OQMCAR: An enhanced network coding-aware routing algorithm based on queue state and local topology

Cunbo Lu, Song Xiao and Yinbin Miao
School of Communication Engineering, Xidian University
Xi'an, 710071 - China
[e-mail: 444180647@qq.com]
*Corresponding author: Cunbo Lu

Abstract

Existing coding aware routing algorithms focused on novel routing metric design that captures the characteristics of network coding. However, in packet coding algorithm, they use opportunistic coding scheme which didn't consider the queue state of the coding node and are equivalent to the conventional store-and-forward method in light traffic load condition because they never delay packets and there are no packets in the output queue of coding node, which results in no coding opportunity. In addition, most of the existing algorithms assume that all flows participating in the network have equal rate. This is unrealistic since multi-rate environments are often appeared. To overcome above problem and expand network coding to light traffic load scenarios, we present an enhanced coding-aware routing algorithm based on queue state and local topology (OQMCAR), which consider the queue state of coding node in packet coding algorithm where the control policy is of threshold-type. OQMCAR is a unified framework to merge single rate case and multiple rate case, including the light traffic load scenarios. Simulations results show that our scheme can achieve higher throughput and lower end-to-end delay than the current mechanisms using COPE-type opportunistic coding policy in different cases.

Keywords: Coding aware routing, packet coding algorithm, queue state, unified framework, multiple rates

---

1. Introduction

First proposed by Ahlswede et al. [1], network coding is becoming an emerging communication paradigm that can provide the performance improvement of wireless network. Exploiting the broadcasting nature of the physical layer, network coding can deliver multiple packets in a single transmission, and thus yields much higher throughput. This has been proved from theoretical study to practical implementations. Liu et al. [2] and Le et al. [3] provide the upper bounds on the throughput gain achieved by network coding. Network coding is also applied to cooperative communication to achieve more diversity order [4]. COPE [5], the first practical wireless network coding system for multi-hop wireless network, is shown to be capable of achieving higher throughput than traditional store-and-forward method. However, COPE relies on traditional routing mechanism to establish the transmission path and passively waits for coding opportunities. This may lead to loss of many potential coding opportunities. Therefore, routing with awareness of potential coding opportunities was proposed and has been investigated by many researchers, which it is referred to as coding-aware routing [6]. In COPE, the coding scheme never delays packets. When the wireless channel is available, the node takes the packet at the head of its output queue, checks which other packets in the queue may be encoded with this packet, combines those packets together, and broadcasts the combined version. If there are no coding opportunities, the node does not wait for the arrival of a matching codable packet. We call this coding fashion "COPE-type opportunistic coding".

However, existing coding-aware routing protocols such as ROCX [6], CAMP [7], RCR [8], DCAR [9], PNCAR [10], CPTT [11], and TCAR [12] focused on the novel routing metric design that captures the characteristics of network coding. In packet coding algorithm, they use COPE-type opportunistic coding [5] in selected coding node. Opportunistic coding scheme never delays packets and doesn't consider the queue state. It is equivalent to the store-and-forward approach in light traffic load scenarios. This is because the coding node has no packets in its output queue which results in no coding opportunity given that there is little congestion in the network. On the other hand, in network coding node, using network coding always is not the appropriate solution for all situations in coding-aware routing. Because of the stochastic nature as well as possible asymmetry between the matched flows in coding node, network coding may result in severe delay and packet loss because packets need to wait to be network-coded with each other and consume the finite buffer. In fact, the increased throughput achieved through network coding may be offset by delay incurred by waiting for coding opportunities in coding queue especially in multi-rate environments and light traffic load scenarios.

Most existing coding-aware routing protocols such as ROCX [6], CAMP [7], DCAR [9], PNCAR [10], and TCAR [12] assume that all flows participating in the network coding have the same transmission rate, which is untrue for most applications. In fact, multi-rate environments are often appeared. In multi-rate environments, network coding is an entirely new problem. RCR [8] is derived from the observation that multi-path routing with traffic splitting can increase coding gain when combined with the coding-aware routing mechanism. CPTT [11] includes the effect of multiple rates into the design of a novel routing metric for path selection. Neither of them considers the queue state of the selected coding node in packet coding algorithm. Atya et al [13] and Chaporkar et al [14] showed network coding is not a magical solution for all wireless network scenarios. And in some scenarios, applying the

store-and-forward approach leads to higher network throughput. Regulating the use of network coding in conjunction with store-and-forward in coding node can potentially result in improved long-term throughput benefits. Should the coding node wait for a coding opportunity or just transmit the packet without coding? This is the fundamental question we seek to answer.

Generally, there are two ways of delay and queue-length control: Delay based [15-17] and Queue-length based [18]. In the former one, if the waiting time of a packet is larger than a predefined threshold, the packet is forwarded without network coding. By this, the delay can be accurately controlled. But we must maintain a clock for every arriving packet in coding node. As the amount of maintained information increase, the design and implementation of controller become more complex. In the latter one, the packets are forwarded without coding if the queue length exceeds a threshold value. This way is much simpler than the former one, but the delay of a packet may be large.

In related work [19-23], the queue state of coding node is mainly considered, regardless of the coding aware routing. Chen et al. [19] formulate a Markov model of the queue state and propose a threshold-type policy to strike the optimal trade-off between power efficiency and QoS. Yu-Pin Hu et al. [20] show that if the queue state can be modeled as a Markov process, the queue length is enough to be observed as an optimal control action and this can be further detailed in [21-22]. However, there is common in these researches that the assumed topology is a basic coding structure or a line network. And in later works [20], there is no restriction on the length of the queue. In some situation, the optimal threshold may not achieve. Yu-Pin Hu et al. [23] analyze the queue state and present an on-line algorithm for making transmit/wait decisions at the coding node. But in each time unit, the related parameters are updated. The implementation becomes more complex. All of these methods are impractical to implement in coding aware routing.

In this paper, we propose an enhanced network coding-aware routing algorithm named OQMCAR, which considers the queue state in packet coding algorithm. An efficient "transmit or wait" decision for packets is made according to the queue-length based threshold policy at every transmission opportunity instead of the regular COPE-type opportunistic coding policy as used in existing coding-aware routing algorithms. COPE-type opportunistic coding policy can be seen a special instance of the queue-length based threshold policy. OQMCAR introduces network training phase before data transmission to search the optimal threshold. OQMCAR overcomes the limitation of COPE-type opportunistic coding in light traffic load condition and is a unified framework for single rate case and multiple rate case. Simulation results demonstrate that our mechanism can significantly improve the throughput performance and achieves lower end-to-end delay in any traffic rates as compared with existing work. It is simple, efficient and easy to be implemented.

