Abstract—Opportunistic routing and network coding have been considered as effective strategies for improving the throughput of Wireless Mesh Networks (WMN). However, most existing work studied opportunistic routing and network coding separately. This has largely limited the ability of the above strategies from effectively improving the network performance. To achieve improved network throughput, in this paper, we propose a coding-aware opportunistic routing mechanism for WMNs. The design goal is achieved by effectively integrating the above two strategies such that decision on each packet forwarding is made with the awareness of potential coding opportunities. Simulation results show that our proposed mechanism can remarkably improve the network throughput.

Keywords- Opportunistic routing, network coding, wireless mesh networks

I. INTRODUCTION

Design of routing protocols for Wireless Mesh Networks (WMN) needs to consider the existence of low quality wireless links in such networks. Traditional routing protocols typically select a fixed path connecting source and destination. Each intermediate node selects its only next hop along the discovered path. Such strategy works well in reliable networks. However, it may not be the ideal choice for WMNs wherein links are lossy. This is because such fixed routing strategies are likely to trigger more retransmissions at the MAC layer. Opportunistic routing [1]-[4] has recently been proposed as an effective strategy to obtain high throughput in wireless networks with lossy links. Opportunistic routing exploits the broadcast nature of wireless medium. With opportunistic routing, each packet holder chooses a subset of its neighboring nodes as the potential next hops. Among those nodes (a) in the pre-determined next-hop candidates and (b) receiving the packet, the one leading to the best route to the destination will be chosen as the next sender. Opportunistic routing can reduce the retransmission rate greatly and exploits those long but low-quality links to reduce the total number of transmissions for a packet to reach its destination.

Network coding [8]-[22] has recently been another promising approach for increasing network throughput. The benefit of network coding in WMNs also comes from the inherent broadcast nature of wireless medium. Fig. 1 gives a typical example for illustrating how network coding can reduce the total number of transmissions. In Fig. 1, nodes A and B want to exchange packets via a common relay node R. A sends packet \( p_1 \) to B and B sends packet \( p_2 \) to A. After R receives both \( p_1 \) and \( p_2 \), it will broadcast a new packet “\( p_1 \text{ xor } p_2 \)”. Upon the receipt of the xor-ed packet, both A and B can then decode their interested packet. The number of transmissions in this example is reduced from four to three.

However, most existing network coding mechanisms in the literature passively wait for the appearance of coding opportunities. The passive strategy in these mechanisms largely limits their ability in achieving high network throughput. How to effectively combine network coding and localized unicasting still remains open.

In this paper, we propose a new coding-aware opportunistic routing mechanism, which actively creates more coding opportunities by fully utilizing the local state information kept at nodes. Our mechanism design is based on an observation such that further coding gain is obtainable if decision on packet forwarding can be made with the awareness of coding opportunities. Accordingly, our mechanism effectively combines practical inter-flow network coding and opportunistic routing. Our mechanism is localized and its implementation does not require synchronization (either local or global) among nodes. Simulation results demonstrate that our mechanism can improve the network performance for WMNs with multiple concurrent flows as compared with related work.

The rest of this paper is organized as follows. In Section II, we briefly review related work. In Section III, we present the detailed design description of the proposed mechanism. In Section IV, we conduct detailed simulations to evaluate the performance of our mechanism by comparing it with related work. In Section V, we conclude this paper.


## II. RELATED WORK

In this section, we briefly review related work in opportunistic routing and network coding, respectively.

Recently, several opportunistic routing schemes have been proposed. The extremely opportunistic routing protocol (ExOR) [1] introduces the concept of being opportunistic when wireless links are lossy. However, ExOR requires tight node coordination, which is difficult to be implemented in large networks. Besides ExOR, some other work [2]-[4] has been proposed in this area and this work mainly focused on how to maximize the gain in opportunistic packet forwarding. In [5]-[7], the actual efficacy of opportunistic routing has been investigated by developing various analytical models.

Network coding was first proposed by Ahlswede et al. [8] and it was initially targeted at tackling the multicast issue. Much existing work (e.g., [9]-[12]) has been focused on the construction of efficient network coding codes. In [9], Li et al. proved that linear code is enough to achieve the multicast capacity. Koetter et al. [10] proposed polynomial time matrix multiplication and inversion algorithms that can be used for encoding and decoding. Some recent work [13]-[15] has studied the design of network coding mechanisms for wireless environment.

