A batched network coding scheme for wireless networks

Xiaoyan Zhu · Hao Yue · Yuguang Fang ·
Yumin Wang

Abstract Several wireless network coding schemes apply either inter-flow traffic or intra-flow traffic, but not both. This paper proposes a novel batched network coding scheme to deal with both inter-flow and intra-flow traffics, which attempts to combine the advantages of both network coding approaches. Based on the idea in the well-known network coding scheme COPE, our batched network coding scheme allows each node to make use of intra-flow network coding technique to improve the transmission reliability in a lossy environment, consequently obtaining higher throughput. Moreover, we also utilize the multiple-path transmitting scheme to further increase the throughput of wireless networks with low link delivery probability. Finally, using a simplified network topology model, we show theoretically that our proposed scheme outperforms COPE significantly, particularly when the link quality is low.

Keywords Wireless networks · Network coding · Inter-flow traffic · Intra-flow traffic · Throughput

X. Zhu · H. Yue · Y. Fang (✉) · Y. Wang
National Key Laboratory of Integrated Services Networks,
Xidian University, 710071 Xi'an, China
e-mail: fang@ece.ufl.edu

X. Zhu
e-mail: xyzhu@mail.xidian.edu.cn

H. Yue
e-mail: hyue@ufl.edu

Y. Wang
e-mail: ymwang@xidian.edu.cn

Y. Fang
Department of Electrical and Computer Engineering, University
of Florida, Gainesville, FL 32611, USA

1 Introduction

Network Coding was first introduced by Ahlswede et al. [1], with which the multicast capacity of communication networks can be achieved. Later, Li et al. [2] showed that linear codes are sufficient for multicast traffic to achieve the maximum capacity bounds. At the same time, Koetter and Medard [3] proposed an algebraic approach and showed that coding and decoding can be done in the polynomial time. Ho et al. [4, 5] presented the concept of random linear network coding, which makes network coding more practical, especially in distributed networks such as wireless networks.

In the last few years, many researchers have made their efforts to develop viable network coding techniques in wireless networks [6–8]. A great deal of attention has been focused on dealing with practical issues and developing implementable protocols with network coding [9–13]. Generally speaking, network coding techniques in wireless networks can be divided into two categories: *inter-flow network coding* and *intra-flow network coding*. In the former, coding is operated on packets from different flows; while in the latter, coding is done over the packets belonging to the same flow. These two network coding techniques can increase the overall throughput of wireless networks from different aspects as we will explain next.

COPE [9] is the first practical wireless network coding scheme designed to deal with inter-flow traffic in wireless networks. With opportunistic listening and opportunistic coding, COPE intends to exploit the shared nature of wireless medium. By combining what one neighbor wants with what other neighbors have, a router with COPE can transmit multiple packets to different neighbors in a single transmission. Experiments have shown that COPE can significantly improve network throughput [9].

Unfortunately, COPE does not take into account the unreliable and unpredictable nature of wireless medium. Compared to wired links, wireless links have much poorer qualities due to channel fading and interference. COPE uses hop-by-hop acknowledgments (ACKs) and retransmissions to make sure that each next-hop node can receive the coded packets. However, when the loss rate is high, retransmitting these packets and ACKs would waste a lot of bandwidth, especially when the network scale is very large. Additionally, high loss rate may cause a detrimental effect on opportunistic coding process, as will be explained in Sect. 6, leading to degraded performance of COPE. So COPE leaves a problem open: how to deal with the highly lossy environment effectively?

Intra-flow network coding [14] is a network coding scheme to deal with intra-flow traffic, which can effectively handle reliability issue. In this scheme, after coding, each received packet contains some information about all original packets and no packet is specific to both sender and receiver. Therefore, the sender does not need to learn which particular packet is lost during the transmission. It only needs to get an ACK from the destination node when it has enough information to decode and recover the intended packet correctly. Intra-flow network coding can yield higher throughput by using advanced error-control coding technique to deal with reliability issue, which differs from inter-flow network coding. MORE [15] exploits intra-flow network coding and opportunistic routing, another technique that achieves high throughput in face of lossy wireless links.

It seems that the idea of intra-flow network coding can be used to address the problem in COPE we have discussed before, leading to our proposed scheme. Before transmission, a node will first use COPE to code the inter-flow packets in batches, followed by intra-flow network coding to form a new set of packets for final transmission. This way, the reliability issue in COPE can be effectively addressed. The resulting scheme is referred to as C&M, a new network coding scheme which takes advantage of both COPE and intra-flow network coding technique such as MORE. As a result, it could significantly increase the throughput of wireless networks. Like COPE, C&M sits between the MAC and IP layer, which keeps clear architectural abstraction and can be easily implemented. To the best of our knowledge, the only paper related to the idea presented here is [16], in which the combination of intra-flow and inter-flow network coding schemes is explored via simulations. However, there are significant differences. Chuan et al. [16] intends to demonstrate the advantage of using mixed network coding could improve the network performance when only overhearing information is used, while our paper actively designs the transmission scheme to fully utilize the advantage of the combined network

coding approach. Chuan et al. [16] evaluates the scheme based mostly on simulations, while our paper develops analytical results to characterize the performance of our scheme.

This paper is organized as follows. In the next section, we will use a simple example to identify the problem in COPE. In Sect. 3, we present our proposed scheme C&M, followed by the detailed implementation. In Sect. 5, we demonstrate the throughput benefit of C&M theoretically. In Sect. 6, we discuss how to improve our scheme.

2 A simple motivating example

In this section, we use a simple example to discuss the problem in COPE when operating in a lossy wireless environment, and to demonstrate the benefit of our approach.

Consider the scenario in Fig. 1, where the labels on the links are referred to their delivery probabilities. The links between nodes S and R have loss probability 50% in both directions. Now S wants to transmit n packets to R.

Fragouli et al. [14] have discussed this example for three different schemes. If S retransmits each packet until it successfully receives the corresponding ACK from R, each packet will need four transmissions on average, and $4n$ in all. This is obviously not a good approach due to the high number of transmissions. Another non-coding scheme with better performance implements in rounds, where S first sends all packets without stopping, waits for a batched ACK from R to tell whether each packet has arrived at the destination, and then deals with those packets not successfully received in the similar way. It only requires $2n + 2\log_2 n$ transmissions on average. However, it is still not good due to the extra ACKs for S to identify the lost packets and the increase of protocol complexity. Finally, the third scheme is an ideal solution with intra-flow network coding. S continues to transmit random linear combinations of the n packets, which is of the form $p'_j = \sum_i c_{ji} p_i$, where p_i denotes the ith packet of all the n packets, and c_{ji}s are random coefficients picked by S. When R receives n such coded packets, it could decode and recover all original packets p_i by solving a k-variable system of linear equations as below:
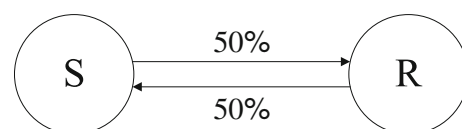

Fig. 1 A simple motivating network in a lossy wireless environment

$$\begin{cases} c_{11}p_1 + c_{12}p_2 + \cdots + c_{1n}p_n = p'_1 \\ c_{21}p_1 + c_{22}p_2 + \cdots + c_{2n}p_n = p'_2 \\ \qquad\qquad\qquad \vdots \\ c_{n1}p_1 + c_{n2}p_2 + \cdots + c_{nn}p_n = p'_n \end{cases}$$

Then, it immediately sends one acknowledgement to S as a stop signal. In this scheme, S will use $2n + 2$ transmissions on average to pass n packets to R, which reaches the lower bound for this transmission scenario.