The remainder of this paper is organized as follows. Section 2 explores the coding opportunity analytically. Section 3 presents a unified framework for wireless network coding with random packets arrival in single rate case and multiple rate case and obtains the analytical result on the throughput by formulating Markov models for the queue states. Based on these results, we present the implementation details for OQMCAR scheme in Section 4. Section 5 presents performance evaluation results. Section 6 concludes the paper.

2. Coding Opportunity Over A Wireless Network

In this section, based on the analysis for COPE-type opportunistic coding in [24], we present

an analysis of coding opportunity related to queue state. As shown in **Fig. 1** [24], in a basic coding structure, suppose a given coding node I_R has $N \geq 2$ neighbors within its communication range. Suppose I_R has q packets buffered in its coding queue.
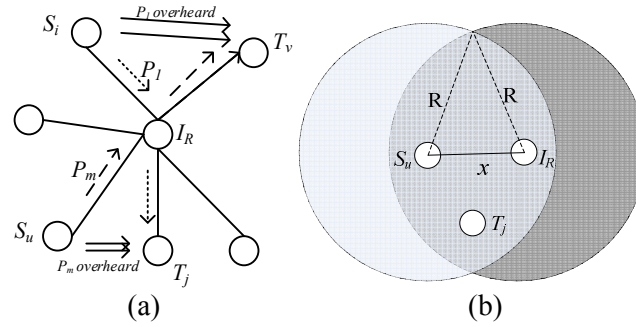

Fig. 1. (a) a coding opportunity at I_R with N neighbors; (b) An illustration on location relationship between I_R and S_u.

We define the coding opportunity at node I_R as that if there exists at least one packet in the coding queue that can be encoded with P_1, which is assumed to be the head of the coding queue, and the encoded packet can be decoded to the native packets at corresponding intended nexthops. This event is denoted as E_c. In our analysis, we assume all nodes are independently and randomly developed and the unicast traffic distribution is uniform. As shown in **Fig.1**, for the packet P_1 (the head of the queue), the previous hop is S_i and the next hop is T_j. We denote C_m the event that there exists a packet P_m in the queue that $P_1 \oplus P_m$ is decodable at their intended nexthops. As shown in **Fig. 1**, the previous hop of P_m is S_u and the next hop of P_m is T_v. To guarantee the encoded packet $P_1 \oplus P_m$ to be decoded, the node T_v must have P_1 in its queue and the node T_j must have P_m in its queue, as shown in **Fig. 1a**.

To satisfy this requirement, there are three possibilities.

Case 1) $T_j = S_u$ and $S_i = T_v$.

Case 2) $T_j = S_u$ and T_v has S_i within its communication range, or $S_i = T_v$ and T_j has S_u within its communication range.

Case 3) $T_j \neq S_u$ and $S_i \neq T_v$, and T_j has S_u within its communication range and T_v has S_i within its communication range.

Above three possibility probabilities are denoted by P_{case1}, P_{case2} and P_{case3}, respectively. We denote $P(C_m \mid n = N)$ as the probability of event C_m given that node I_R has N neighbors and $P(E_c \mid n = N, q)$ as the probability of event E_c given that node I_R has N neighbors and the length of its coding queue is q. We also denote P_h as the probability that T_j is located within the communication range of S_u, as shown in **Fig.1b**. In this figure, the communication range of each node is R, and the distance between S_u and I_R is $x(x \leq R)$.

According to the above definitions, we have these probabilities [24]:

$$P\left(P_{case1} \mid n = N\right) = \frac{1}{N(N-1)} \tag{1}$$

$$P_h = \frac{1}{90R} \int_0^R \arccos \frac{x}{2R} dx - \frac{4}{\pi R^3} \int_0^R x \sqrt{R^2 - \frac{x^2}{4}} dx \tag{2}$$

$$P\left(P_{case2} \mid n = N\right) = \frac{2(N-2)P_h}{N(N-1)} \tag{3}$$

$$P\left(P_{case3} \mid n = N\right) = \frac{(N-2)(N-3)P_h}{N(N-1)} \tag{4}$$

$$P\left(C_m \mid n = N\right) = P\left(P_{case1} + P_{case2} + P_{case3} \mid n = N\right) = \frac{1 + (N^2 - 3N + 2)P_h}{N(N-1)} \tag{5}$$

$$P\left(E_c \mid n = N, q\right) = 1 - [1 - \frac{1 + (N^2 - 3N + 2)P_h}{N(N-1)}]^{q-1} \tag{6}$$

Different from the analysis of coding opportunity related to local topology (i.e. the relation of $P\left(E_c \mid n = N, q\right)$ with N) [24], we analyze the coding opportunity related to queue state here (i.e. the relation of $P\left(E_c \mid n = N, q\right)$ with q).

In coding-aware routing, the local coding structure can be easily determined (i.e. N can be determined) in routing discovery. Under the premise, from Eq. (6), we can obtain that the probability of coding opportunity increases with the length of coding queue (q) increasing.

3. Analysis Of Queue State At A Coding node

In this section, the queue state of a coding node using the queue-length based threshold policy will be analyzed. First, Section 3.1 presents a background for the analysis. Section 3.2 presents a two-dimensional Markov chain to model the queue state of the queue-length based threshold policy and characterizes the Markov chain. Based on this Markov chain, in Section 3.3, we perform the computation of threshold.

3.1 Background

Without loss of generality, we consider a 802.11 wireless network. Note that in wireless network, the coding opportunity of three or more packets is much less than that of two packets [12]. Therefore, the throughput improvement in wireless network is mainly achieved by encoding two flows. In this paper, we only consider the opportunity of encoding two flows. To achieve the maximum throughput gain of coding aware routing, a sufficient number of encoded packets must be created at the selected coding node. Whether in single rate case or multiple rate case including the light traffic load scenarios, the packet arrival is asynchronous and random in selected coding node due to the DCF random mechanism used in the IEEE802.11 standard. This may result in severe delay and packet loss because packets need to wait to be network-coded with each other and consume the finite buffer. This may greatly reduce the overall throughout. An intuitive idea to improve the long-term throughout in single rate case and multiple rate case is to allow opportunistic transmission of some packets without network coding. This motivates the study of OQMCAR, which can opportunistically transmit

some uncoded packets according to the queue state. There is a trade-off between packet delay and network coding opportunity. The analysis of queue state of the coding node can characterize a unified description of OQMCAR in single rate case and multiple rates case.

