

A Review On COPE Architecture in Multi-hop Wireless Network

Chintan Thakor¹, Ms. Rachna Patel²

¹Computer Engineering Department, Uka Tarsadia University,

²Computer Engineering Department, Uka Tarsadia University.

Abstract — COPE is a new architecture for wireless networks. It uses the concept of network coding in a packet network where data is divided into packets. Network coding is applied to the contents of packets. In addition to forwarding packets, routers mix packets from different sources to increase the information content of each transmission. It inserts a coding scheme between the IP and MAC layers. It shows that intelligently mixing packets increases network throughput. Prior work on network coding is mainly theoretical and focuses on multicast traffic. This technique has several important benefits such as an increase in throughput and an improvement in packet delivery ratio, delay, reliability of the network.

Keywords- COPE, Network Coding, Multi-hop Wireless Networks, throughput, delay, packet delivery ratio

I. INTRODUCTION

COPE is a new forwarding architecture that substantially improves the throughput of wireless networks. COPE inserts a coding scheme between the IP and MAC layers, which identifies coding opportunities and benefits from them by forwarding multiple packets in a single transmission.

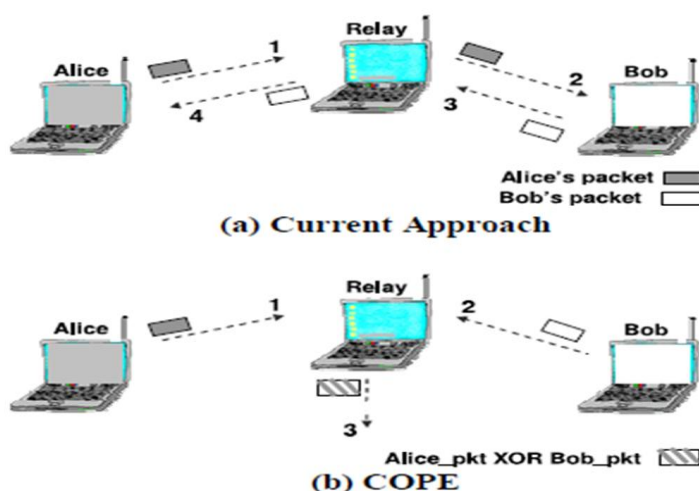


Figure 1: A simple example of how COPE increases the throughput ^[1]

To give the reader a feel for how COPE works, we start with a fairly simple example. Consider the scenario in Figure 1, where Alice and Bob want to exchange a pair of packets via a router. In current approaches, Alice sends her packet to the router, which forwards it to Bob, and Bob sends his packet to the router, which forwards it to Alice. This process requires 4 transmissions. Now consider a network coding approach. Alice and Bob send their respective packets to the router, which XORs the two packets and broadcast the XOR-ed version. Alice and Bob can obtain each other's packet by XOR-ing again with their own packet. This process takes 3 transmissions instead of 4. Saved transmissions can be used to send new data, increasing the wireless throughput.

In fact, COPE leads to larger bandwidth savings than are apparent from this example. It exploits the shared nature of the wireless medium to broadcast each packet in a small neighborhood around its path. Each node stores the overheard packets for a short time. It also tells its neighbors which packets it has heard by annotating the packets it sends. When a node transmits a packet, it uses its knowledge of what its neighbors have heard to perform opportunistic coding. The node XORs multiple packets and transmits them as a single packet if each intended next hop has enough information to decode the encoded packet. This extends COPE beyond two flows that traverse the same nodes in reverse order (as in the Alice-and-Bob example), and allows it to XOR more than a pair of packets.

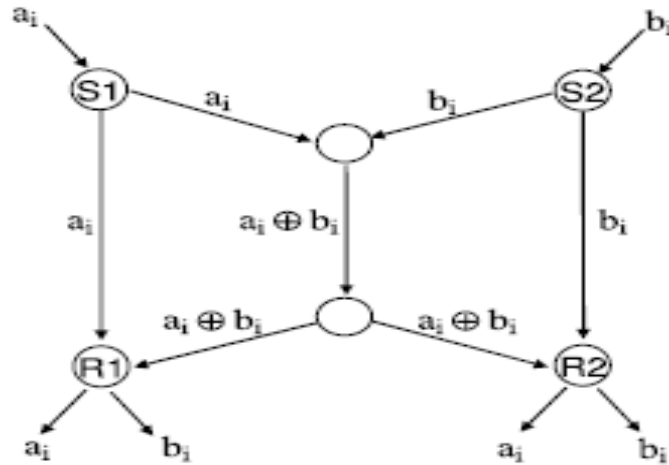


Figure 2: A simple scenarios showing how network coding improves throughput. ^[1]