The example in Fig. 1 could be viewed as one-hop COPE in a lossy wireless network, in which the n packets S wants to transmit are COPE-encoded packets. In multi-hop wireless networks, COPE makes use of hop-by-hop ACKs and retransmissions to guarantee reliability in the hop-by-hop fashion, as the first scheme described above. Obviously, too many retransmissions and the delay of ACKs will waste a lot of bandwidth and decrease the coding gain achieved by COPE. To make matters worse, when a wireless network transmits TCP flows, high loss rate may cause too many timeouts and excessive back-offs for TCP flows. Consequently, there are only too few packets stored in each node, which may not be enough to take advantage of the opportunistic coding in COPE. As a result, COPE may not achieve high gain in throughput in a lossy environment.

As discussed before, intra-flow network coding may provide an elegant solution to deal with the lossy environment with low complexity. In the next section, we will introduce C&M, which uses the intra-flow coding instead of hop-by-hop ACKs and retransmissions to improve the performance of COPE.

3 C&M overview

In this section, we present our design called C&M to deal with lossy links. Based on COPE, C&M maintains all of its techniques with some modifications incorporating intra-flow network coding. For the completeness and clarity of the presentation, we include some of the main ideas on COPE and details can be found in [9].

3.1 Generating COPE packets

COPE uses three main techniques to generate COPE packets: opportunistic listening, opportunistic coding and neighbor state learning (i.e., what packets each neighbor has). C&M still exploits them to generate COPE packets for further intra-flow network coding. Opportunistic listening allows each node to snoop on all communications over the wireless medium and store all the overheard packets in its buffer, called *Packet Pool*. Each node periodically
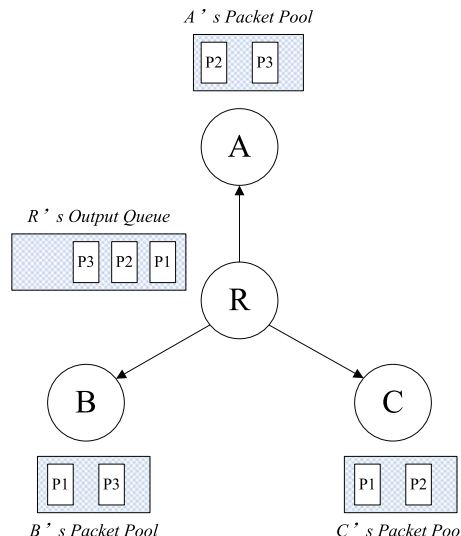

Fig. 2 An example of opportunistic coding rule

broadcasts reception reports to tell its neighbors about its state (i.e., what packets are in its packet pool), which are inserted in the data packets or the control packets the node sends.

COPE allows a node to transmit multiple packets in a single transmission, which is guaranteed by opportunistic coding with the coding rule [9]: *A node can XOR n packets $p_1, \ldots, p_n$ together to transmit to n nexthops $p_1, \ldots, r_n$ only if each next-hop r_i has all $n - 1$ packets p_j for $j \neq i$.* It chooses the largest n satisfying the coding rule in order to obtain the maximum coding benefits. Note that opportunistic coding only operates on non-encoded packets, which are named *Native Packets*.

For example, as illustrated in Fig. 2, router R has to transmit three native packets to nodes A, B and C, which are denoted as p_1, p_2 and p_3, respectively. Node A is the nexthop of p_1, p_2 is sent to node B, and p_3 is meant for node C. The three nodes have overheard some packets as shown in the figure. Assume that router R has known which packets each node has. According to the coding rule, when R is allowed to send, it should broadcast $p_1 \oplus p_2 \oplus p_3$, which could make all the three nodes A, B and C receive their intended packets in a single transmission, respectively.

Each node maintains a FIFO output queue, where all the native packets this node has to forward are stored. Every time a node starts to generate a COPE packet, it first picks up the native packet at the head of the output queue, and then searches the whole queue for appropriate packets to be encoded together with this packet according to the above coding rule.

COPE makes use of two methods for each node to learn its neighbor's state. One is the reception report, as

explained earlier. The other is guessing, which leverages the delivery probability between each pair of nodes computed by some wireless routing protocols. When lacking deterministic information, a node uses the delivery probability between a packet's previous hop and its neighbor to obtain the probability that this neighbor has overheard the packet. Guessing is very useful when there is severe loss of reception reports during transmission.

Suppose that a node wants to encode n packets together to generate a COPE packet. To ensure that the COPE packet can be decoded by all the nexthops with high probability, COPE sets a threshold value G. The sender calculates the probability that each nexthop can decode the COPE packet, which is equal to the product of the probabilities that the nexthop has heard all of the $n - 1$ packets except for the packet intended for it. If the calculated probabilities are all greater than G, which means that each nexthop could decode the COPE packet with at least probability G, this coding decision is permitted and a COPE packet is generated.

Sometimes, a node may make a wrong guess on certain neighbor's state, which will cause the nexthop not being able to decode the COPE packet. In this case, the corresponding native packet is encoded with a new group of native packets and retransmitted by the sender.

3.2 Creating and transmitting C&M packets

When a sender seizes the channel for data transmission, it first generates some COPE packets using the techniques described above. These COPE packets belong to the same batch with a batch ID assigned to them. Let k denote the number of COPE packets in a batch, and k may vary in the different batches. Then, the sender creates different random linear combinations of the k COPE packets in the current batch, which are called C&M packets. A C&M packet could be denoted as $p'_j = \sum_i c_{ji} p_i$, where the c_{ji}'s are random coefficients chosen by the sender, and the p_i's are COPE packets of the same batch. We call $\overrightarrow{c}_j = (c_{j1}, \ldots, c_{jk})$ the C&M code vector of packet p'_j, which is used in decoding process.

The sender attaches to each C&M packet a C&M header which contains the C&M code vector, the batch ID, the sender IP address and the list of the IP addresses of all receivers. Here, the receivers are referred to the next-hop nodes of the native packets encoded together to generate the k COPE packets. For example, suppose there are two COPE packets in the current batch. One contains two native packets which are sent to nodes A and B, respectively. The other is encoded from three native packets that are meant for nodes B, C and D, respectively. Then the C&M header of resulting C&M packet will list all of the four nodes A, B, C and D as receivers.

The sender does not stop transmitting different C&M packets from the current batch until it receives some ACKs, depending on which it makes a decision on whether moving to the next batch. We will explain this in detail in Sect. 4.

