

Article

A Low-Jitter Wireless Transmission Based on Buffer Management in Coding-Aware Routing

Cunbo Lu ¹, Liangtian Wan ^{2,*}

¹ School of Communication Engineering, Xidian University, Xi'an 710071, Shanxi, China;
E-Mail: lucunbo1989@gmail.com

² Department of Information and Communication Systems, Hohai University, Changzhou 213022, Jiangsu, China

* Author to whom correspondence should be addressed; E-Mail: wanliangtian1@163.com;
Tel.: +86-139-4514-9131.

Academic Editor: Andrew Hudson-Smith

Received: 12 June 2015 / Accepted: 26 August 2015 / Published: 31 August 2015

Abstract: It is significant to reduce packet jitter for real-time applications in a wireless network. Existing coding-aware routing algorithms use the opportunistic network coding (ONC) scheme in a packet coding algorithm. The ONC scheme never delays packets to wait for the arrival of a future coding opportunity. The loss of some potential coding opportunities may degrade the contribution of network coding to jitter performance. In addition, most of the existing coding-aware routing algorithms assume that all flows participating in the network have equal rate. This is unrealistic, since multi-rate environments often appear. To overcome the above problem and expand coding-aware routing to multi-rate scenarios, from the view of data transmission, we present a low-jitter wireless transmission algorithm based on buffer management (BLJCAR), which decides packets in coding node according to the queue-length based threshold policy instead of the regular ONC policy as used in existing coding-aware routing algorithms. BLJCAR is a unified framework to merge the single rate case and multiple rate case. Simulations results show that the BLJCAR algorithm embedded in coding-aware routing outperforms the traditional ONC policy in terms of jitter, packet delivery delay, packet loss ratio and network throughput in network congestion in any traffic rates.

Keywords: coding aware routing; delay jitter; buffer management; unified framework; multiple rates

1. Introduction

In recent years, with the rapid development of wireless networks technologies, wireless multimedia applications have become widely used in daily life. The quality of service (QoS) demand of real-time applications challenges the design of a wireless network. Jitter is a key parameter to characterize the QoS in a wireless network.

First proposed by Ahlswede *et al.* [1], network coding is an effective solution for a wireless network because it can provide the performance improvement in throughput. Liu *et al.* [2] and Le *et al.* [3] provide the upper bounds of the throughput gain achieved by network coding. Network coding is also applied to Computer-and-Forward networks [4]. In this paper, we consider the simple one-hop XOR coding as in [5]. Exploiting the broadcasting nature of the physical layer, network coding can deliver multiple packets in a single transmission, thus the congestion is alleviated, which yields lower packet delay and jitter. For example, consider a three-node chain topology case as shown in Figure 1. Here node S1 and node S2 want to exchange a pair of packets *a* and *b* via node *R* which acts as a relay node.

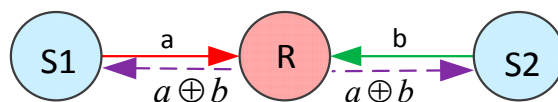


Figure 1. A three-node chain topology of network coding.

In the conventional store-and-forward approach, node S1 sends its packet *a* to node *R*, and then it is forwarded to node S2. Similarly, node S2 sends packet *b* to node S1 in two transmissions. However, in the network coding approach, when the two packets *a* and *b* are received at node *R*, they are combined by a bit-wise XOR operation, and then encoded packet $a \oplus b$ is broadcast to nodes S1 and S2 simultaneously. Because node S1 (resp. S2) has packet *a* (resp. *b*), node S1 (resp. S2) can get its required packet by XORing the encoded packet $a \oplus b$ and packet *a* (resp. *b*). Note that, in this case, a total of three transmissions are required instead of four transmissions in the conventional store-and-forward approach.

The first practical unicast routing with network coding, named as COPE, was proposed in [5]. However, COPE treats routing discovery and coding opportunity detection separately. Nodes on path passively detect coding opportunities existing among the received packets. Therefore, routing coupling with potential coding opportunity detection was proposed, which it is referred to as coding-aware routing [6]. To illustrate it clearly, an example is given in Figure 2 where there exist two flows to be routed.