3.2 Markov Model Description of Queue State

Assume the queue of a coding node is finite, which can hold at most K packets. Since we only consider two coding flows. Let X_1, X_2 be the number of packets (type 1 and type 2) from corresponding flow in coding node's queue. The queue state at coding node can be represented by a two-dimensional random variable denoted by $X = (X_1, X_2)$. Because the future size of coding node's queue depends only on its current size, the changes over time of X can be modeled as a Markov process. According to Yu-Pin Hsu et al. [20], for that, a stationary policy based on queue length is optimal for the network throughput to trade-off coding opportunity and packet delay.

We adopt a queue-length threshold policy to make efficient decisions for packets at coding node. That is, in coding aware routing, the selected coding node will wait until either a matching packet arrives or the length of the nonempty queue exceeds its threshold. This is easy to implement. We denote $X(t) = (X_1(t), X_2(t))$ as the Markov process that represents (X_1, X_2) at the beginning of the time t. If we define an observation window Δ_t, large enough to capture a transmission event in the wireless medium, then the dynamics of $X(n\Delta_t) = (X_1(n\Delta_t), X_2(n\Delta_t))$, with $n = 0,1,2,...$, can be represented with a discrete-time Markov chain. As the source always has packets to transmit, there will be a transmission in each time interval Δ_t. Then, in each interval Δ_t, the Markov process performs a transition. Without loss of generality, assume that in each interval Δ_t, a packet of type i (i=1, 2) arrives with probability p_i for i=1, 2. We redefine $X(n\Delta_t) = (X_1(n\Delta_t), X_2(n\Delta_t))$ as $X[n] = (X_1[n], X_2[n])$. To analyze the possible queue states of the coding node, we shall first give the following two lemmas.

Lemma 1: The queue state of coding node satisfies for all n.
$$X_1[n]X_2[n] = 0$$

Proof: We proof this by contradiction. Assume that $X_1[n]X_2[n] > 0$. We obtain that $X_1[n] > 0$, $X_2[n] > 0$. At this time, there is coding opportunity to transmit from the queue. Network coding can be always performed to transmit one packet from matched flows per observation window. Therefore, the number of at least one type packet will be reduced to zero with probability 1. Then $X_1[n] = 0$ or $X_2[n] = 0$, which contradicts the assumption. So the assumption is not held. Then we obtain $X_1[n]X_2[n] = 0$.

Remark: Lemma1 implies that at the beginning of each observation window, the coding node's queue has only one type of packet. In fact, there is only one output queue in the node. We assume that the optimal thresholds for the matched flows are equal. This assumption can highly simplify the description and analysis of queue state.

Lemma 2: The queue state of coding node satisfies $\|X[n+1] - X[n]\|_1 \leq 1$, where $\|*\|_1$ denotes the 1-norm of a vector.

Proof: We proof this by contradiction. Assume that there is a possibility that $\|X[n+1] - X[n]\|_1 > 1$.

That is $X_1[n+1] + X_2[n+1] - X_1[n] - X_2[n] > 1$.

According to Lemma 1, $X_1[n+1]X_2[n+1] = 0$, $X_1[n]X_2[n] = 0$.

Under these constraints, we obtain $X_1[n+1] - X_1[n] > 1$ when $X_2[n+1] = X_2[n] = 0$, or $X_2[n+1] - X_2[n] > 1$ when $X_1[n+1] = X_1[n] = 0$.

According to our definition for the Markov process, there is at most one transmission event in each interval. Then $X_1[n+1] - X_1[n] \leq 1$ or $X_2[n+1] - X_2[n] \leq 1$, which contradicts the assumption. So the assumption is not held. Then we obtain $\|X[n+1] - X[n]\|_1 \leq 1$

Remark: Lemma 2 implies that the queue state can only transit to its neighbor states or the current state itself.

3.3 Computation of Threshold Policy

For any system using the queue-length based threshold policy using thresholds $L_1 = L_2 = L$, the states are (0, 0), (1, 0), (2, 0) ... (L, 0), (0, 1), (0, 2) ... (0, L). To prevent packet loss due to the queue overflow, the value of L is no more than that of K.

The regular COPE-type opportunistic coding policy as used in existing coding-aware routing algorithms is a specific threshold policy. In opportunistic coding policy, we have $L_1 = L_2 = 0$. For the following, we shall first prove the existence of the optimal threshold at the coding node by the analytical formula and the simulation result.

According to (6), we obtain that the probability of coding opportunity increases with the length of coding queue (q) increasing. With q increases, forcing packets to be coded in coding nodes may induce large delay. This may in turn degrade the network throughput. We consider two extreme cases:

Case 1) q=1. The probability of coding opportunity is 0. There is no coding opportunity. The packet delay for waiting a future coding opportunity is 0. In this case, the network coding approach is equivalent to the store-and-forward approach. There are no throughput benefits given that there is no network coding gain.

Case 2) q=∞. The probability of coding opportunity is 1. Coding opportunities are always available, and waiting for such opportunities can cause substantial delay in transmission. The increased throughput achieved through network coding may be offset by delay incurred by waiting for these coding opportunities. To the extreme, network coding may in fact hurt the throughput performance.

Given the above discussion, to improve long-term throughput, there is a trade-off between network coding and packet delay. The optimal threshold exists.

In the following, we evaluate the throughput performance of the queue-length policy with various thresholds and compare its performance with the store-and-forward approach via ns2 simulation. In our simulation, the topology is a line ad-hoc network with 3 nodes. The distance between adjacent nodes is 200m. The first node and the last node exchange information. In the experiment, we use 802.11 at the Mac layer with TwoRayGround propagation model, and all data flows are generated by UDP traffic source with CBR model. The transmission range is set to 250 meters and the carrier sensing range is set to 550 meters. The coding queue size is set to 50. The traffic rates are 3kbps and 1kbps respectively. The simulation time is 300s. **Figure 2** shows the network throughput of the queue-length based threshold policy by different threshold and the store-and-forward approach (AODV).
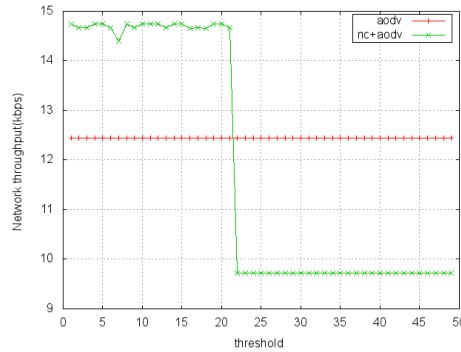
Fig. 2. Network throughput versus threshold.