3.3 Packet decoding

When one C&M packet arrives, the receiver examines whether it is linearly independent of the previously received and buffered packets. If not, it is discarded by the receiver. When the receiver stores k C&M packets, it decodes the whole batch and recovers the k COPE packets from solving a k-variable system of linear equations by the simple matrix inversion:

$$\begin{pmatrix} p_1 \\ \vdots \\ p_k \end{pmatrix} = \begin{pmatrix} c_{11} & \cdots & c_{1k} \\ \vdots & \ddots & \\ c_{k1} & \cdots & c_{kk} \end{pmatrix}^{-1} \begin{pmatrix} p'_1 \\ \vdots \\ p'_k \end{pmatrix}$$

where p'_j is a C&M packet obtained from the COPE packets p_is and the C&M code vector $\overrightarrow{c}_j = (c_{j1}, \ldots, c_{jk})$.

For each COPE packet, the receiver first checks the COPE header to see whether this COPE packet contains a native packet intended for this receiver. If so, the receiver searches its packet pool for the native packets whose information are listed in the COPE header. Assuming the COPE packet contains n native packets, naturally the receiver could find $n - 1$ native packets from the packet pool according to the opportunistic coding rule. The receiver XORs the $n - 1$ native packets with the current received COPE packet to retrieve the native packet intended to it.

4 Detailed implementation

In this section, we present the detailed implementation for our proposed scheme C&M. Since our C&M scheme is based on COPE, we can only modify the COPE implementation accordingly.

4.1 Packet format

C&M inserts two variable-length coding headers in each packet, as shown in Figs. 3 and 4. They are, respectively, called the COPE header and the C&M header here, which both contain a few different blocks.

COPE header sits between the MAC and IP header and only contains two parts without the ACK block, which differs from the original one. In our C&M, ACK is delivered separately from the data packets. The first block contains metadata for decoding, such as the number of native packets encoded together, the ID (source IP address
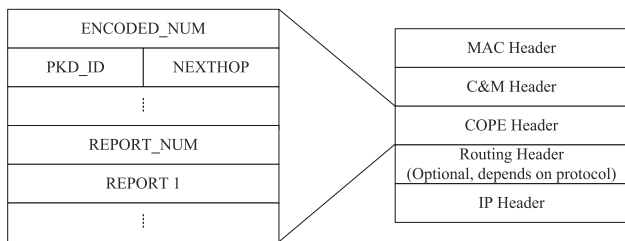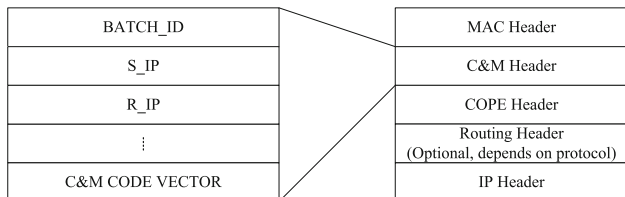
ENCODED_NUM
PKD_ID | NEXTHOP
⋮
REPORT_NUM
REPORT 1
⋮

MAC Header
C&M Header
COPE Header
Routing Header (Optional, depends on protocol)
IP Header

Fig. 3 COPE header

BATCH_ID
S_IP
R_IP
⋮
C&M CODE VECTOR

MAC Header
C&M Header
COPE Header
Routing Header (Optional, depends on protocol)
IP Header

Fig. 4 C&M header

and IP sequence number) and the nexthop of each native packet. The second block is made up of reception reports, which consists of the number of the reports and some necessary information of each reported packet for the update of neighbor's state.

C&M header resides between the MAC and COPE header and starts with the batch ID, which identifies the batch to which the C&M packet belongs, is followed by the sender IP address and the list of all receivers' IP addresses and ends with C&M code vector, which informs how to obtain the C&M packet from the COPE packets.

4.2 Pseudo-broadcast

We integrate the pseudo-broadcast technique proposed in COPE protocol into our C&M design. Pseudo-broadcast piggybacks on 802.11 unicast. It fills the link-layer destination field with the MAC address of one of the receivers listed in the C&M header. When a node overhears a C&M packet with MAC address different from its own, it examines the C&M header to see if it is an intended receiver. If so, it processes this packet further. Otherwise, it discards it. With pseudo-broadcast, not only the MAC can detect collisions and then conduct the backoff properly, but also the transmission is more reliable than the simple broadcast [9].

4.3 Acknowledgement

There are two kinds of ACKs in C&M. One is used when the receiver successfully decodes a COPE packet and extracts the native packet intended for it. The other is sent to inform the sender that it cannot decode certain COPE packet due to lack of enough information, which may be

caused by incorrect guessing as described in Sect. 3.1, which should be called NAK (negative acknowledgement) more exactly. The existence of the second ACK is necessary and important. For example, when a sender does not receive any ACK from a special receiver for a long period of time, it may be caused by the fact that the receiver has not received enough C&M packets to perform decoding due to the wireless lossy links, or it may be because the receiver has decoded the whole batch, but cannot extract its native packet from the COPE packet. The sender is likely to interpret the latter reason as the former one and continue with the useless transmissions of C&M packets when there is only the first kind of ACK. So the latter kind of ACK could make the sender decide to begin a new batch as soon as possible. The native packets that cannot be decoded by the receiver will get encoded with another set of native packets and are retransmitted in new C&M packets.

In our design, ACK is a packet, not a block in the COPE header. It is transmitted separately from the data packets. ACKs are also given priority over data packets and delivered reliably. When the receiving node finishes the decoding process of some COPE packets, it immediately sends an ACK packet to the sending node.