Using the shortest path routing without coding opportunity detection, the disjoint paths shown in Figure 2a are very likely to be chosen. On the other hand, if we use a coding-aware routing, as shown in Figure 2b, two flows 1→2→4 and 4→2→3 intersect at node 2 which has the opportunity to perform network coding.

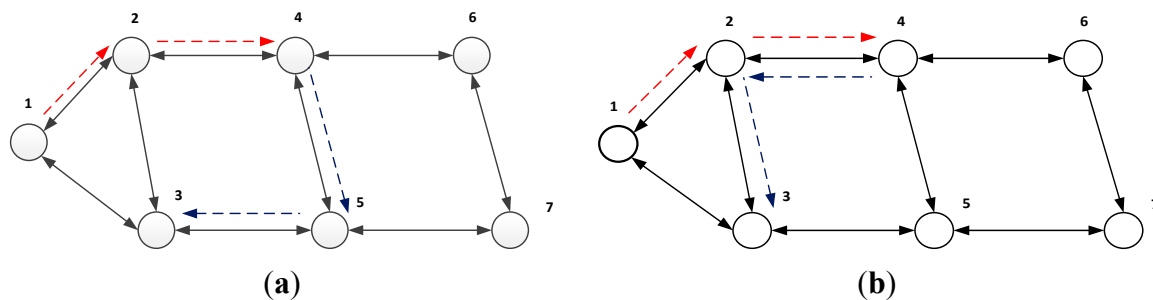


Figure 2. (a) Routing without coding consideration; (b) Routing with coding consideration at node 2.

However, existing coding-aware routing protocols such as ROCX [6], RCR [7], DCAR [8], PNCAR [9], CPTT [10], TCAR [11], NCQR [12], and LCMR [13] focused on the novel routing metric design that captures the characteristics of network coding. In the packet coding algorithm, they use ONC scheme [5] in selected coding node. When the wireless channel is available, the node takes the head packet in its coding queue, and it also checks which other packets in the coding queue may be encoded with this head packet, does XOR operation for those packets, and broadcasts the XORed packet to its neighboring nodes. If there exist no coding opportunities, our node does not wait for the arrival of a future coding opportunity and will only store and forward the head packet according to the established route table.

However, the ONC scheme does not consider the queue state and never delay packets to wait for the arrival of a future coding opportunity. Because of the stochastic nature of the packet arrival process of various flows, this may result in the loss of some potential coding opportunities and the contribution of network coding to delay and jitter performance is low. In network congestion, it is significant to reduce packet jitter in a wireless network considering the demand of the real-time applications.

Most existing coding-aware routing protocols such as ROCX [6], DCAR [8], PNCAR [9], TCAR [11], NCQR [12], and LCMR [13] 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 often appear. In multi-rate environments, network coding is an entirely new problem. Existing multi-rate based coding-aware routing protocols such as RCR [7], CPTT [10] also use ONC scheme [5] in selected coding node. RCR [7] is derived from the observation that multi-path routing with traffic splitting can increase coding gain when it combines with the coding-aware routing mechanism. CPTT [10] includes the effect of multiple rates on the design of a new routing metric for path selection. However, neither of them considers the queue state of the selected coding node in packet coding algorithm.

On the other hand, considering the queue state of the coding node, if the node delays packets to wait for the arrival of a future coding opportunity, this may result in severe delay, jitter and packet loss because packets need to wait to be network-coded with each other and consume the finite buffer. This may reduce the network performance, especially for real-time traffic. In fact, the reduced jitter achieved through network coding may be offset by jitter induced by waiting for coding opportunities in coding queue. An intuitive and straightforward solution is to allow opportunistic transmission of some packets without network coding to improve long-term jitter 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.

For the solution of above question, there is a method for queue-length control [14]: the packets are forwarded without coding if the queue length exceeds a threshold value. This way is much simpler, but the jitter of a packet may be large if the threshold is not right choice.