The figure shows that in the queue-length based threshold policy, when the selected threshold in the network coding node exceeds some value, network coding hurts the throughput performance since the negative influence of packet delay on throughput due to waiting for future coding opportunities outperforms the positive influence of network coding. In scenarios where the selected threshold is greater than 22, applying the store-and-forward approach leads to higher network throughput. The influence of the queue state on the network performance is serious. The simulation result proves that the optimal threshold exists.

Note that the stochastic process $\{(X_1[n], X_2[n]), n \geq 0\}$ is an irreducible discrete-time Markov chain with state space $\{(0, L), (0, L-1), \cdots, (0,1), (0, 0), (1, 0), \cdots, (L-1, 0), (L, 0)\}$.

The transition probabilities in this Markov chain are given for $0 \leq i, j < L$ by

$$P_r\{X[n+1]=(i+1,0) \mid X[n]=(i,0)\}=p_1(1-p_2)$$
$$P_r\{X[n+1]=(0,j+1) \mid X[n]=(0,j)\}=(1-p_1)p_2$$
$$P_r\{X[n+1]=(i-1,0) \mid X[n]=(i,0)\}=(1-p_1)p_2$$
$$P_r\{X[n+1]=(0,j-1) \mid X[n]=(0,j)\}=p_1(1-p_2)$$
$$P_r\{X[n+1]=(i,0) \mid X[n]=(i,0)\}=p_1p_2+(1-p_1)(1-p_2)$$
$$P_r\{X[n+1]=(0,j) \mid X[n]=(0,j)\}=p_1p_2+(1-p_1)(1-p_2)$$
$$P_r\{X[n+1]=X[n]=(L,0)\}=p_1+(1-p_1)(1-p_2)$$
$$P_r\{X[n+1]=X[n]=(0,L)\}=p_2+(1-p_1)(1-p_2)$$

From above discussion about transition probabilities and Lemma1, 2, one can easily obtain a Markov chain to model the coding node's queue state. Let $\pi_{0,0}$, $\pi_{i,0}$, $\pi_{0,j}$ denote the steady-state probabilities of queue states (0, 0), (i, 0) and (0, j), respectively.

In the following, we give the analytical expressions of the steady-state probabilities.

Theorem 1: The steady-state probability of the queue state is expressed as

$$\pi_{0,0} = \frac{1}{(\frac{1-\theta^{L+1}}{1-\theta})(1+\frac{1}{\theta^L})-1}$$

$$\pi_{i,0} = \theta^i \pi_{0.0}$$

$$\pi_{0,j} = \pi_{0,0}/\theta^j$$

for all $0 < i,j \leq L$, where

$$\theta = \frac{p_1(1-p_2)}{(1-p_1)p_2}$$

Proof: The balance equations for $0 < i, j \leq L$ are:

$$\pi_{i,0} = \theta\pi_{i-1,0} \tag{7}$$

$$\theta\pi_{0,j} = \pi_{0,j-1} \tag{8}$$

According to the probability normalization property, we have

$$\pi_{0,0} + \sum_{i=1}^{L}\pi_{i,0} + \sum_{j=1}^{L}\pi_{0,j} = 1 \tag{9}$$

Substitute (7), (8) into (9), we can obtain the expression of $\pi_{0,0}$. Further, we can prove the theorem.

Remark: Using theorem 1, we can obtain the performance measure: the expected number of transmission per observation window, which can characterize the long-run average throughput.

In a paired transmission using network coding, one packet of type 1 and one packet of type 2 can be transmitted together as a single packet using XOR coding, so we can count a paired transmission as two transmissions which represent the transmission of the two packets.

Assume that we count an individual transmission and a paired transmission using network coding as one transmission and two transmissions, respectively.

The expected number of transmissions per observation window can be calculated as

$$g(L) = 2p_1p_2\pi_{0,0} + 2p_2\sum_{i=1}^{L}\pi_{i,0} + 2p_1\sum_{j=1}^{L}\pi_{0,j} + p_1(1-p_2)\pi_{L,0} + p_2(1-p_1)\pi_{0,L} \tag{10}$$

Now the optimal value of the threshold, L, can be found by minimizing the discrete function $g(L)$ in (10). As a simple approximation approach, the global minimum of $g(L)$ can be found by evaluating this function over a simple set $\{0,1,\cdots,K\}$, where K is at most several tens. We can introduce a training phase before the data transmission to search the optimal threshold for a coding node by recording the value of throughput corresponding to each threshold ranging from 1 to the maximum size of the queue minus 1. For each threshold, to characterize the long-run average throughput, we can measure the transmitted bytes in a fixed sample interval (such as 0.5s) when the data transmission is stable. This can result in very low implementation complexity due to the finite buffer of the coding node and the length of the buffer being at most several tens. The control policy at the coding node is easy to implement, and there is no computational overhead when packet arrivals. This makes these policies particularly attractive for practical implementation.

Once the network parameters are given, the optimal value can also be calculated offline in advance.

4. Coding-Aware Routing Mechanism OQMCAR

In this section, we incorporate the queue-length based threshold policy and describe the implementation of OQMCAR.

4.1 Basic Idea of OQMCAR

In OQMCAR, we consider the queue state of the coding node in packet coding algorithm to improve the long-run throughput in any traffic rates. The mechanism begins with selecting a path with maximum coding opportunities to send packets in routing discovery. In process of data transmission, when the intermediate node is the selected coding node, an efficient "transmit or wait" decision for packets is made according to the queue-length based threshold policy at every transmission opportunity instead of the regular COPE-type opportunistic coding policy which can be seen a special instance of the former where the threshold is 0. In this scheme, the coding node allows transmission without network coding only when the queue length exceeds a given threshold. The optimal threshold is determined in the introduced training phase before the data transmission. When the intermediate node is not a network coding node, the node only stores and forwards the packet according to the route table. This mechanism can make a good trade-off between packet delay due to waiting for a future opportunity and increased throughput due to network coding to improve the long-term throughput. Compared with the traditional COPE-type opportunistic coding method, this algorithm can greatly improve network throughput in any traffic rates. This mechanism is simple, efficient, and attractive for practical implementation.

4.2 Description of OQMCAR

The entire process consists of three stages: routing discovery, network training and data transmission.

　　1) Routing discovery

　　For each node in a wireless network, it maintains a neighbor list of all its one-hop neighbors and neighbors' neighbors. This information can be collected by periodically sending Hello probing messages. This list shall play a key role in detecting the potential coding opportunities of a path and packet coding algorithm.

　　When a new flow joins the wireless network, the source node begins the coding aware routing discovery process which is described as follows:

　　Step 1: The source node initiates the route discovery by broadcasting the Route Request (RREQ) message. The RREQ contains the path it has traversed, as any source routing does.