4.4 Flow charts for C&M

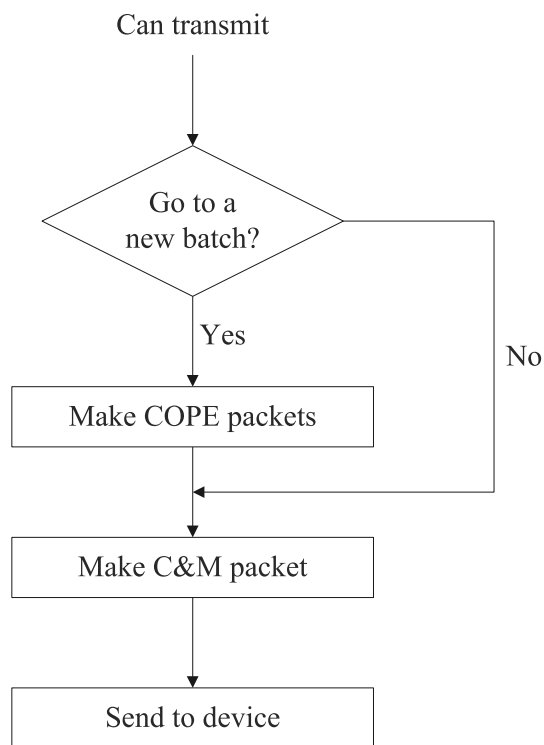Figures 5 and 6 show the flow charts for C&M.
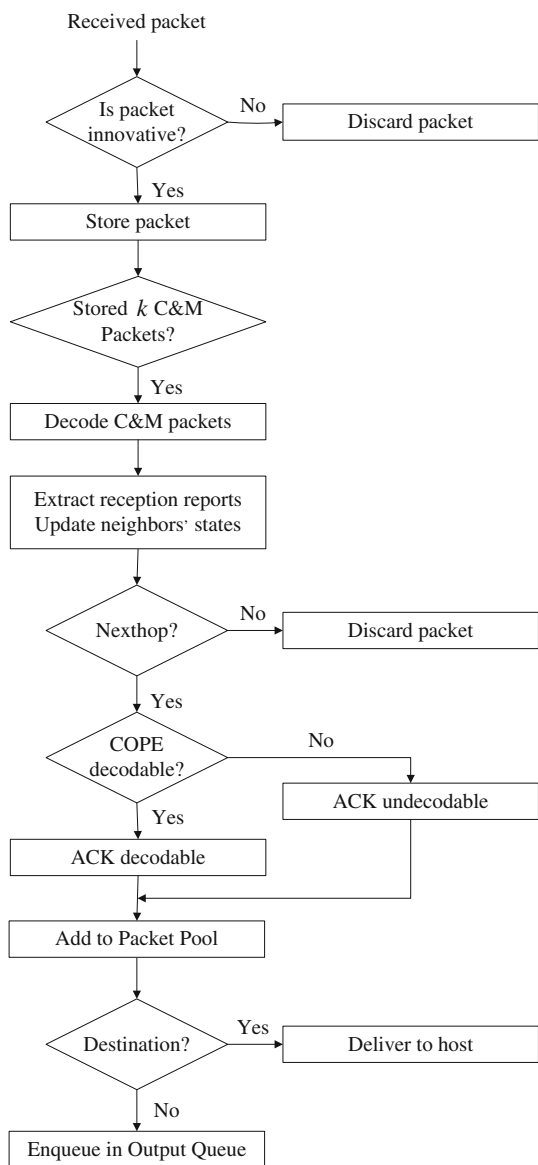
Fig. 5 Flow chart: sender side

Fig. 6 Flow chart: receiver side

When a node seizes an opportunity to send, if a new batch begins, it creates k COPE packets with reception reports in the COPE headers. Then it generates a random linear combination of the k packets, adds C&M header and sends it to the nexthops. Otherwise, it keeps on creating and broadcasting C&M packets from the current batch.

When a receiver hears a packet, it first examines the C&M header and checks whether its address is on the list of receivers. If so, the node checks whether this packet is linearly independent of the C&M packets from the same batch it has already received and stored in its buffer. The receiver only stores the innovative packets and discards non-innovative ones (a packet is called innovative if it is independent of existing packets and is called non-innovative otherwise).

Then, this receiver keeps waiting for the next C&M packet until it has received k innovative packets when it can decode the whole batch and obtain the k COPE packets by solving a k-variable system of linear equations.

The follow-on steps exactly follow what COPE does except for the ACKs. The receiver extracts the reception reports in the k COPE packets to update its neighbors' states. Because some of the packets are useless (i.e., they contain no native packets intended for this receiver), the receiver only finds and attempts to decode the useful ones to retrieve the native packets intended for it. As soon as it finishes decoding, the receiver sends an acknowledgement to the sender. Further process relies on whether the node is the final destination or just an intermediate node. If the receiver has to forward these native packets, it will place them into its output queue for further transmission. Otherwise, if the node is the destination of the flow, it will hand these packets to transport layer.

5 Performance evaluation

In this section, we analytically characterize the throughput gains for both COPE and C&M in a wireless lossy environment based on a simplified network topology. Here, the throughput gain is defined as the ratio of the throughput for the common non-coding approach to the throughput for the network coding approach (either COPE or C&M), which is equal to the ratio of the average number of transmissions required for common non-coding approach to the average number of transmissions required for network coding approach (either COPE or C&M) to deliver the same set of packets. To simplify the analysis, we have not taken the transmissions of ACKs into account because of the infrequent usage in our C&M scheme and the negligible packet size. Moreover, because the probability that randomly coded packets are not linearly independent is proven to be exponentially low [4], in our analysis we ignore the case that a receiver hears a non-innovative C&M packet. The analytical results can be used to study how lossy environment would impact the performance of wireless network coding schemes.

In this paper, we study the simple topology illustrated in Fig. 7, where nodes A and B want to exchange packets with each other and p is the delivery probability of each link in the network, which reflects how lossy a wireless link is.
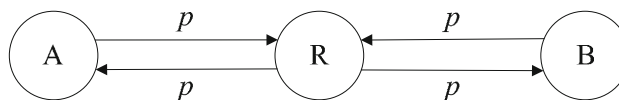

Fig. 7 A simplified topology for theoretical analysis

Here, we denote the packets of nodes A and B as p_A and p_B, respectively. We study two cases: $p = 1$ and $0 < p < 1$.

When the delivery probability equals to 1 (i.e., there is no packet loss when transmitting), in common transmitting approach, exchanging a pair of packets for A and B via R requires four transmissions. For example, node A sends its packet p_A to router R, which forwards it to node B, and B also sends its packet p_B to R, which then forwards it to node A, which exactly takes four transmissions. With COPE, each of nodes A and B first sends its own packet to R. Then R XORs the two packets to produce the coded version $p^+ = p_A \oplus p_B$ and broadcasts it. Nodes A and B could extract each other's packet from packet p^+ by using their own packets with XOR operation. COPE only needs three transmissions, so it provides a throughput gain of $\frac{4}{3} = 1.33$.

When $p < 1$, if we make use of the common transmission approach, packet p_A has to be delivered $\frac{1}{p}$ times on average from node A to router R. Another $\frac{1}{p}$ transmissions are also needed for R to forward this packet to node B. Similarly, B's transmitted packet p_B to router R or R's forwarded p_B to A also requires $\frac{1}{p}$ transmissions. As a result, $\frac{4}{p}$ transmissions are indispensable in all.

In COPE, each of nodes A and B first uses $\frac{1}{p}$ transmissions to send their own packets to R. We need to compute the average number of transmissions for R to broadcast the coded packet $p^+ = p_A \oplus p_B$ to both A and B. Let $\mathcal{E}_n$ denote the event that nodes A and B take exactly n transmissions for R to deliver the coded packet p^+ successfully. For this event to happen, the following two events must happen: (1) one node of A and B successfully receives p^+ only at the nth transmission, while the other node successfully receives p^+ not in the last nth transmission, which is denoted as *Event 1*; (2) Both nodes A and B receive p^+ successfully only at the nth transmission, which is called *Event 2*. Then, with some simple probabilistic argument, we obtain

$$\Pr(\textit{Event 1}) = \binom{2}{1} \cdot (1-p)^{n-1} p \cdot \left[1 - (1-p)^{n-1}\right],$$

$$\Pr(\textit{Event 2}) = p(1-p)^{n-1} \cdot p(1-p)^{n-1}.$$

Let N denote the number of transmissions needed for R to deliver p^+ successfully. Let p_n denotes the probability that R needs exactly n transmissions to deliver p^+ successfully, which is the union of *Event 1* and *Event 2*. Then, we have

$$p_n = \Pr(\mathcal{E}_n) = \Pr(\textit{Event 1}) + \Pr(\textit{Event 2})$$
$$= \binom{2}{1} \cdot (1-p)^{n-1} p \cdot \left[1 - (1-p)^{n-1}\right]$$
$$+ p(1-p)^{n-1} \cdot p(1-p)^{n-1}.$$

Therefore, the average number of transmissions $E(N)$ for R to deliver the coded packet p^+ to both A and B can be calculated by

$$E(N) = \sum_{n=1}^{\infty} n p_n = \frac{2}{p} - \frac{p(2-p)}{\left[1 - (1-p)^2\right]^2} = \frac{3 - 2p}{p(1-p)}.$$

Thus, when the delivery probability of each link is p, COPE brings the throughput gain as follows:

$$G_{COPE,2} = \frac{\frac{4}{p}}{\frac{2}{p} + E(N)} = \frac{8 - 4p}{7 - 4p}.$$

where $G_{COPE,2}$ denotes the throughput gain of COPE with two nodes.