The idea underlying network coding is usually illustrated using the famous butterfly example. Consider the network in Figure 2, where source S1 wants to deliver the stream of messages a_i to both R1 and R2, and source S2 wants to send the stream of messages b_i to the same two receivers. Assume all links have a capacity of one message per unit of time. If routers only forward the messages they receive, the middle link will be a bottleneck, which for every time unit, can either deliver a_i to R1 or b_i to R2. In contrast, if the router feeding the middle link XORs the two messages and sends $a_i \oplus b_i$ (or any linear combination of a_i and b_i), as shown in the figure, both receivers obtain two messages in every time unit. Thus, network coding is allowing the routers to mix the bits in forwarded messages can increase network throughput.

1.1 COPE's design is based on two key principles

- COPE disposes of the point-to-point abstraction and embraces the broadcast nature of the wireless channel. Network designers typically abstract the wireless channel as a point-to-point link, and then adapt forwarding and routing techniques designed for wired networks for wireless. In contrast, COPE exploits the broadcast property of radios instead of hiding it under an artificial abstraction [1].
- COPE employs network coding. Our work is rooted in the theory of network coding, which allows the routers to mix the information content in the packets before forwarding them. Prior work on network coding is mainly theoretical and focuses on multicast traffic [1].

II. OVERVIEW OF COPE

We introduce COPE, a new forwarding architecture for wireless networks. It inserts a coding layer between the IP and MAC layers, which detects coding opportunities and exploits them to forward multiple packets in a single transmission. COPE incorporates three main techniques:

2.1 Opportunistic listening:

Wireless is a broadcast medium, creating many opportunities for nodes to overhear packets when they are equipped with omni-directional antennae. COPE sets the nodes in promiscuous mode, makes them snoop on all communications over the wireless medium and store the overheard packets for a limited period T (the default is $T = 0.5s$) [1].

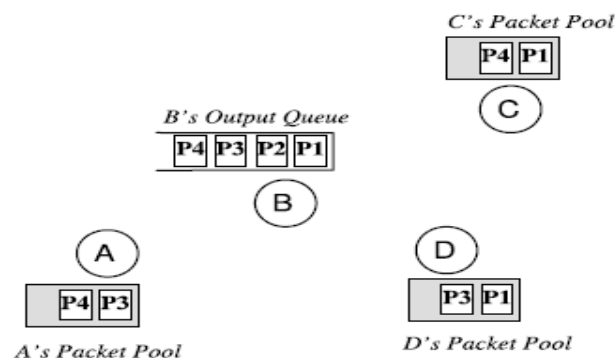


Figure 3(a): B can code packets it wants to send ^[1]

In addition, each node broadcasts reception reports to tell its neighbors which packets it has stored. Reception reports are sent by annotating the data packets the node transmits. A node that has no data packets to transmit periodically sends the reception reports in special control packets.

2.2 Opportunistic coding

The key question is what packets to code together to maximize throughput. A node may have multiple options, but it should aim to maximize the number of native packets delivered in a single transmission, while ensuring that each intended next hop has enough information to decode its native packet. The above is best illustrated with an example. In Figure 3(a), node B has 4 packets in its output queue p_1, p_2, p_3 , and p_4 . Its neighbors have overheard some of these packets. The table in Figure 3(b) shows the next hop of each packet in B's queue. When the MAC permits B to transmit, B takes packet p_1 from the head of the queue.

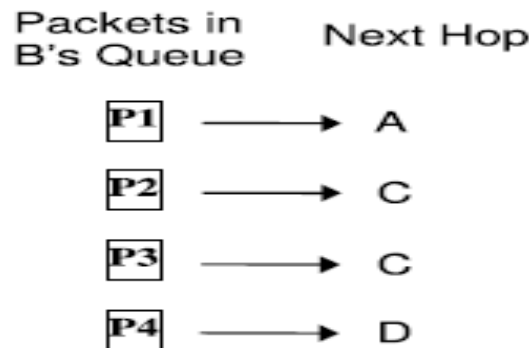


Figure 3(b): Next hops of packets in B's queue ^[1]

Assuming that B knows which packets each neighbor has, it has a few coding options as shown in Figure 3(c). It could send $p_1 \oplus p_2$. Since node C has p_1 in store, it could XOR p_1 with $p_1 \oplus p_2$ to obtain the native packet sent to it, i.e., p_2 . However, node A does not have p_2 , and so cannot decode the XOR-ed packet. Thus, sending $p_1 \oplus p_2$ would be a bad coding decision for B, because only one neighbor can benefit from this transmission. The second option in Figure 3(c) shows a better coding decision for B. Sending $p_1 \oplus p_3$ would allow both neighbors C and A to decode and obtain their intended packets from a single transmission. Yet the best coding decision for B would be to send $p_1 \oplus p_3 \oplus p_4$, which would allow all three neighbors to receive their respective packets all at once.