In related work [15–18], the queue state of coding node is mainly considered regardless of the coding aware routing. Chen *et al.* [15] formulated a Markov model of the queue state and proposed a queue-length control policy to strike the optimal trade-off between power efficiency and QoS. Yu-Pin Hu *et al.* [16] show that if the queue state can be modeled as a Markov process, the queue lengths is enough to be observed as an optimal control action and this can be further detailed in [17,18]. However, in these researches, there is an identical fact that the assumed topology is a basic coding structure or a line network. In addition, in works [16–18], there are different queues for different flows, and there is no restriction on the length of these queues. In some situations, the optimal thresholds may not achieve.

In this paper, on the basis of routing discovery in coding-aware routing, from the point of view of data transmission, we propose a low-jitter wireless transmission algorithm based on buffer management named BLJCAR, which considers the queue state in packet coding algorithm. BLJCAR consists of two stages: network jitter training and data transmission. In the data transmission phase, BLJCAR decides packets in coding node according to the queue-length based threshold policy instead of regular ONC policy as used in existing coding-aware routing algorithms. The network jitter training phase is introduced before the data transmission phase to search for the optimal threshold of the queue-length control policy. BLJCAR is a unified framework for the single rate case and multiple rate case. Note that the ONC policy can be seen as a special instance of the queue-length based threshold policy where the threshold is 0. Simulations results show that the BLJCAR wireless transmission algorithm embedded in coding-aware routing outperforms the traditional ONC policy in terms of jitter, packet delivery delay, packet loss ratio and network throughput in network congestion in any traffic rates. It is a simple, efficient and easy algorithm 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. The threshold of a queue-length based threshold policy adopted is computed as well. Based on these results, we present the implementation details for the BLJCAR scheme in Section 4. Section 5 presents performance evaluation results. Section 6 concludes the paper.

2. Coding Opportunity over a Wireless Network

Note that in a wireless network, the coding opportunity of three or more packets is much less than that of two packets [11]. Even in a wireless network with dense nodes, the coding probability of three or more packets is still very low. In random topology, the coding probability of three packets is about 2% of that of two packets under the scenario with 200 nodes. Then, the coding opportunity of three or more packets is negligible. In this paper, we consider only the coding opportunity of two packets. Without loss of generality, we consider only one real-time flow. In this section, based on the analysis of ONC in [5], we present an analysis of coding opportunity related to queue state.

2.1. Network Model

In a basic coding structure, a given coding node r with $N \geq 2$ neighbors in its communication range is considered. Note that nonuniform distribution is a special case of uniform distribution. In our analysis, we assume that all nodes are independently and randomly deployed and the unicast traffic distribution is uniform. All the nodes have an identical transmission range R , and two nodes i and j can communicate with each other if their Euclidean distance $d(i, j)$ is not longer than R . Suppose node r has L packets buffered in its coding queue.

2.2. Coding Opportunity Analysis

Suppose that two packets M_1 and M_2 are received by relay node r , and the predecessor and successor nodes of M_1 and M_2 are (S_1, D_1) and (S_2, D_2) , respectively. Packets M_1 and M_2 can be encoded in node r if one of the following four situations happens:

Case (1) $S_1 = D_2, S_2 = D_1$

Case (2) $S_2 = D_1 \neq S_1 \neq D_2, 0 < d(D_2, S_1) \leq R$

Case (3) $S_1 = D_2 \neq S_2 \neq D_1, 0 < d(D_1, S_2) \leq R$

Case (4) $D_1 \neq D_2 \neq S_1 \neq S_2, 0 < d(D_2, S_1) \leq R, 0 < d(D_1, S_2) \leq R$

Four possibility probabilities mentioned above are denoted by $P_{case1}, P_{case2}, P_{case3}$ and P_{case4} , respectively. We denoted $P_{coding}(N)$ as the probability that packets M_1 and M_2 can be encoded in node r with N neighbors. We also denote $p_r(N, L)$ as the probability that there exists a coding opportunity at node r with N neighbors and L packets buffered in its coding queue.

(1) Calculation of P_{case1}

As shown in Figure 3a, two selected nodes S_1 and D_1 are in the communication range of the given coding node r . In such case, the neighboring node pairs (S_1, D_1) among the N neighbors of node r may appear $N(N-1)$ times.