Next, we analyze the throughput gain of C&M in the network as illustrated in Fig. 7 when $p < 1$. C&M requires each of A and B send k packets to router R. Here, the packets from node A are denoted as $p_{A,1}, p_{A,2}, \ldots, p_{A,k}$, and those from B are denoted as $p_{B,1}, p_{B,2}, \ldots, p_{B,k}$. R encodes every pair of packets from A and B, and produces k COPE packets $p_i^+ = p_{A,i} \oplus p_{B,i} (i = 1, 2, \ldots, k)$. When router R is allowed to transmit, it creates a random linear combination of the k COPE packets and broadcasts it, which is in the form of $p_j' = \sum_i c_{ji} p_i$, where the c_{ji}s are random coefficients picked by R, and p_i's are COPE packets obtained at the present time. R does not stop transmitting until both A and B have received at least k C&M packets, when with high probability they could decode and obtain native packets intended for them as explained earlier in Sect. 3.3.

In the whole process above, either node A or B needs transmitting $\frac{k}{p}$ times on average to deliver k packets to router R. Now we need to find the average number of transmissions for R to deliver k C&M packets to nodes A and B successfully. Similar to the analysis for COPE, let $\mathcal{E}_n$ denote the event that R needs exactly n transmissions to deliver k C&M packets. This event happens in two scenarios. It may happen when only one node of A and B successfully receives k C&M packets in the n transmissions and the kth C&M packet is received in the nth transmission while the other nodes receives k C&M packets before nth transmission, which is denoted as *Event 1*. The event $\mathcal{E}_n$ can also happen when both nodes A and B receive kth C&M packet in the nth transmission, which is denoted as *Event 2*. Then, we have

$$\Pr(\textit{Event 1}) = \binom{2}{1}\binom{n-1}{k-1} p^{k-1}(1-p)^{(n-1)-(k-1)} p$$
$$\cdot \left[\sum_{i=k}^{n-1} \binom{n-1}{i} p^i (1-p)^{n-1-i}\right].$$

$$\Pr(\textit{Event 2}) = \binom{n-1}{k-1} p^{k-1}(1-p)^{(n-1)-(k-1)} p$$
$$\cdot \binom{n-1}{k-1} p^{k-1}(1-p)^{(n-1)-(k-1)} p$$

Thus, the probability that nodes A and B have both received at least k C&M packets at exactly nth transmission

is $p_n = \Pr(Event\ 1) + \Pr(Event\ 2)$ and hence, the average number of transmissions for router R to ensure that at least k C&M packets have reached A and B can be obtained by $E(N) = \sum_{n=1}^{\infty} np_n$. The value of $E(N)$ depends on the delivery probability p and the number of C&M packets k. Therefore, the throughput gain of C&M is:

destinations via router R. We still assume that the delivery probabilities on all links are equal, which is denoted as p. This is reasonable because all links are in the same transmission environment and the model provides a good approximation when all nodes are operating under IEEE 802.11 MAC protocol.

$$G_{C\&M,2,k} = \frac{\frac{4k}{p}}{\frac{2k}{p} + E(N)} = \frac{\frac{4k}{p}}{\frac{2k}{p} + \sum_{n=k}^{\infty} np_n}$$

$$= \frac{\frac{4k}{p}}{\frac{2k}{p} + \sum_{n=k}^{\infty} n\binom{n-1}{k-1}p^{2k}(1-p)^{n-k} \cdot \left\{ 2\left[\sum_{i=k}^{n-1}\binom{n-1}{i}p^{i-k}(1-p)^{n-1-i}\right] + \binom{n-1}{k-1}(1-p)^{n-k}\right\}}$$

where $G_{C\&M,2,k}$ represents the throughput gain for C&M when there are two nodes in the network and the number of COPE packets in the current batch is k.

When $k = 1$, because the transmitting scheme in C&M is the same as in COPE, their throughput gains are equal (i.e., $G_{COPE,2} = G_{C\&M,2,1}$). Figure 8 plots the throughput gain $G_{C\&M,2,k}$ as a function of delivery probability p, with different value of k. It shows that when p decreases, the throughput gain for C&M decreases correspondingly, but it is always higher than the throughput gain for COPE (corresponding to the curve with $k = 1$). Moreover, when the channel is lossier, i.e., when p is smaller, C&M outperforms COPE much more significantly.

The analysis for the example in Fig. 7 can be extended to a more general scenario, in which s traffic flows intersect at router R, i.e., s nodes attempt to deliver packets to s

Fig. 8 Throughput gain as a function of delivery probability when the number of nodes is 2

Once again, we start with the analysis for COPE. If $p = 1$, it is easy to obtain COPE's throughput gain $G_{COPE,s} = \frac{2s}{s+1}$. When p is not equal to 1, the common traditional transmission approach needs $\frac{2s}{p}$ transmissions to accomplish the transmission process. With COPE, it takes a node $m(1 \le m \le s)\frac{1}{p}$ times to send its packet p_m to router R, and thus the total number of transmissions is $\frac{s}{p}$ when there are s nodes. R encodes these packets and sends the coded packet $p^+ = p_1 \oplus p_2 \oplus \cdots \oplus p_s$ to all the s destinations. The corresponding destination of node m uses $s - 1$ packets $p_j(1 \le j \le s, j \ne m)$ stored in its packet pool to achieve decoding and thereby obtains packet p_m from node m. Below we first derive the number of transmissions on average needed for R to deliver the coded packet p^+ to the s destination nodes.