　　Step 2: Upon receiving an RREQ, an intermediate node first checks whether the RREQ has already traversed through itself. If so, the node discards the RREQ to prevent loop. If not, this node updates the path information in the RREQ and rebroadcasts the updated RREQ to discover remaining path to the destination node.

　　Step 3: When an RREQ reaches the destination node, the destination replies with the Route Reply (RREP) message using the reverse path back to the source node. The RREP is a unicast message that contains the path information.

　　Step 4: Upon receiving an RREP, an intermediate node can obtain the path information for the new path. Each node also maintains the path information for all the existing flows relayed by itself. Given this information and a neighbor list of all its one-hop neighbors and neighbors' neighbors, this node can check whether the new flow can be encoded with some existing flow(s) according to the COPE-style coding conditions related to local topology as in TCAR [12]. If there is a coding opportunity, we mark this node as coding-possible in the RREP.

　　Step 5: When the RREP(s) return to the source node, a routing decision is made based on the potential coding opportunities and the benefit of each available path, and the source node starts to enter the training phase.

　　Step 6: When the first data packet reaches an intermediate node, it stores the path

information for the selected path.

   2) Network training

    After the route establishment, the network enters a training phase. Given the network parameters, record the value of throughput corresponding to each threshold ranging from 1 to the maximum size of the coding queue minus 1. Among these values of throughput, we can find the maximum one. The corresponding threshold is the optimal threshold for the network.

   For a given threshold, when the data transmission is stable, we measure the transmitted bytes in a fixed sample interval (such as 0.5s).

   Remark: Due to the fact that the length of the queue is at most several tens, searching the optimal threshold of the network can be completed in tens of seconds.

   3) Data transmission

   The source begins to send packets along the selected new path after the training phase. Once a new path is established on an intermediate node, this node will remember that from which flow the packets can be encoded with the packets from the new flow, we name both flows as matched flow.

   In intermediate node, when the received packet is not native, the node firstly decodes the encoded packet to get the native one. If the encoded packet cannot be decoded, it will be dropped. If the packet including the decoded one is native, when the intermediate node is not a coding node, this node will only store and forward the packet according to the established route table. When the intermediate node is a coding node, once there are packets from both matched flows, this node does XOR operation for both types of packet. If there is only one type of packet, this node inserts it into the queue. If the queue length exceeds a selected threshold determined by the training phase, the packet is transmitted without network coding, otherwise waiting.

   In order to further describe the data transmission of intermediate nodes, the flow chart is showed in **Fig. 3**.
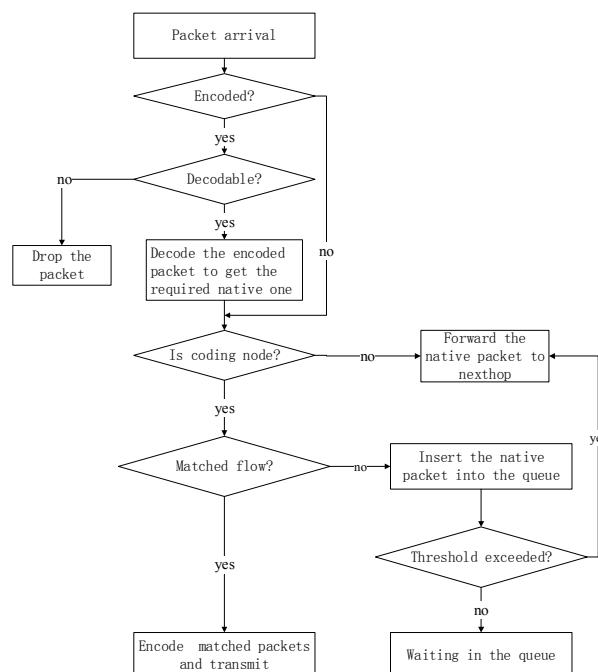

Fig. 3. Flow chart for data transmission of intermediate nodes in our proposed algorithm.

4.3 An Illustrative Example

We use the wireless network in **Fig. 4** to illustrate how OQMCAR works. Suppose the flow $0 \rightarrow 1 \rightarrow 2 \rightarrow 3$ is an existing flow.
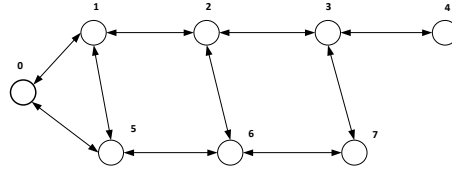

Fig. 4. An example of the wireless topology

When a new flow $5 \rightarrow 2$ joins the wireless network, the OQMCAR goes as follows:
1) Routing discovery
Step 1: Node 5 initiates the discovery by sending RREQ and adds itself into the path information in the RREQ.
Step 2: When node 1 receives the RREQ, it first checks whether the RREQ has already traversed through itself. If so, node 1 discards the RREQ to prevent loop. If not, node 1 updates the path information (i.e., adding node 1 into the path) in the RREQ and rebroadcasts the updated RREQ.
Step 3: Suppose one RREQ reaches node 2 through the path $5 \rightarrow 1 \rightarrow 2$, node 2 replies with RREP using the reverse path back to the source node. The RREP is a unicast message that contains the path information.
Step 4: When node 1 receives this RREP, node 1 can obtain the path information $5 \rightarrow 1 \rightarrow 2$ for the new flow $5 \rightarrow 2$. Node 1 discovers that the new path can be encoded with the existing flow $0 \rightarrow 1 \rightarrow 2 \rightarrow 3$, thus marking the node 1 as coding-possible in the RREP.
Step 5: The RREP finally returns to the source node 5 with information of potential coding opportunities.
The route chosen by OQMCAR for flow $5 \rightarrow 2$ is $5 \rightarrow 1 \rightarrow 2$. The selected coding node is node 1. After the route establishment, the network enters a training phase.
2) Network training
Define B_i as the transmitted bytes in an fixed sample interval T corresponding to the threshold i and L as the maximum size of the coding queue. T is set to 0.5s.
Step 1: Node 5 sends data packets along the selected path $5 \rightarrow 1 \rightarrow 2$. Node 1 decides the packets according to the queue-length based threshold policy and initiates the threshold to 1. When the data transmission is stable, node 1 records the transmitted bytes B_1 in 0.5s and increases threshold by 1. Then node 1 records B_2 in the next 0.5s and increases threshold by 1. By this manner, node 1 can get $B_1, B_2, \cdots, B_{L-1}$.
Step 2: From $B_1, B_2, \cdots, B_{L-1}$, we can choose the maximum value. The corresponding subscript is the optimal threshold.
3) Data transmission
In node 1, once there are packets from flow $0 \rightarrow 1 \rightarrow 2 \rightarrow 3$ and flow $5 \rightarrow 1 \rightarrow 2$, this node does XOR operation for both types of packet. If there is only one type of packet, node 1 inserts it into the queue. If the queue length exceeds a selected threshold determined by the training phase, the packet is transmitted without network coding, otherwise waiting.