Recently, some work [16]-[19] has been focused on providing efficient unicast communications by using network coding. The COPE mechanism [19] considers WMNs with multiple unicast flows and it shows that network coding can increase the throughput of WMNs. However, COPE utilizes coding opportunities passively, which limits its ability in achieving high performance. In [20][21], the authors proposed centralized network coding aware routing schemes for maximizing the overall coding opportunities in the whole network by using linear programming, which is known to have the scalability issue and is difficult to be employed in dynamic large networks. The MAC-independent Opportunistic Routing & Encoding (MORE) protocol [22] adopts a different strategy for supporting coding-aware routing such that intra-flow random network coding is embedded into opportunistic routing. MORE achieves high performance in terms of reliable transmissions. However, MORE has not considered the issue of how to increase coding gains when multiple flows are available.

## III. PROPOSED MECHANISM

### A. Observation

Before providing the detailed design of our mechanism, we first illustrate how extra gains are possible if decisions on packet forwarding can be made with the awareness of coding opportunities.

Consider the example network in Fig. 2. Suppose there are two flows: One from node $A$ to node $E$ and the other from node $F$ to node $A$. The shortest paths for these two flows are $A \rightarrow B \rightarrow C \rightarrow E$ and $F \rightarrow D \rightarrow B \rightarrow A$. In this shortest paths case, we can see that there exists no coding opportunity. However, if $F$ can forward its packet via $E$ to $A$, then the path from $F$ to $A$ will be $F \rightarrow E \rightarrow C \rightarrow B \rightarrow A$. Although this path has one more extra hop than the corresponding shortest path, it creates coding opportunities at nodes $C$ and $B$. The total number of packet transmissions can then be reduced. This motivates us to study how to effectively increase the coding gain, if possible, when nodes forward packets.

### B. Proposed Mechanism

In this subsection, we present the detailed design description of our Coding Aware Opportunistic Routing mechanism, which is referred to as CAOR hereafter. CAOR aims at increasing the coding gain when multiple simultaneous unicast flows exist. CAOR is essentially an opportunistic forwarding mechanism and it optimizes the selection of forwarder on a per-hop basis. In this paper, we assume there is no limitation on power or processing ability at nodes and nodes can overhear all transmissions in their neighborhood provided that no collisions occur. Next, we first define a few notations that we will use later.

**Buffer:** Each node has a buffer that stores those packets it has overheard, received, and sent out in the past $T$ seconds. We set $T=0.5$ second as done in COPE.

**Output queue:** Each node maintains a queue of packets that it needs to forward further.

**Sender:** The node that initializes a packet transmission by selecting a packet from its output queue.

**Primary Packet:** Original non-coded data packet.

**Forwarder:** A chosen next-hop of the primary packet. This chosen forwarder will become the next sender for this primary packet after it has received and decoded the primary packet.

**Forwarder set:** The set of next hop candidates for a particular sender, which can be selected as forwarder for relaying a primary packet.

CAOR contains the following three major components: Forwarder set selection, best forwarder selection, and priority-based forwarding.

#### 1) Forwarder Set Selection

Before a sender sends out a packet, it should first choose its forwarder set for this transmission. The preconditions that a node can be selected into such a forwarder set are as follows: (1) It is a direct neighboring node of the sender; (2) It is closer to the destination, measured in terms of the Expected Transmission Count (ETX) [23] value as defined next; (3) Nodes in the forwarder set must be able to mutually overhear each other. The ETX value associated with a link is the inverse...
of the delivery probability of the link. The ETX value associated with a node is its shortest distance from the node to the destination in terms of the ETX metric. Here, the length of a path is defined as the sum of ETX values of its constituent links.

When a sender is ready to send a packet, it inserts an extra header into this packet, which lists all nodes in the forwarder set. Nodes in the forwarder set are ranked according to their distances to the destination.

2) Best Forwarder Selection

A key issue in CAOR is how to select the forwarder that has the maximum coding opportunities among nodes in the forwarder set in a localized manner while keeping the protocol overhead low. The amount of coding opportunities is measured by how many native packets can be successfully delivered in a single transmission of a coded packet. For this purpose, each node in the forwarder set should be able to independently produce the same selection result based on the information that each of them locally keeps.