The expression for P_{case1} is

$$P_{case1} = \frac{1}{N(N-1)} \tag{1}$$

(2) Calculation of P_{case2}

Figure 3b illustrates the second case, where the selected node D_2 is in the communication range of the selected node S_1 and the given coding node r . Note that the nodes S_1 and D_2 are the neighbors of node r . Next, we explore the location relationship of nodes S_1, D_2 and r . Figure 4 illustrates a location model that node D_2 is located within the communication range of S_1 and r .

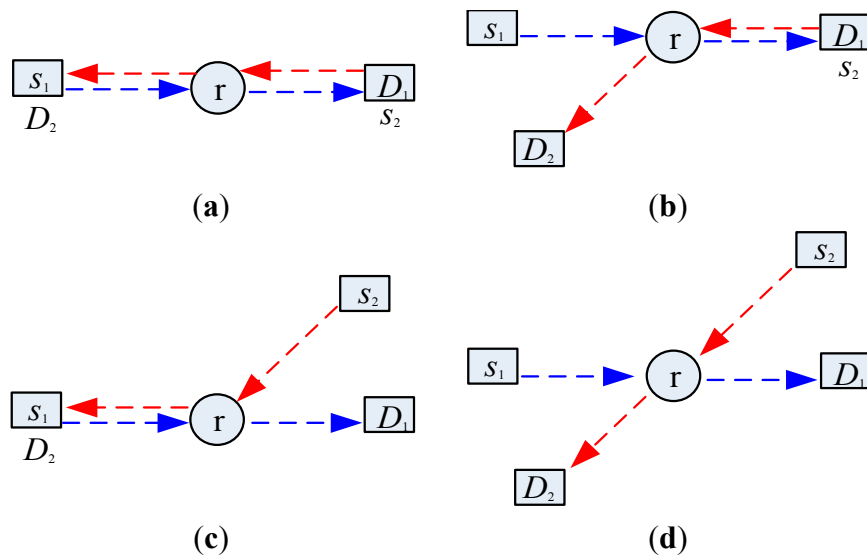


Figure 3. (a) Illustration of situation in Case One; (b) Illustration of situation in Case Two; (c) Illustration of situation in Case Three; (d) Illustration of situation in Case Four.

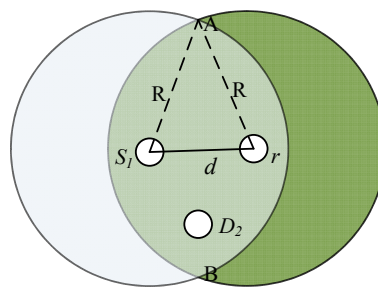


Figure 4. The location relationship between S_1 and r .

As shown in Figure 4, the selected node D_2 is located at the intersecting communication range of the selected node S_1 and the given coding node r . We also denote this probability as P_h . From Figure 4, it is observed that P_h is the probability that D_2 is located within the overlapping area between the ice-blue and the olive circles. In this figure, the distance between S_1 and r is $d(d \leq R)$. A and B are the two distinct intersecting points of the two circles with $d(S_1, r) < R$. Denote Π_2 as the expected area of the region jointly covered by node S_1 and node r . To calculate P_h , we first calculate the area of Π_2 . As shown in Figure 4, in the ice-blue circle being centered at S_1 , the area of $\Pi_2 / 2$ equals to the area of sector S_1AB minus the area of triangle S_1AB . Thus, we have

$$\begin{aligned} \Pi_2 &= 2 \left\{ \frac{1}{2} \times R^2 \times 2 \arccos\left(\frac{d}{2R}\right) - \frac{1}{2} \times \frac{d}{2} \times 2 \sqrt{R^2 - \frac{d^2}{4}} \right\} \\ &= 2R^2 \arccos\left(\frac{d}{2R}\right) - \frac{d}{2} \sqrt{4R^2 - d^2} \end{aligned} \tag{2}$$