5. Performance Evaluation

To evaluate the performance of OQMCAR, we implement the OQMCAR under NS-2, and compare it with the typical on-demand routing protocol AODV and COPE-type opportunistic coding scheme based on coding aware routing mechanism which is represented [6-12].

We implement the COPE-type opportunistic coding scheme based on coding aware routing according to the common mechanism used in most present coding aware routing protocol, in which the routing metric design includes the coding opportunity related to local topology and the path length, while using COPE-type opportunistic coding scheme in packet coding algorithm.

We define the COPE-type opportunistic coding scheme as "COPE-style" policy. Note that the "COPE-style" policy is a special instance of the queue-length threshold based policy, where the threshold is 0.

In "COPE-style" policy, the coding node never delays packets. If there are no coding opportunities, the node does not wait for the arrival of a matching codable packet. While in OQMCAR, at the coding node, an efficient "transmit or wait" decision for packets is made according to the queue-length based threshold policy. The coding node allows transmission without network coding only when the queue length exceeds a selected optimal threshold. This optimal threshold can make a good trade-off between packet delay due to waiting for a future opportunity and increased throughput due to network coding to improve the long-term throughput. Thus the COPEstyle-based coding aware routing will not perform any better than our proposed algorithm.

In our simulation, we consider the wireless topology in **Fig. 4**. There are two flows in this topology. For flow 1, the source node is node 0 and the destination node is node 2. For flow 2, the source node is node 2 and the destination node is node 5.

In the experiments, we use 802.11 at the MAC layer with TwoRayGround propagation model, and both data flows are generated by UDP traffic source with CBR model. The transmission range and the carrier sensing range are set to 250 meters and 550 meters, respectively. The queue size is set to 50. Only data packets are considered in the computation of throughput and average end-to-end delay. Nodes are not energy limited. We varied the offered load of both flows to test the performance of the three mechanisms. We assume the network is static so that the path of both flows does not change during the simulation.

5.1 Experiment 1

In this experiment, both flows have the same transmission rate. The rate of both flows varies from 100kbps to 400kbps with step of 20kbps. **Fig. 5** shows the network throughput due to the three mechanisms versus the offered load.
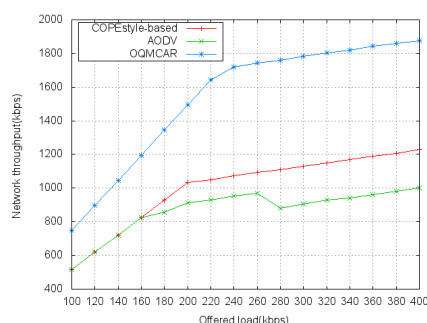

Fig. 5. Network throughput versus offered load.

From the figure, it is seen that OQMCAR can improve the network throughput substantially as compared with AODV and COPEstyle-based one in all cases. OQMCAR performs best because it considers the queue state of coding node in packet coding algorithm and could better optimize the trade-off of the effect of the packet delay and network coding on network throughput, resulting in improved long-term throughput benefits. AODV has the worst performance because it doesn't consider network coding. When the offered load is less than 160kbps, COPEstyle-based routing and AODV have the same performance. This is because COPEstyle-based one does not have any coding opportunity given that there is little congestion in the network. Using opportunistic coding, the coding node has no packets in its output queue. The opportunistic coding is equivalent to the traditional store-and-forward method in light traffic load scenarios. On average, OQMCAR provides about 44.8% throughput gain over COPEstyle-based one. As the offered load continues to grow, the congestion in the network begins to appear and increases. COPEstyle-based routing performs better than AODV because of the coding opportunities created in the routing discovery phase. When the offered load is larger than 240kbps, both OQMCAR and COPEstyle-based one increase linearly. OQMCAR achieves about 52%-60% throughput improvement over the COPEstyle-based routing.

Fig. 5 shows that OQMCAR can suit for the single rate case, including the light traffic load scenarios.

5.2 Experiment 2

In this experiment, the traffic load of flow 1 is twice the load of flow 2. We varied the rate of flow 2 from 30kbps to 310kbps with step of 20kbps to test the throughput performance of the three mechanisms. The results are shown in **Fig. 6**.
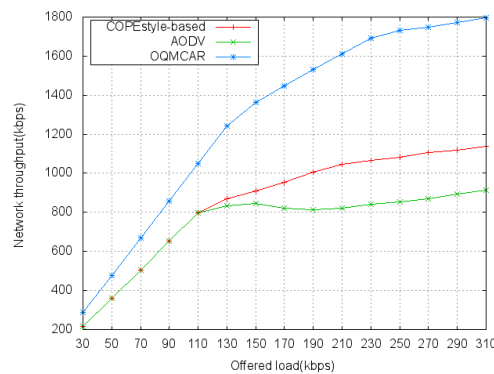

Fig. 6. Network throughput versus offered load.

From **Fig. 6**, it is observed that OQMCAR performs best in all cases by considering the queue state of coding node in packet coding algorithm and the effective queue management mechanism. AODV performs worst because it has no coding mechanism. When the offered load is less than 110kbps, the network has little congestion and little potential coding opportunity. Thus COPEstyle-based routing performs similarly to AODV. When the offered load continues to grow, the network congestion begins to appear and increases. COPEstyle-based routing outperforms AODV because network coding reduces the number of transmissions, alleviates congestion, and yields higher throughput. From the simulation results,

it is seen that OQMCAR achieves an about throughput 32% higher in light traffic load cases and up to 60% higher in network congestion than COPEstyle-based routing.

Fig. 6 shows that OQMCAR can suit for the multiple rate case, including the light traffic load scenarios.

5.3 Experiment 3

For a more thorough understanding of the throughput performance of OQMCAR in multiple rate case, the rate of one of both flows is fixed. In this experiment, the rate of flow 1 is fixed to 200kbps and the rate of flow2 varies from 100kbps to 400kbps with step of 20kbps. **Fig. 7** shows the overall network throughput of OQMCAR, COPEstyle-based routing and AODV, respectively.
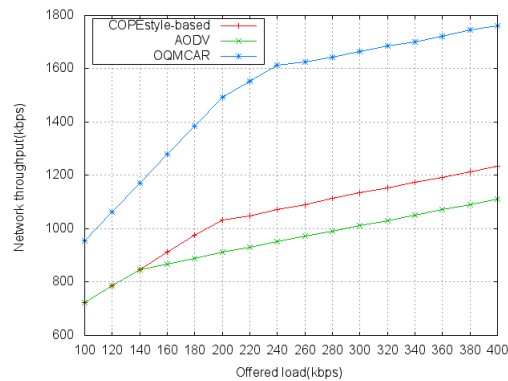

Fig. 7. Network throughput versus offered load.

