-flow interference introduces some extra performance overhead, but it can improve network performance markedly when the traffic has high pressure. c) The greedy aggregation mechanism and the greedy coding algorithm make the packet delivery more flexible, and improve the network performance.

7. Conclusion

To improve the performance of deterministic coding-aware routing in wireless networks, we introduced a DGCDR protocol which focuses on the multi-flow environment. DGCDR can explore the coding opportunities of multiple intersecting flows by the decoding policy and the coding condition. To protect the availability of the coding opportunites, we analyze the multi-flow interference issue and describe the implementation of DGCDR.

Acknowledgment

This research was supported in part by the Key Laboratory of Aerospace Information Security and the 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, 61572380, U1536204, the Major State Basic Research Development Program of China under Grant No. 2014CB340600, and the National High Technology Research and Development Program (863 Program) of China under Grant No. 2014BAH41B00.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at 10.1016/j.comnet.2016.05.027

References

[1] A. Khreishah, I.M. Khalil, J. Wu, Distributed network coding-based opportunistic routing for multicast, in: Proceedings of ACM MobiHoc, 2012.

[2] X. Yin, Z. Li, X. Wang, A matroid theory approach to multicast network coding, in: Proceedings of IEEE INFOCOM, 2014.

[3] H. Khamfroush, D.E. Lucani, P. Pahlevani, F.H.P. Fitzek, Relay-assisted network coding multicast in the presence of neighbours, in: Proceedings of European Wireless Conference, 2015.

[4] M. Chen, Y. Qian, S. Mao, W. Tang, X. Yang, Software-defined mobile networks security, ACM/Springer Mobile Networks and Applications, 2016.

[5] S. Sengupta, S. Rayanchu, S. Banerjee, An analysis of wireless network coding for unicast sessions: The case for coding-aware routing, in: Proceedings of IEEE INFOCOM, 2007.

[6] P. Pahlevani, D.E. Lucani, M.V. Pedersen, F.H.P. Fitzek, Playncool: Opportunistic network coding for local optimization of routing in wireless mesh networks, in: Proceedings of IEEE Globecom Workshops, 2013.

[7] J. Chen, T. Li, R. Du, Efficient reliable opportunistic network coding based on hybrid flow in wireless network, in: China Communications, 2011, pp. 125–131.

[8] K. He, J. Chen, R. Du, Q. Wu, G. Xue, X. Zhang, Deypos: Deduplicatable dynamic proof of storage for multi-user environments, IEEE Transactions on Parallel and Distributed Systems, 2016.

[9] A. Khreishah, I.M. Khalil, 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, 2013.

[10] D. Koutsonikolas, C. Wang, Y. Hu, Ccack: Efficient network coding based opportunistic routing through cumulative coded acknowledgments, in: Proceedings of IEEE INFOCOM, 2010.

[11] S. Wang, A. Vasilakos, H.B. Jiang, X.Q. Ma, W.Y. Liu, K. Peng, B. Liu, Y. Dong, Energy efficient broadcasting using network coding aware protocol in wireless ad hoc network, in: Proceedings of IEEE ICC, 2011a.

[12] J. Wang, J. Wang, K. Lu, Y. Qian, B. Xiao, N. Gu, Optimal design of linear network coding for information theoretically secure unicast, in: Proceedings of IEEE INFOCOM, 2011b.

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

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

[15] H. Khamfroush, D.E. Lucani, J. Barros, Geocode: A geographic coding-aware communication protocol, in: Proceedings of IEEE Conference on Intelligent Transportation Systems (ITSC), 2011.

[16] M. Chen, Y. Hao, Y. Li, C. Lai, D. Wu, On the computation offloading at ad hoc cloudlet: Architecture and service models, IEEE Communications, 2015.

[17] A. Khreishah, I.M. Khalil, J. Wu, Universal opportunistic routing scheme using network coding, in: Proceedings of IEEE SECON, 2012.

[18] S. Wang, G. Tan, Y. Liu, H. Jiang, T. He, Coding opportunity aware backbone metrics for broadcast in wireless networks, in: Proceedings of IEEE INFOCOM, 2013.