Coding Option	Is it good?
P1 + P2	Bad Coding (C can decode but A can't)
P1 + P3	Better Coding (Both A and C can decode)
P1 + P3 + P4	Best Coding (Nodes A, C, and D can decode)

Figure 3 (c): Possible coding options ^[1]

The above example emphasizes an important coding issue. Packets from multiple unicast flows may get encoded together at some intermediate hop. But their paths may diverge at the next hop, at which point they need to be decoded. If not, unneeded data will be forwarded to areas where there is no interested receiver, wasting much capacity [1]. The coding algorithm should ensure that all next hops of an encoded packet can decode their corresponding native packets. This can be achieved using the following simple rule: To transmit n packets, $p_1, \dots, p_n$, to n next hops, $r_1, \dots, r_n$, a node can XOR the n packets together only if each next-hop r_i has all $n - 1$ packets p_j for $j \neq i$. This rule ensures that each next hop can decode the XOR-ed version to extract its native packet. Whenever a node has a chance to transmit a packet, it chooses the largest n that satisfies the above rule to maximize the benefit of coding.

2.3 Learning neighbor state:

But how does a node know what packets its neighbors have? As explained earlier, each node announces to its neighbors the packets it stores in reception reports. However, at times of severe congestion, reception reports may get lost in

collisions, while at times of light traffic, they may arrive too late, after the node has already made a suboptimal coding decision [1]. Therefore, a node cannot rely solely on reception reports, and may need to guess whether a neighbor has a particular packet. Occasionally, a node may make an incorrect guess, which causes the coded packet to be undecodable at some next hop. In this case, the relevant native packet is retransmitted, potentially encoded with a new set of native packets [1].

III. PACKET CODING AND PRIORITY SCHEDULING ALGORITHM

3.1 Packet coding algorithm:

The packet coding scheme inside COPE is one of key components for achieving high transmission efficiency. Each node maintains one output queue, two virtual queues $Q_{i,1}$ and $Q_{i,2}$ (one for small packets and the other for large packets) for each neighbor i as shown in Figure 4, and a hash table that is keyed on packet-id [2]. For each packet in the output queue, the table indicates the probability of each neighbor having that packet. When a new packet is added to the output queue, an entry pointing to this packet is added to some virtual queue which is determined by the packet's next hop and the packet size [2].

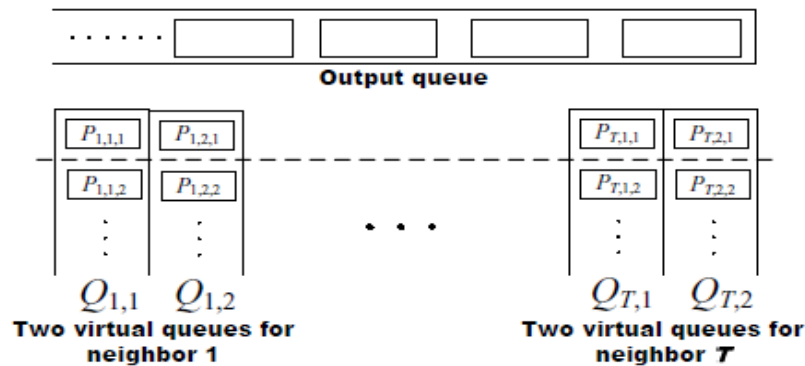


Figure 4: Packet coding framework in COPE ^[2]

In following rules used in COPE, Rule 1 are used for determining the candidate packets and Rule 2 and 3 are used for selecting packets $P_{i2}, P_{i3}, \dots, P_{iL}$ from candidate packets.

- 1) To limit packet reordering, only the packets at the heads of the virtual queues are candidates to encode with $P_{i1}[2]$.
- 2) Ensure that each neighbor to whom a packet $P_i \in \{P_{i1}, P_{i2}, P_{i3} \dots P_{iL}\}$ is headed has a large probability (greater than a threshold G) of retrieving this native packet P_i [2].
- 3) COPE gives preference to XORing packets of similar lengths, because XORing small packets with larger ones reduce bandwidth savings. Packets headed to the same next hop will never be encoded together, since the next hop will not be able to decode them [2].