Let Q denote the probability that it takes exactly n transmissions for a receiver to successfully receive p^+ and let P denote the probability that a receiver successfully receives p^+ in at least one of $(n - 1)$ transmissions. We have

$$Q = (1-p)^{n-1}p,$$
$$P = 1 - (1-p)^{n-1}.$$

Thus, the probability p_n that R needs exactly n transmissions to deliver p^+ to all receivers is

$$p_n = \binom{s}{1}Q^1P^{s-1} + \binom{s}{2}Q^2P^{s-2} + \cdots + \binom{s}{s}Q^sP^0$$

$$= (Q + P)^s - \binom{s}{0}Q^0P^s$$

$$= [1 - (1-p)^n]^s - \left[1 - (1-p)^{n-1}\right]^s$$

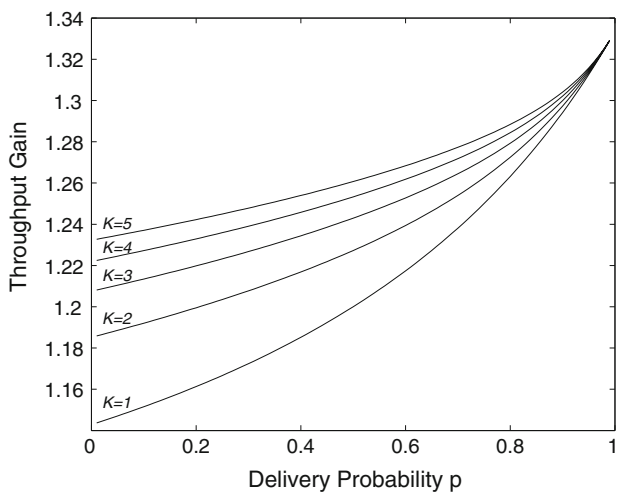The average number of transmissions $E(N)$ could be represented as below:

$$E(N) = \sum_{n=1}^{\infty} n\left([1 - (1-p)^n]^s - [1 - (1-p)^{n-1}]^s\right).$$

Therefore, we obtain the throughput gain for COPE:

$$G_{COPE,s} = \frac{\frac{2s}{p}}{\frac{s}{p} + E(N)}$$

$$= \frac{\frac{2s}{p}}{\frac{s}{p} + \sum_{n=1}^{\infty} n\left([1 - (1-p)^n]^s - [1 - (1-p)^{n-1}]^s\right)}$$

Finally, we study the throughput gain for C&M when the delivery probability p is less than 1 and there are s traffic flows in the network. C&M requests each of the s senders to deliver k packets to router R, which takes $\frac{sk}{p}$ transmissions. Let $p_{m,1}, p_{m,2}, \ldots, p_{m,s}$ denote the k packets from node $m (1 \le m \le s)$. R encodes these packets and generates k COPE packets $p_i = p_{1,i} \oplus \cdots \oplus p_{m,i} \oplus \cdots \oplus p_{s,i}$. When it is allowed to transmit, R broadcasts a C&M packet, which is a random linear combination of the k COPE packets in the form of $p'_j = \sum_i c_{ji} p_i$, to the s destination nodes. Router R will not stop transmitting until

So, the probability p_n that R takes exactly n transmissions to deliver k C&M packets is

$$p_n = \binom{s}{1} Q^1 P^{s-1} + \binom{s}{2} Q^2 P^{s-2} + \cdots + \binom{s}{s} Q^s P^0$$

$$= (Q+P)^s - \binom{s}{0} Q^0 P^s$$

$$= \left[\binom{n-1}{k-1} p^{k-1}(1-p)^{(n-1)-(k-1)} p + \sum_{i=k}^{n-1}\binom{n-1}{i}\right.$$

$$\left. \times p^i(1-p)^{n-1-i}\right]^s - \left[\sum_{i=k}^{n-1}\binom{n-1}{i} p^i(1-p)^{n-1-i}\right]^s$$

$$= p^{sk}\left\{\left[\binom{n-1}{k-1}(1-p)^{n-k} + \sum_{i=k}^{n-1}\binom{n-1}{i} p^{i-k}\right.\right.$$

$$\left.\left. (1-p)^{n-1-i}\right]^s - \left[\sum_{i=k}^{n-1}\binom{n-1}{i} p^{i-k}(1-p)^{n-1-i}\right]^s\right\}$$

Thus, the average number $E(N)$ of transmissions for R to deliver k C&M packets can be calculated and therefore, the throughput gain for C&M is given by

$$G_{C\&M,s,k} = \frac{\frac{2sk}{p}}{\frac{sk}{p} + E(N)}$$

$$= \frac{\frac{2sk}{p}}{\frac{sk}{p} + \sum_{n=k}^{\infty} n p_n}$$

$$= \frac{\frac{2sk}{p}}{\frac{sk}{p} + \sum_{n=k}^{\infty} n p^{sk}\left\{\left[\binom{n-1}{k-1}(1-p)^{n-k} + \sum_{i=k}^{n-1}\binom{n-1}{k-1} p^{i-k}(1-p)^{n-1-i}\right]^s - \left[\sum_{i=k}^{n-1}\binom{n-1}{i} p^{i-k}(1-p)^{n-1-i}\right]^s\right\}}$$

all of the s destination nodes have received at least k C&M packets. We need to compute the average number of transmissions for R to transmit k C&M packets to all s destination nodes.

Let Q denote the probability that it takes exactly n transmissions for a receiver to successfully receive k C&M packets and let P denote the probability that a receiver successfully receives at least k C&M packets in the first $(n-1)$ transmissions. Then, we have

$$Q = \binom{n-1}{k-1} p^{k-1}(1-p)^{(n-1)-(k-1)} p,$$

$$P = \sum_{i=k}^{n-1}\binom{n-1}{i} p^i(1-p)^{n-1-i}.$$

We observe that the throughput gain for C&M is the same as that for COPE when $k = 1$, i.e., $G_{COPE,s} = G_{C\&M,s,1}$. For selected values of s and k, we can draw $G_{C\&M,s,k}$ as a function of p, shown in Fig. 9. We observe that when s is fixed, the throughput gain for C&M increases as k increases, which are all higher than throughput gain for COPE (when $k = 1$). The lossier the link (i.e., p is smaller), the better the C&M performs.

6 Improvements

So far, we only illustrate the basic idea about our scheme C&M. In this section, we start with an insight on the

Fig. 9 Throughput gain for C&M as a function of delivery probability with different s and k
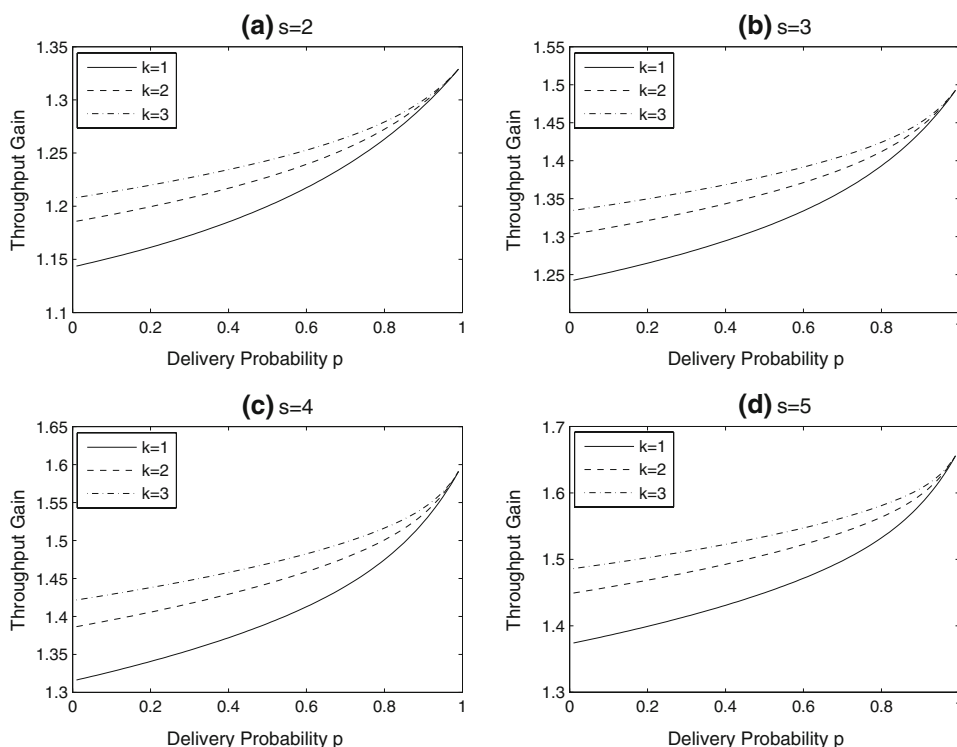
6.1 Factors affecting the performance of COPE