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

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

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

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

[23] A. Asterjadhi, E. Fasolo, M. Rossi, J. Widmer, M. Zorzi, Toward network coding-based protocols for data broadcasting in wireless ad hoc networks, IEEE Transactions on Wireless Communications, 2010.

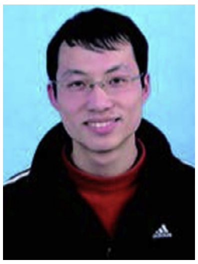
[24] J. Chen, K. He, R. Du, M. Zheng, Y. Xiang, Q. Yuan, Dominating set and network coding-based routing in wireless mesh networks, IEEE Transactions on Parallel and Distributed Systems, 2016.

[25] B. Guo, H.K. Li, C. Zhou, Y. Cheng, Analysis of general network coding conditions and design of a free-ride-oriented routing metric, IEEE Transactions on Vehicular Technology, 2011.

[26] J. Chen, K. He, Q. Yuan, R. Du, J. Wu, Mucar: A greedy multi-flow-based coding-aware routing in wireless networks, in: Proceedings of IEEE SECON, 2015.

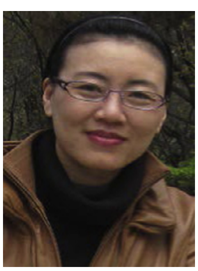
[27] P. Li, S. Guo, S. Yu, A.V. Vasilakos, Codepipe: An opportunistic feeding and routing protocol for reliable multicast with pipelined network coding, IEEE Transactions on Wireless Communications, 2012.

Jing Chen received the Ph.D. degree in computer science from Huazhong University of Science and Technology, Wuhan. He is currently a full professor at Computer School, Wuhan University. His research interests in computer science are in the areas of cloud security, network security. He has published more than 80 research papers in many international journals and conferences, such as IEEE Transactions on Computers, IEEE Transactions on Parallel and Distributed Systems, International Journal of Parallel and Distributed System, INFOCOM, SECON, TRUSTCOM, NSS. He acts as a reviewer for many Journals and conferences, such as IEEE Transactions on Information Forensics & Security, IEEE Transactions on Wireless Communication, IEEE Transactions on Industrial Informatics, Computer Communications, and GLOBCOM.



Kun He is a PH.D student of Wuhan University. His research interests include cryptography, network security, mobile computing, and cloud computing. He has published research papers in IEEE Transactions on Parallel and Distributed Systems, IEEE Transactions on Computers, International Journal of Communication Systems, Security and Communication Networks, and IEEE TRUSTCOM.

Quan Yuan is an Assistant Professor in the Department of Math and Computer Science, University of Texas-Permian Basin, TX, USA. His research interests include Mobile Computing, Routing Protocols, Peer-to-Peer computing, Parallel and Distributed Systems, and Computer networks. He has published more than 30 research papers in many international journals and conferences, such as IEEE Transactions on Parallel and Distributed Systems, INFOCOM, MobiHoc, SECON, TRUSTCOM.



Ruiying Du received the BS, MS, PH.D degrees in computer science in 1987, 1994 and 2008, from Wuhan University, Wuhan, China. She is a full professor at computer school, Wuhan University. Her research interests include network security, wireless network, cloud computing and mobile computing. She has published more than 80 research papers in many international journals and conferences, such as IEEE Transactions on Parallel and Distributed System, International Journal of Parallel and Distributed System, INFOCOM, SECON, TRUSTCOM, NSS.

Jie Wu (F'09) is the chair and a Laura H. Carnell professor at Temple University. His research interests include cloud computing and information security. His publications include more than 600 papers in scholarly journals, conference proceedings, and books. He has served on several editorial boards, including IEEE Transactions on Computers and the Journal of Parallel and Distributed Computing. He was the general cochair for IEEE MASS 2006, IEEE IPDPS2008, and IEEE DCOSS 2009 and was the program cochair for IEEE INFOCOMM 2011. He served as the general chair for IEEE ICDCS 2013. He was an IEEE Computer Society distinguished visitor and was chair for the IEEE Technical Committee on Distributed Processing (TCDP). He is currently an ACM distinguished speaker and is a fellow of the IEEE.