3.2 Packet decoding:

Packet decoding is simple. Each node maintains a Packet Pool, in which it keeps a copy of each native packet it has received or sent out. The packets are stored in a hash table keyed on packet id, and the table is garbage collected every few seconds. When a node receives an encoded packet consisting of n native packets, the node goes through the ids of the native packets one by one, and retrieves the corresponding packet from its packet pool if possible. Ultimately, it XORs the $n - 1$ packets with the received encoded packet to retrieve the native packet meant for it [1].

3.3 Priority scheduling for packets:

Priority scheduling gives high priority to control packets. It maintains control packets and data packets in separate queues in FIFO order. Currently, this scheme is used in most comparison studies about mobile ad hoc networks. The control queue has higher priority than data queues. Among data queues, we experiment with various scheduling algorithms

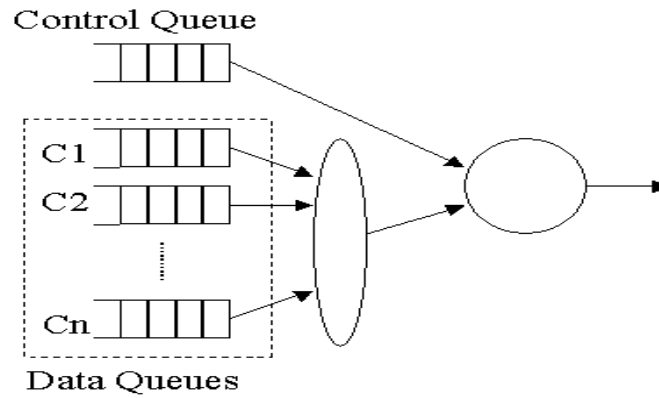
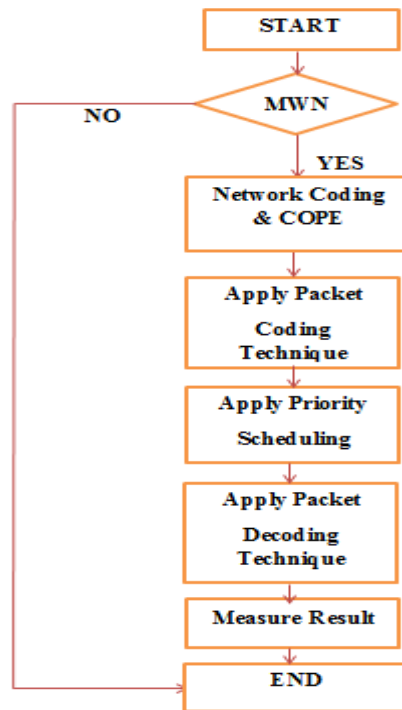


Figure 5: Packet scheduler. ^[8]

IV PROPOSED TECHNIQUE



V. SIMULATION RESULT

NS-2 is the most popular and powerful simulator. NS-2 is an object-oriented discrete time event simulator and its modular design made it to be extensible. The simulator chosen for this project is the network simulator NS-2, version 2.34. NS 2 is a discrete event simulator targeted at networking research which provides support for simulation of TCP in multi-hop wireless network.

Table 1: Simulation Parameters.

Parameters	Value
Channel Type	Channel/Wireless Channel
MAC Type	Mac/802.11
Max packet in queue	50
Number of nodes	50
Number of source nodes	5,10,15,20
Routing protocol	AOV and DSR
X dimension of topography	2588
Y dimension of topography	680
Simulation time	150s

Here we simulate AODV and DSR routing protocol with 5,10,15,20 source nodes in multi-hop wireless network.

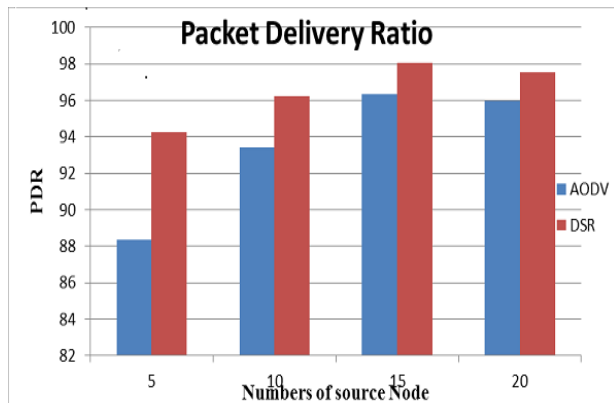


Figure 6: Packet Delivery Ratio of AODV and DSR Routing Protocol.