For all the rates from 100kbps to 400kbps, we observe significant throughput improvement of OMCAR over COPEstyle-based routing and AODV. OQMCAR achieves an about throughput 32%-38% higher in light traffic load cases (100kbps-140kbps) and 40-50% higher in network congestion than the COPEstyle-based routing.

5.4 Experiment 4

For a thorough understanding of the delay performance of OQMCAR in single rate case, we have recorded the average end-to-end delay corresponding to experiment 1. **Fig. 8** shows the average end-to-end delay of OQMCAR, COPEstyle-based routing and AODV, respectively.
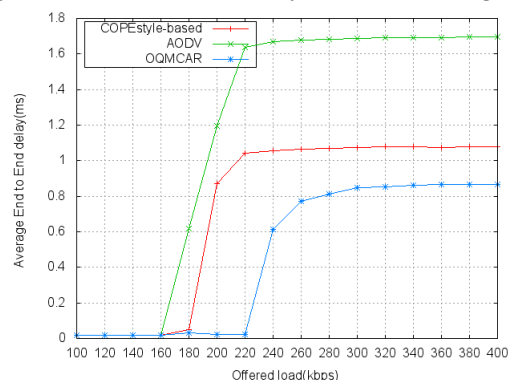

Fig. 8. Average end-to-end delay versus offered load.

From the figure, it shows that OQMCAR achieves the optimal time efficiency among all the protocols in the steady state. When the offered load is less than 160kbps, the network has little congestion and the end-to-end delay of three mechanism is very low. As the offered load continues to grow, the network congestion begins to appear and increases. With the increase of the offered load, the competition is increased, and nodes need to wait more time to have opportunity to send. The end-to-end delay of three mechanisms varies and finally increases to a stable state. During the steady state, OQMCAR and COPEstyle-based routing outperform AODV in time efficiency because network coding reduces the number of transmissions, alleviates congestion, and reduces the queuing delay of packets. Compared with COPEstyle-based routing, the effective queue management mechanism of OQMCAR can result in improved throughput benefits and the transmission competition is greatly reduced. So, the end-to-end delay of OQMCAR is less than that of COPEstyle-based one. In the steady state, the average end-to-end delay of COPEstyle-based routing is about 1.25 times of that of OQMCAR.

5.5 Experiment 5

For a thorough understanding of the delay performance of OQMCAR in multiple rate case, we have recorded the average end-to-end delay corresponding to experiment 2. **Fig. 9** shows the average end-to-end delay of flow 2 in OQMCAR, COPEstyle-based routing and AODV, respectively.
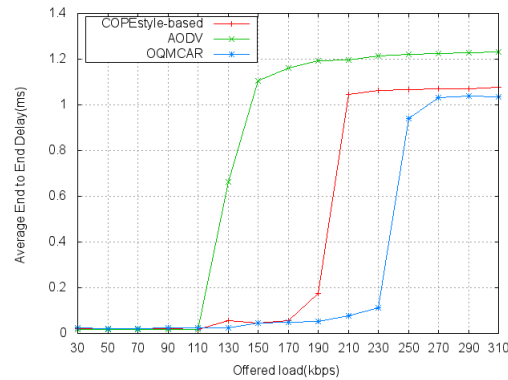

Fig. 9. Average end-to-end delay versus offered load.

From the figure, it shows that the delay of OQMCAR is the minimum among all the protocols in the steady state. In the steady state, the average end-to-end delay of COPEstyle-based routing is about 1.04 times of that of OQMCAR.

6. Conclusion

In this paper, we present OQMCAR, an enhanced network coding-aware routing algorithm which solves the problem of "whether coding" at the coding node according to the queue-length based threshold policy. Based on the theoretical analysis of coding opportunity and the random mapping-based method, a Markov model of the queue state of coding node is formulated to obtain analytical result regarding the throughput in a unified way for single rate case and multiple rate case including the light traffic load scenarios under both cases. OQMCAR introduces the network training phase after routing discovery and before data

transmission to search the optimal threshold of the queue-length based control policy. Because the regular COPE-type opportunistic coding scheme can be seen a special instance of the queue-length threshold based policy where the threshold is 0, the network throughput can be significantly improved compared with existing mechanisms using COPE-type opportunistic coding policy in different cases, even in the light traffic load scenarios. Simulation results demonstrate the improved throughput performance by our scheme. Simulation results also show that our scheme can achieve lower end-to-end delay. The proposed algorithm is simple, efficient, and easy to be implemented in wireless network with any traffic rates.

For future work, we plan to carry out empirical study of OQMCAR and apply it in real network. We are also exploring timing control problems with network coding in video transmission system, particularly how to maintain the certain delay constraint with maximizing coding opportunity in multi-hop wireless network.

References

[1] R. Ahlswede, N. Cai, S.-Y. R. Li and R. W. Yeung, "Network information flow," *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp. 1204-1216, July, 2000. Article (CrossRef Link)

[2] J. Liu, D. Goeckel and D. Towsley, "Bounds on the gain of network coding and broadcasting in wireless networks," in *Proc. of 26th IEEE Int. Conf. on Computer Communications (INFOCOM 07)*, pp. 724-732, May 6-12, 2007. Article (CrossRef Link)

[3] J. Le, J. C.S. Lui and D. M. Chiu, "How many packets can we encode?-an analysis of practical wireless network coding," in *Proc. of 27th IEEE Int. Conf. on Computer Communications (INFOCOM 08)*, pp. 13-18, April 13-18, 2008. Article (CrossRef Link)

[4] Y. Zou, J. Zhu and B. Zheng, "A fully distributed opportunistic network coding scheme for cellular relay networks," in *Proc. of IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 2937-2942, April 7-10, 2013. Article (CrossRef Link)

[5] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard and J. Crowcroft, "XORs in the air: practical wireless network coding," *IEEE/ACM Tansactions on Networking*, vol. 16, no. 3, pp. 497-510, June, 2008. Article (CrossRef Link)