We first provide an example to illustrate how the best forwarder can be selected. Consider the scenario shown in Fig. 3-a. Node A wants to send a packet \( p_1 \) to node D, while node E wants to send a packet \( p_2 \) to node A. After nodes A and E finished their transmissions, nodes B and D have received \( p_2 \), and nodes B and C have received \( p_1 \) (See Fig. 3-b). Then the next step is how to select a best forwarder for \( p_1 \) to reach A and for \( p_2 \) to reach D. Traditional opportunistic routing protocols will select C as the forwarder to reach D because C is much closer to D than B. However, from Fig. 3-c, we can see that this is not the best solution since this decision will lead to two transmissions, one for packet \( p_1 \) to D and the other for packet \( p_2 \) to A. In contrast, if both B and C can realize that B has more network coding opportunities than C and let B serve as the only forwarder, only one transmission is needed (see Fig. 3-d). Thus, B is the best forwarder from the perspective of network throughput. Accordingly, if the node with the most coding opportunities can be chosen as the forwarder in each packet transmission, the throughput of network will be greatly improved as a result.

The key in forwarder selection lies in how all nodes in the forwarder set can agree on which one of them has the most coding opportunities in a localized manner. From the above example, we can see that if node C knows all the packets kept by B and each of B’s neighbors, then C can locally compute the coding chances available at B. Therefore if all nodes in a forwarder set exchange their knowledge about the packets stored in theirs and their neighbors’ buffers, then all these nodes can compute the same results on the amount of coding opportunities for a particular primary packet such that all these nodes will know which one of them is the best forwarder for this particular transmission.

The packet-specific information exchanging among nodes is carried out in an opportunistic manner with little extra overhead. COAR adopts the same opportunistic listening and periodic reception report broadcasting procedures as used in COPE. The difference is as follows. In COPE, a reception report tells neighbors which packet the report sender currently stores. However, in COAR, a reception report includes not only information regarding those packets kept at the report initiator, but also at each of the report initiator’s neighbors. This enables each node to have a two-hop local view on packet available at its neighbors and neighbors’ neighbors.

Procedures for calculating coding chances for a particular primary packet are shown in Fig. 4. Upon the receipt of a reception report from a neighbor (say \( y \)), node \( x \) can locally compute how much the coding chances at \( y \) is for each primary

![Figure 3. Example illustrating the selection of forwarder: The next-hop of each packet is shown in (a). Nodes B and C consist of the forwarder set of A to reach D. Nodes B and D consist of the forwarder set of E to reach A. (b) shows the received packets of each node after transmissions of both packets from their originating nodes, respectively. Node A can choose C as its forwarder as (c) shows, but it should pick B as the forwarder because B can maximize the overall throughput in this example.](image)
packet p in its own output queue. First, x picks packet p from y’s output queue, checks different patterns for p to be encoded with other packets (s) kept by y to obtain the maximum possible coding gain at y for p. Here, the key difference from COPE is that CAOR considers the first k packets in the y’ output queue as seeds for coding. This can greatly increase the coding chances as compared with the FIFO strategy adopted in COPE. In CAOR, both coding gain, computation overhead, and extra packet delivery latency (if any) are controllable by adjusting the value of k. Each node in a forward set will execute the same procedure to get the best forwarder regarding a specific primary packet.

One issue associated with the above forward selection process is the possible inconsistency of information kept at neighboring nodes, which may cause the suboptimal selection. This issue can be caused by report delivery latency for two-hop report dissemination, or caused by collision or congestion. To handle this, in addition to reporting packets that a node really stores in its packet buffer, each node makes guessing about packets it would receive: If a node has received M packets, numbered from \(p_{m,1}\) to \(p_{m,M}\), belong to the same flow, via either overhearing or receiving, it extrapolates that it will receive the next two packets \(p_{m,2}\) and \(p_{m,3}\) of this flow and can report these two packets in the current reception report. In our implementation, we set M to three. We make guessing about two packets because reception report can be sent with data packet, thus guessing two packets is enough for two hops delay. This guessing is reasonable for long lived flows. Second, if congestion occurs in the network, reception reports may get lost. Each node solves the problem by using the guessing mechanism in COPE, which uses the delivery probability of the link between the sender and a neighbor as the probability that this neighbor has received the packet sent by this sender.