Since the nodes are uniformly distributed, $\Pr(d \leq x)$ equals to the area of the circle having radius x and being centered at S_1 over the area having radius R and being centered at S_1 [19]. Thus, we have

$$F_1(x) = \Pr(d \leq x) = \frac{\pi x^2}{\pi R^2} = \frac{x^2}{R^2} \tag{3}$$

Then, the probability density function (PDF) is $f_1(x) = \frac{2x}{R^2}$ and the average area of the intersection is expressed as

$$E(\Pi_2) = \int_0^R (2R^2 \arccos(\frac{x}{2R}) - \frac{x}{2} \sqrt{4R^2 - x^2}) \times \frac{2x}{R^2} dx$$

$$= 4 \int_0^R x \arccos(\frac{x}{2R}) dx - \frac{1}{R^2} \int_0^R x^2 \sqrt{4R^2 - x^2} dx \tag{4}$$

where the two indefinite integral are given by

$$\int x \arccos(\frac{x}{2R}) dx = (\frac{x^2}{2} - R^2) \arccos(\frac{x}{2R}) - \frac{x \sqrt{4R^2 - x^2}}{4} \tag{5}$$

$$\int x^2 \sqrt{4R^2 - x^2} dx = -\frac{x}{4} (4R^2 - x^2)^{\frac{3}{2}} + \frac{R^2 x}{2} \sqrt{4R^2 - x^2} + 2R^4 \arcsin(\frac{x}{2R}) \tag{6}$$

Combing (4), (5) and (6), we get

$$E(\Pi_2) = 4 \times (\frac{\pi}{3} - \frac{\sqrt{3}}{4}) R^2 - \frac{1}{R^2} \times (\frac{\pi}{3} - \frac{\sqrt{3}}{4}) R^4 = (\pi - \frac{3\sqrt{3}}{4}) R^2 \tag{7}$$

Thus, according to the definition of P_h , the expression for P_h is

$$P_h = \frac{E(\Pi_2)}{\pi R^2} = \frac{\pi - 3\sqrt{3}/4}{\pi} \approx 0.586 \tag{8}$$

On the basis of Case (1), the selected node D_2 is the coding node r 's neighbors. In such case, node D_2 among the remaining $N - 2$ neighbors of node r may appear $N - 2$ times.

In summary, the expression for P_{case2} is

$$P_{case2} = (N - 2) \times P_h \times P_{case1} = \frac{(N - 2)P_h}{N(N - 1)} \tag{9}$$

(3) Calculation of P_{case3} :

Figure 3c illustrates the third case. Similar to the calculation of P_{case2} , we get

$$P_{case3} = P_{case2} = \frac{(N - 2)P_h}{N(N - 1)} \tag{10}$$

(4) Calculation of P_{case4} :

Figure 3d illustrate the fourth case, where node D_2 is in the communication range of node S_1 and node r . In addition, node S_2 is in the communication range of node D_1 and node r . On the basis of Case (1), the selected nodes D_2 and S_2 are in the communication range of the given coding node r . In such case, the neighboring node pairs (D_2, S_2) among the remaining $N - 2$ neighbors of node r may appear $(N - 2)(N - 3)$ times.

The expression for P_{case4} is

$$P_{case4} = (N - 2)(N - 3) \times P_h \times P_{case1} = \frac{(N - 2)(N - 3)P_h}{N(N - 1)} \tag{11}$$

Combining (1), (8), (9), (10) and (11), we obtain

$$P_{coding}(N) = P_{case1} + P_{case2} + P_{case3} + P_{case4} = \frac{1 + 0.586(N-1)(N-2)}{N(N-1)} \tag{12}$$

Hence, for an coding queue of length L , we have

$$P_r(N, L) = 1 - (1 - P_{coding}(N))^{L-1} = 1 - \left[1 - \frac{1 + 0.586(N-1)(N-2)}{N(N-1)}\right]^{L-1} \tag{13}$$

The basic coding structure can be easily determined (*i.e.*, N can be determined) in coding-aware routing discovery. To illustrate, consider the example in Figure 2, we can obtain $N = 3$, because coding node 2 has three neighbors.

Under the premise that N is fixed, from Equation (13), we can obtain that $P_r(N, L)$ increases with the length of coding queue (L) increasing.

Remark: In the above analysis, the necessary prerequisite is the assumption of coding opportunity based on two packets. When there are more than two packets, the prerequisite is unchanged. The coding opportunity of three or more packets has no influence on our analysis which is based on two packets. Therefore, in this case, the influence of queue state is consist with that based on two packets.