[6] B. Ni, N. Santhapuri, Z. Zhong and S. Nelakuditi, "Routing with opportunistically coded exchanges in wireless mesh networks," in *Proc. of Second IEEE Workshop on Wireless Mesh Networks (WiMesh 06)*, pp. 157-159, September 25, 2006. Article (CrossRef Link)

[7] S. Han, Z. Zhong, H. Li, G. Chen, E. Chan and A. K. Mok, "Coding-aware multi-path routing in multi-hop wireless networks," in *Proc. of 27th IEEE Int. Conf. on Performance, Computing and Communications (IPCCC 08)*, pp. 93-100, December 7-9, 2008. Article (CrossRef Link)

[8] Y. Yan, Z. Zhao, B. Zhang, H. T. Mouftah and J. Ma, "Rate-adaptive coding-aware multiple path routing for wireless mesh networks," in *Proc. of 2008 IEEE Global Communications Conf. (GLOBECOM)*, pp. 1-5, November 30-December 4, 2008. Article (CrossRef Link)

[9] J. Le, John C.S. Lui and Dah-Ming Chiu, "DCAR: Distributed coding-aware routing in wireless networks," *IEEE Transactions on Mobile Computing*, vol. 9, no. 4, pp. 596-608, April, 2010. Article (CrossRef Link)

[10] Y. Peng, Y. Yang, X. Lu and X. Ding, "Coding-aware routing for unicast sessions in multi-hop wireless networks," in *Proc. of 2010 IEEE Global Communications Conf. (GLOBECOM)*, pp. 1-5, December 6- 10, 2010. Article (CrossRef Link)

[11] H. Yue, X. Zhu, C. Zhang and Y. Fang, "Cptt: A high-throughput coding-aware routing metric for multi-hop wireless networks," in *Proc. of 2012 IEEE Global Communications Conf. (GLOBECOM)*, pp. 5687-5692, December 3-7, 2012. Article (CrossRef Link)

[12] Wang W., Wu W., Guan Q. and Wang J., "TCAR: A new network coding-aware routing mechanism based on local topology detection," *Journal of Central South University*, vol. 21, no. 8, pp. 3178−3185, August, 2014. Article (CrossRef Link)

[13] A. O. F. Atya, I. Broustis, S. Singh, D. Syrivelis, S. V. Krishnamurthy and T. F. La Porta, "Wireless network coding: Deciding when to flip the switch," *in Proc. of 32th IEEE Int. Conf. on Computer Communications (INFOCOM 13)*, pp. 260-264, April 14-19, 2013. Article (CrossRef Link)

[14] P. Chaporkar and A. Proutiere, "Adaptive network coding and scheduling for maximizing throughput in wireless networks," in *Proc. of 13th annual ACM Int. Conf. on Mobile Computing and Networking (MobiCom 07)*, pp. 135-146, September 9-14, 2007. Article (CrossRef Link)

[15] E. Carrillo and V. Ramos, "On the Impact of Network Coding Delay for IEEE 802.11s Infrastructure Wireless Mesh Networks," in *Proc. of 28th IEEE Int. Conf. on Advanced Information Networking and Applications (AINA 14)*, pp. 305-312, May13-16, 2014. Article (CrossRef Link)

[16] J. T. Charith Gunasekara, A. S. Alfa and P. Yahampath, "A queueing theoretic model for opportunistic network coding," in *Proc. of IEEE Int. Conf. on Computing, Networking and Communications (ICNC)*, pp. 999-1004, January 28-31, 2013. Article (CrossRef Link)

[17] Y. Yuan, K. Wu, W. Jia and Y. Peng, "On the queueing behavior of inter-flow asynchronous network coding," *Computer Communications*, vol. 35, no. 13, pp. 1535-1548, July, 2012. Article (CrossRef Link)

[18] J. Zhang and P. Fan, "Optimal Scheduling for Network Coding: Delay vs Efficiency," in *Proc. of 2010 IEEE Global Communications Conf. (GLOBECOM)*, pp. 1-5, December 6-10, 2010. Article (CrossRef Link)

[19] Wei C., Khaled B. Letaief and Z. Cao, "Buffer-aware network coding for wireless networks," *IEEE/ACM Transactions on Networking*, vol. 20, no. 5, pp. 1389-1401, October, 2012. Article (CrossRef Link)

[20] Y. P. Hsu, N. Abedini, S. Ramasamy, N. Gautam, A. Sprintson and S. Shakkottai, "Opportunities for network coding: To wait or not to wait," in *Proc. of 2011 IEEE Int. Symposium on Information Theory (ISIT)*, pp. 791-795, July 31-August 5, 2011. Article (CrossRef Link)

[21] S. Ramasamy, "Delay-aware Scheduling in Wireless Coding Networks: To Wait or Not to Wait," *Master thesis*, Texas A&M University, 2010. Article (CrossRef Link)

[22] A. Mohapatra, N. Gautam, S. Shakkottai and A. Spintson, "Network Coding Decisions for Wireless Transmissions with Delay Consideration," *IEEE Transactions on Communications*, vol. 62, no.8, pp. 2965-2976, August, 2014. Article (CrossRef Link)

[23] Y. P. Hsu and A. Sprintson, "Opportunistic network coding: Competitive analysis," in *Proc. of 2012 IEEE Int. Symposium on Network Coding (NetCod 2012)*, pp. 191-196, June 29-30, 2012. Article (CrossRef Link)

[24] H. Guo, Y. Qian, K. Lu and N. Moayeri, "Backbone routing over multihop wireless networks: increased network coding opportunity," in *Proc. of IEEE Int. Conf. on Communications (ICC)*, pp. 1-5, May 23-27, 2010. Article (CrossRef Link)

Cunbo Lu is a Ph.D student at Xidian University, Xi'an, China. He also received his Bachelor's Degree from The PLA Information Engineering University, Zhengzhou, China and received M.S degree from Department of Communication Engineering of Xidian University, Xi'an, China. His research interests are in the fields of wireless network coding, wireless sensor network and compressed sensing.

Song Xiao is a professor and Ph.D director of communication and information system in Xidian University, Xi'an, China. He also received her M.S degree in communication and information system and Ph.D degree in signal and information processing in Xidian University, Xi'an, China in 2001 and 2004, respectively. Her research interests are in the fields of image/video compression and coding, joint source channel coding, multimedia transmission systems over wired/wireless network, robust image and video communication etc.

Yinbin Miao is a Ph.D student at Xidian University, Xi'an, China. He also received his Bachelor's Degree from Department of Communication Engineering of Jilin University, Changchun, China and received M.S degree from Department of Communication Engineering of Xidian University, Xi'an, China. His research interests are in the fields of searchable encryption, cloud computing and network security.