In case guessing at some nodes is inconsistent, a coded packet may be non-decodable and decisions on the best forwarder selection for a particular packet at some nodes may diverge. Each non-decodable primary packet will be retransmitted, possibly, encoding with a new set of packets. The divergence on forwarder selection could possibly lead to redundant transmissions and degrade the network performance if not well handled. This can be solved by the priority-based forwarding process in the following subsection, which works in a way such that the node first forwards a packet wins and the rest withdraws their retransmission after overhearing the transmission. Such suboptimal forwarder selection is the price of being opportunistic and no tight coordination required among neighbors. Even suboptimal, it can still provide more coding gain than the fixed forwarder selection strategy with high probability.

In summary, our best forwarder selection has the following characteristics. First, the coding chances are calculated by considering all the packets stored in nodes’ output queue, which outperforms the FIFO manner. Second, the forwarder of each packet is selected according to the up-to-date status of concurrent flows in the network to maximize the throughput.

#### 3) Priority-Based Forwarding

After all nodes in a forwarder set compute the coding opportunities for a particular primary packet, they independently use a priority-based forwarding to minimize the synchronization requirement. Each node sets its transmission timer proportional to the reciprocal of its local coding chances for the (primary) packet. A tie will be broken by each node’s ETX metric, that is, the one with smaller ETX value to destination has higher priority in this case. So the node with more coding opportunities will forward the packet earlier, and other nodes hearing this forwarding will cancel their timers. This mechanism can also prevent the redundant transmission if forwarder selection at different nodes is inconsistent.

It may happen that a low priority node for a primary packet \(p_1\) forwards \(p_1\) before a high priority node does, due to the fact that this primary packet \(p_1\) may be encoded and sent along with another primary packet \(p_2\). In this case, the pre-computed high priority node for \(p_1\) cancels its scheduled transmission once it overhears this transmission.

### IV. SIMULATION RESULTS

In this section, we conducted simulations to evaluate the performance of CAOR using ns-2 by comparing it with COPE. In the implementation of both mechanisms, we used 802.11 at the MAC layer and enabled the promiscuous mode to implement the pseudo-broadcast in COPE. 200 static nodes were placed in a 1000×1000 m² square field. UDP flows were generated randomly. The radio range is approximately 100 meters. In COPE, Greedy Perimeter Stateless Routing (GPSR) [24] is used as the underlying geographical routing protocol for packet forwarding. We evaluated the performance of both protocols in terms of network throughput. Only data packets are considered in the computation of throughput. Each result was averaged over five random network topologies. All random topologies were generated by the setdest tools in ns-2.

#### A. Impact of Flow Number

In this test, we varied the number of flows in the network. The source and destination of each flow was chosen randomly. Fig. 5 shows that the network throughput due to both protocols versus flow number. From the figure, it is seen that CAOR can improve the network throughput substantially as compared with COPE. On average, CAOR provides 15 percent...
throughput gain over COPE, and with a maximum ratio of 28 percent. When flow number is small, the probability that paths for different flows coincide is small. Thus CAOR performs similarly to COPE. As the number of flows grows, CAOR creates much more coding opportunities than COPE. When the number of flows continues to grow, however, the performance of both mechanisms becomes similar again. That because as flow number further increases, the probability of flows could meets is increased at the same time, so there is little room for CAOR to increase the coding opportunities. Also both mechanisms deteriorate by the high level of congestion when the number of concurrent flows grows.

B. Impact of Path Length

Next, we evaluated the impact of path length on network throughput. In this test, we fixed the total number of flows to 20. We varied the geometrical distance between the source and the destination of each flow. Fig. 6 shows that (1) network throughput due to either protocol drops with path length since longer path consumes more network resources, and (2) the performance of CAOR drops much slower than that of COPE. This is because the more coding opportunities in CAOR created by the growing of path length can largely alleviate the negative impact of congestion so that less steep performance degradation is observed. From the figure, it is seen that CAOR can provide up to 60 percent throughput gain over COPE.

V. CONCLUSION

In this paper, we proposed a novel localized coding-aware opportunistic routing mechanism to increase the throughput of wireless mesh networks. With our mechanism, packet forwarding is carried out with the awareness of coding opportunities and no synchronization is required among nodes. Simulation results demonstrate that our mechanism achieves high performance in increasing the network throughput. Our future work will focus on the design of coding-and-congestion aware routing mechanism for wireless mesh networks.

ACKNOWLEDGMENT

This work was supported in part by High-Tech Research and Development Program of China under Grant No. 2006AA01Z207-1, the Knowledge Innovation Program of the Chinese Academy of Sciences under Grant No. KGJCX2-YW-110-6 and NSFC grant (Grant No. 60702074).

REFERENCES