Opportunistic coding is the most important part in COPE, which enables transmitting multiple packets in a single transmission by coding. It is easy to see that coding cannot be done efficiently if the number of native packets stored in a node is very small. However, only enough packets cannot guarantee significant coding gain either. According to the coding rule, COPE does not code packets heading to the same next-hop together. The more packets targeted at different next-hops, the more coding opportunities appear to be. Therefore, even if a node has many native packets stored in the output queue, the coding gain is still small when most of the packets are from the same traffic flow. So there are two main factors that determine the coding gain of COPE: the number of packets stored in the sender's output queue and the fairness.

Unfairness in wireless networks mainly comes from the comparative quality of links, usually referred to as the capture effect [9]. The sender with better link usually captures the medium for longer time and transmits more packets. Although routing protocols can be used to counteract the capture effect by balancing the link load, the unfairness still happens in practice.

6.2 How to increase the delivery probability

COPE is known to significantly improve the throughput under two conditions simultaneously: high delivery probabilities of all the input traffics of a node, which could enable that there are enough native packets in the output queue to be encoded, and near equality of such probabilities, which guarantees the fairness. Hence, if we could increase the delivery probabilities of the input traffics to a certain high level (e.g., above 80%), which can also be made nearly equal at the same time, we may boost the performance further.

A simple and efficient approach is to transmit packets along multiple paths simultaneously. Consider the example
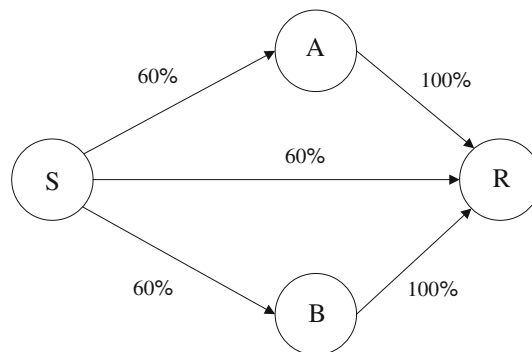

Fig. 10 Multiple-path transmission example

in Fig. 10, where the labels on the links are referred to their delivery probabilities. Now node S intends to send packets to R. S will pick the best path to R, maybe the path $S \rightarrow R$ directly. As a result, the delivery probability is 60%. However, if S broadcasts the packets, which are transmitted along the three paths (i.e. $S \rightarrow R$, $S \rightarrow A \rightarrow R$ and $S \rightarrow B \rightarrow R$), the probability of delivering a packet increases from 60% to $1 - (1 - 0.6)^3 = 93.6\%$.

In fact, the opportunistic routing [15], which enables any node overhearing a packet and closer to the destination to participate in transmitting this packet, could be viewed as a modification of a multiple-path transmission scheme, and can indeed increase the delivery probability.

6.3 Modification of C&M

In order to obtain the optimal path used in C&M, we need to compute the delivery probability for each link.

When the sender has a chance to transmit a packet, it selects several alternative paths in order to increase the delivery probability. Because C&M works on each hop of a transmission, we only allow the sender to choose the paths with only one relay node, as paths $S \rightarrow A \rightarrow R$ and $S \rightarrow B \rightarrow R$ of the scenario in Fig. 10. The sender will pick the fewest paths to make the final delivery probability greater than a threshold G (e.g., $G = 0.8$). In that case, each transmission flow into the same node has delivery probability not lower than G, guaranteeing both enough packets to code and the fairness.

However, multiple-path transmission scheme described above could bring in another difficult challenge: multiple intermediate nodes may unnecessarily forward the same packet. This is the same problem caused by opportunistic routing, which has been solved by the MORE protocol [10]. Here, we can use the same approach as in MORE to deal with this problem. Senders and receivers still work following the steps as in Figs. 5 and 6. When each relay node receives a C&M packet, it first checks the batch ID to determine whether the sender has proceeded to the next batch. It only stores linearly independent C&M packets of the current batch from the sender, and transmits the random liner combination of C&M packets in its buffer to the receiver, which is also the random linear combination of the COPE packets of the current batch.

As a final remark, it is also possible to incorporate the networking coding advantage in the routing. In [17], Ni et al. propose a coding-aware routing scheme by mapping the network coding advantage into the routing metric so that a linear programming can be developed to optimize the network coding gain. Although we may not be able to obtain elegant analytical results,we may be able to obtain

more efficient network coding schemes. This will be left as a future research direction.

7 Conclusion

In this paper, we design a novel network coding scheme, called C&M, which combines the features of inter-flow network coding and intra-flow network coding in a natural fashion. It fulfills the goal to achieve high throughput and reliability at the same time. C&M has some potential benefits. For example, in wireless networks, routing and network coding both need to consider each node's capability in terms of energy, computational power, and storage. Some nodes with high computational capability and enough energy may not be chosen to transmit packets by the routing protocol because their neighborhoods have high loss rates. C&M has the ability to increase the delivery probability by using multi-path routing techniques so that these nodes can be used to utilize the network resources more effectively and efficiently.

Our future work focuses on a few aspects. The first one is to run C&M on simulation testbed and evaluate its performance. We will analyze the experimental results carefully to study other characteristics of C&M. The second aspect attempts to combine the design of C&M with opportunistic routing or coding-aware routing as we discussed before. Hopefully, we could come up with more efficient network coding schemes for wireless networks. The third aspect is to investigate the complexity issues in our scheme. For example, the checking of the innovative packets is time-consuming, which is also a problem in all other network coding schemes. We need to address how this checking delay would affect the performance of our scheme. Finally, we need to investigate how the batched network coding delay would impact the end-to-end delay performance. As we know, batched network coding we have proposed here will have to buffer enough packets in order to have significant coding gain, hence it may not be suitable for real-time applications. Fortunately, most network coding applications target at high data rate applications with less delay constraints, and high end-to-end throughput performance may be the goal. We may have to seek tradeoff design for specific applications, and hence end-to-end delay performance under this new network coding scheme may be of great interest.