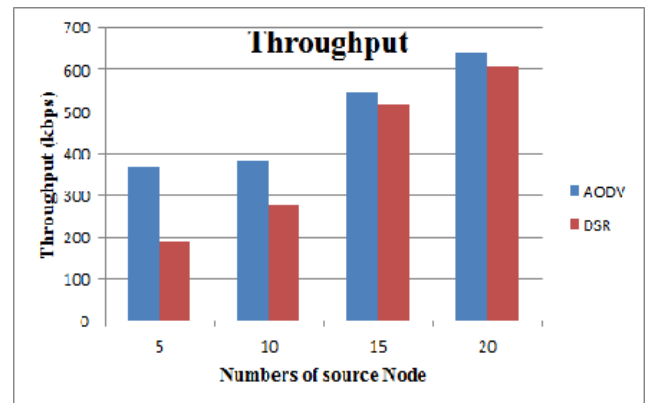


Figure 7: Throughput Measurement of AODV and DSR Routing Protocol.

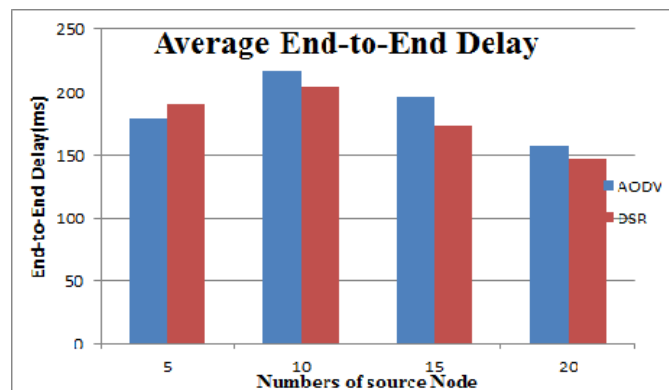


Figure 8: Average End-to-End Delay of AODV and DSR Routing Protocol.

VI CONCLUSION AND FUTURE SCOPE

6.1 Conclusion

Using the new flow oriented queuing structure COPE which can increase potential coding opportunities. It is convenient for allocation of priorities to packet. The network coding technique generalizes the traditional routing approach by allowing the intermediate network nodes to create new packets by combining the packets received over their incoming edges. This technique has several important benefits such as an increase the throughput, packet delivery ratio (PDR), decrease end-to-end delay, save bandwidth and an improvement in the reliability and robustness of the network.

We vary the number of source nodes from 5 to 20 in a fixed topography of 2588*680 meters and the performance varies widely across different number of Source nodes. As far as Throughput is concerned, AODV perform better than DSR. If number of source nodes increase, average end-to-end delay and Packet delivery ratio of AODV is slightly increase then DSR.

6.2 Future Scope

We will measure the performance of COPE architecture for throughput, average end-to-end delay, packet delivery ratio and compare the result with AODV and DSR protocol.

ACKNOWLEDGMENT

I would like to thank Prof. Rachna M. Patel for their guidance in my dissertation research.

REFERENCES

- [1] Y. Sachin Katti, Hariharan Rahul, Wenjun Hu, Dina Katabi, Muriel M'edard and Jon Crowcroft, "XORs in The Air: Practical Wireless Network Coding", IEEE, pp. 497 – 510, June 2008.
- [2] Kaikai Chi, Xiaohong Jiang, and Susumu Horiguchi, "A General Packet Coding Scheme for Multi-hop Wireless Networks", IEEE GLOBECOM, pp. 1339-1343, 2007.

- [3] Byung-Gon Chun and Mary Baker, "Evaluation of Packet Scheduling Algorithms in Mobile Ad Hoc Networks", ACM SIGMOBILE, Volume 6, Number 3, pp. 36-49., July 2002.
- [4] Fubao Yang, Shengzhi Ling, Hui Xu and Baolin Sun, "Network Coding-based AOMDV Routing in MANET", IEEE International Conference on Information Science and Technology, Pages 337 – 340, March 23-25, 2012.
- [5] J Baolin Sun, Xiaocheng Lu, Chao Gui, Ying Song, and Hua Chen, "Network Coding-Based On-Demand Multipath Routing in MANET", IEEE 26th International Parallel and Distributed Processing Symposium Workshops & PhD Forum, pp. 1520 – 1524, 2012.
- [6] Ravi Kumar Bansal, "Performance Analysis of Cluster based Routing Protocol in MANETs", Computer Science and Engineering Department thapar Institute Of Engineering and Technology, pp. 370-375, 2006.
- [7] <http://mimo-mesh.com/media/ray.pdf>.