3. Analysis of Queue Length Policy at a Coding Node

In this section, the queue length policy of a coding node will be analyzed. First, Section 3.1 presents a background for the analysis. Then, Section 3.2 performs the computation of threshold.

3.1. Background

Without loss of generality, we consider an 802.11 wireless network. To achieve the maximum jitter gain of coding aware routing, a sufficient number of encoded packets must be created at the selected coding node. Whether in the single rate case or multiple rate case, the packet arrival is asynchronous and random due to the DCF random mechanism. This may result in severe delay, jitter 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 jitter benefits. An intuitive idea to improve the long-term jitter in the single rate case and multiple rate case is to allow opportunistic transmission of some packets without network coding. This motivates the study of BLJCAR, which can opportunistically transmit some uncoded packets according to the queue state. There is a trade-off between packet waiting delay and network coding opportunity. The analysis of queue state of the coding node can characterize a unified description of BLJCAR in single rate case and multiple rates case.

Since we only consider two coding flows, let L_1 and L_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 $L = (L_1, L_2)$. Because the future size of coding node's queue depends only on its current size, the changes over time of L can be modeled as a Markov process. According to Yu-Pin Hsu *et al.* [16], for that, a stationary policy based on queue length is optimal for the network performance to trade-off coding opportunity and packet waiting delay.

In BLJCAR, we adopt a queue-length based threshold policy to make efficient decisions for packets at coding node. That is, 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. In fact, there is only one output queue in the node. We assume that the optimal thresholds for the both flows are equal. This assumption can highly simplify the description and analysis of queue length policy.

3.2. Computation of Threshold Policy

Assume in the queue-length based threshold policy, the thresholds for the both flows are $L_1 = L_2 = L$. To prevent packet loss due to the queue overflow, the value of L is no more than the length of the coding queue, which can hold at most K packets.

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

According to (13), we obtain that the probability of coding opportunity ($P_r(N, L)$) increases with the threshold (L) increasing. As L increases, forcing packets to be coded in coding nodes may induce large delay. This may in turn degrade the jitter benefits. 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, there are no jitter benefits, which means there is no network coding gain.

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

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

In the following, we use the simple scenario in Figure 2 and evaluate the jitter performance of the queue-length policy embedded in coding-aware routing with various thresholds via ns2 simulation. The routing metric design includes the coding opportunity related to local topology and the path length. In the experiment, we use 802.11 at the Mac layer with Two Ray Ground propagation model, and both data flows are generated by UDP traffic source with CBR model. The coding queue size is set to 50. There are two flows in this topology. For flow 1, the source node is node 1 and the destination node is node 4. For flow 2, the source node is node 4 and the destination node is node 3. Both flows have the same transmission rate. The rates of both flows equal 300 kbps. Figure 5 shows the average packet jitter of flow 1 in the queue-length based threshold policy by different thresholds.

The figure shows that in the queue-length based threshold policy, when the threshold is different, the value of the average packet jitter is different as well. This is because there is a different trade-off between the positive influence of network coding on packet jitter and the negative influence of packet delay due to waiting for future coding opportunities. From Figure 5, we can also notice that the threshold selection is very important. If the threshold is not properly selected, the packet jitter performance can be largely influenced. The influence of the queue state on the network performance is serious. From the simulation result, we notice that the optimal threshold exists.

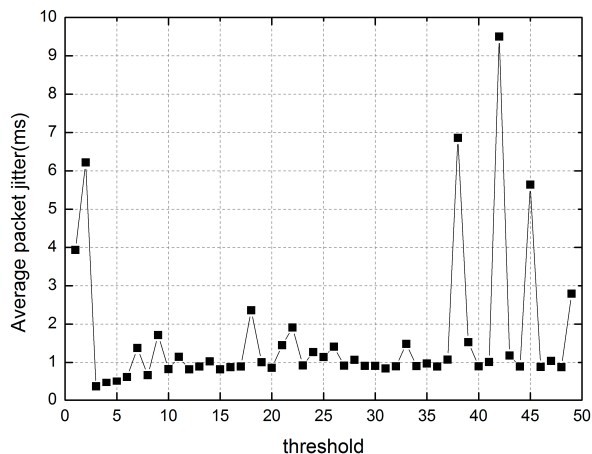


Figure 5. Average packet jitter versus threshold.