References

1. Ahlswede, R., Cai, N., Li, S., & Yeung, R. (2000). Network information flow. *IEEE Transactions on Information Theory, 46*(4), 1204–1216.
2. Li, S., Yeung, R., & Cai, N. (2003). Linear network coding. *IEEE Transactions on Information Theory, 49*(2), 371–381.
3. Koetter, R., & Medard, M. (2003). An algebraic approach to network coding. *IEEE/ACM Transactions on Networking, 11*(5), 782–795.
4. Ho, T., Medard, M., Shi, J., Effros, M., & Karger, D. (2003). On randomized network coding. In *Proc. 41st Annual Allerton Conference on Communication, Control, and Computing*.
5. Ho, T., Medard, M., Koetter, R., Karger, D., Effros, M., Shi, J., & Leong, B. (2006). A random linear network coding approach to multicast. *IEEE Transactions on Information Theory, 52*(10), 4413–4430.
6. Ramamoorthy, A., Shi, J., & Wesel, R. (2005). On the capacity of network coding for random networks. *IEEE Transactions on Information Theory, 51*(8), 2878–2885.
7. Deb, S., Effros, M., Ho, T., Karger, D., Koetter, R., Lun, D., Medard, M., & Ratnakar, N. (2005). Network coding for wireless applications: A brief tutorial. In *Proceedings of the international workshop on wireless ad-hoc networks (IWWAN)*. London, UK.
8. Lun, D., Ratnakar, N., Koetter, R., Medard, M., Ahmed, E., & Lee, H. (2005). Achieving minimum-cost multicast: A decentralized approach based on network coding. In *Proceedings IEEE INFOCOM 2005. 24th annual joint conference of the IEEE computer and communications societies* (Vol. 3). Miami, FL.
9. Katti, S., Rahul, H., Hu, W., Katabi, D., Medard, M., & Crowcroft, J. (2008). XORs in the air: Practical wireless network coding. *IEEE/ACM Transactions on Networking (TON), 16*(3), 497–510.
10. Chachulski, S., Jennings, M., Katti, S., & Katabi, D. (2007). Trading structure for randomness in wireless opportunistic routing. In *Proceedings of ACM SigComm*, Kyoto, Japan.
11. Katti, S., Katabi, D., Hu, W., Rahul, H., & Medard, M. (2005). The importance of being opportunistic: Practical network coding for wireless environments. In *Proceedings of the 43rd annual Allerton conference on communication, control, and computing*. Monticello, IL.
12. Katti, S., Katabi, D., Balakrishnan, H., & Medard, M. (2008). Symbol-level network coding for wireless mesh networks. In *ACM SIGCOMM*. Seattle, WA.
13. Zhang, X., & Li, B. (2008). Dice: A game theoretic framework for wireless multipath network coding. In *Proceedings of the 9th ACM international symposium on mobile ad hoc networking and computing, 2008* (pp. 293–302). New York, NY: ACM.
14. Fragouli, C., Katabi, D., Markopoulou, A., Medard, M., & Rahul, H. (2007). Wireless network coding: Opportunities & challenges. In *IEEE military communications conference (MILCOM'07)* (pp. 1–8). Orlando, FL.
15. Biswas, S., & Morris, R. (2005). ExOR: Opportunistic multi-hop routing for wireless networks. *ACM SIGCOMM Computer Communication Review, 35*(4), 133–144.
16. Chuan, Q., Xian, Y., Gray, C., Santhapuri, N., & Nelakuditi, S. (2008). I2MIX: Integration of intra-flow and inter-flow wireless network coding. In *Proceedings of first IEEE international workshop on wireless network coding (WiNC 2008)*, San Francisco, CA, June 2008.
17. Ni, B., Santhapuri, N., Zhong, Z., & Nelakuditi, S. (2006). Routing with opportunistically coded exchanges in wireless mesh networks. In *Poster session, proceedings of IEEE SECON*, Reston, VA.

Author Biographies

Xiaoyan Zhu received her BE degree in Information Engineering from Xidian University, Xian, China, in July 2000, and her ME degree in Information and Communications Engineering from Xidian University, Xian, China, in March 2004. She is now working towards her Ph.D. degree at Xidian University. Her research interests include wireless security and network coding.

Hao Yue received his BE degree in Telecommunication Engineering from Xidian University, Xian, China, in July 2009. He is now working towards his Ph.D. degree in the Department of Electrical and Computer Engineering at University of Florida. His research interests include wireless networks and network coding.
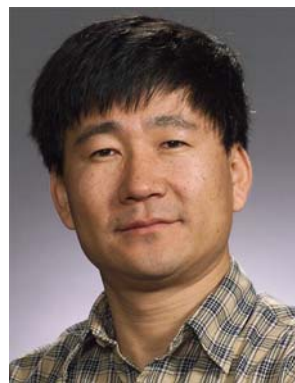
Yuguang Fang received a Ph.D. degree in Systems Engineering from Case Western Reserve University in January 1994 and a Ph.D degree in Electrical Engineering from Boston University in May 1997. He was an assistant professor in the Department of Electrical and Computer Engineering at New Jersey Institute of Technology from July 1998 to May 2000. He then joined the Department of Electrical and Computer Engineering at University of Florida in May 2000 as an assistant professor, got an early promotion to an associate professor with tenure in August 2003 and to a full professor in August 2005. He holds a University of Florida Research Foundation (UFRF) Professorship from 2006 to 2009, a Changjiang Scholar Chair Professorship with Xidian University, Xi'an, China, from 2008 to 2011, and a Guest Chair Professorship with Tsinghua University, China, from 2009 to 2012. He has published over 250 papers in refereed professional journals and conferences. Dr. Fang received the National Science Foundation Faculty Early Career Award in 2001 and the Office of Naval Research Young Investigator Award in 2002, and is the recipient of the Best Paper Award in IEEE International

Conference on Network Protocols (ICNP) in 2006. Dr. Fang is a Fellow of IEEE. He is currently serving as the Editor-in-Chief for IEEE Wireless Communications and serves/served on several editorial boards of technical journals including IEEE Transactions on Communications, IEEE Transactions on Wireless Communications, IEEE Wireless Communications Magazine and ACM Wireless Networks. He was an editor for IEEE Transactions on Mobile Computing and currently serves on its Steering Committee. He has been actively participating in professional conference organizations such as serving as the Technical Program Vice-Chair for IEEE INFOCOM'2005, Technical Program Symposium Co-Chair for IEEE Globecom'2004, and a member of Technical Program Committee for IEEE INFOCOM (1998, 2000, 2003–2010).

Yumin Wang received his BS from the Department of Radio Engineering at Xidian University, China, in July 1959. He was a lecturer from 1959 to 1979. From 1979 to 1981, he was a visiting research scholar in the Department of Electrical Engineering at University of Hawaii. He then went back as an associate professor from 1981 to 1986 and has been a Professor in the School of Telecommunications at Xidian University since 1986. He received the scientific achievement award from China Ministry of Electronics in 1983, the scientific advancement award from China Ministry of Electronics in 1992 and 1997, respectively, and the China National Technological Innovation Award in 2005. He is a senior member of IEEE, a Fellow of the Chinese Institute of Electronics, and a Fellow of the Chinese Institute of Communications. He is also an honorary member of the Board of Directors for the Chinese Institute of Cryptography.