We can use a Markov process to model the queue state of the coding node. Due to the steady-state property of Markov chain, the average packet jitter per observation window can characterize the long-run average packet jitter.

We can introduce a training phase before the data transmission phase to search the optimal threshold for a coding node by recording the value of average packet jitter of the specific real-time flow 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 packet jitter of specific real-time flow, we can measure the average variation in packet waiting delay to transmit successfully in a fixed sample interval (such as 0.2 s) when data transmission is stable.

For the expression for $P_r(N, L)$

$$P_r(N, L) = 1 - (1 - P_{coding}(N))^{L-1} = 1 - \left[1 - \frac{1 + 0.586(N-1)(N-2)}{N(N-1)} \right]^{L-1} \tag{14}$$

Since we consider only the coding opportunity of two packets, the value of N is at most four. We have

$$P_r(3, L) \leq P_r(N, L) \leq P_r(2, L) \tag{15}$$

According to (13), we have

$$P_r(N, L) < P_r(N, L+1) \tag{16}$$

$$P_r(N, \infty) = 1 \tag{17}$$

Under the condition that

$$P_r(2, 11) = 1 - 0.5^{10} \approx 0.999 \tag{18}$$

$$P_r(3, 11) \approx 1 - 0.638^{10} \approx 0.989 \tag{19}$$

we have $0.989 \leq P_r(N, L) \leq 0.999$ according to (15).

Then, according to (16) and (17), when $L \geq 11$, the coding opportunity probability $P_r(N, L)$ approximates 1. When $P_r(N, L)$ is 1, coding opportunities are always available, and waiting for such opportunities can cause substantial delay and jitter in transmission. The negative influence of packet

delay on jitter due to waiting for future coding opportunities may outperform the positive influence of network coding. Then, the searching range can be reduced from $[0, K - 1]$ to $[0, \min(11, K - 1)]$ in training phase, where K is the capacity of the coding queue.

After establishing the coding-aware route, the coding node is selected and the network enters a training phase. The coding node decides packets according to the queue-length based threshold policy and records the value of average variation in packet waiting delay (AVPWD) for the successfully transmitted packets of the specific real-time flow in a fixed sample interval corresponding to each threshold ranging from 1 to $[0, \min(11, K - 1)]$. Among these values of average variation, we can find the minimum value. The corresponding threshold is the optimal threshold for the network.

This can result in very low implementation complexity. The control policy at the coding node is easy to implement, and there is no computational overhead when packets arrive. This makes these policies particularly attractive for practical implementation.

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

4. BLJCAR Mechanism

In this section, we incorporate the queue-length based threshold policy and describe the implementation of BLJCAR. The entire process consists of two stages: network jitter training and data transmission. We firstly describe the details of network jitter training in Section 4.1, and then the details of data transmission are presented in Section 4.2. Finally, we give the performance analysis of the BLJCAR mechanism.

4.1. Network Jitter Training

Let t be the fixed sample interval, p_{ij} be the successfully transmitted packet of the specific real-time flow corresponding to threshold i , $W(p_{ij})$ be the waiting time of packet p_{ij} in coding node, $InTime(p_{ij})$ be the time at which p_{ij} entered the coding node, $OutTime(p_{ij})$ be the time at which p_{ij} leave the coding node, K be the capacity of the coding queue in coding node, M_i be the number of successfully transmitted packets in a fixed sample interval t and $AVPWD(i)$ be the AVPWD corresponding to threshold i .

Then, we have

$$W(p_{ij}) = OutTime(p_{ij}) - InTime(p_{ij}) \tag{20}$$

The variation in packet waiting delay between p_{ij+1} and p_{ij} is $(W(p_{ij+1}) - W(p_{ij}))$. The total variation in packet waiting delay corresponding to threshold i in a fixed sample interval t is $\sum_{j=1}^{M_i} (W(p_{ij+1}) - W(p_{ij}))$.

Based on the above analysis, in the queue-length based threshold policy, the value of $AVPWD(i)$ is calculated as

$$AVPWD(i) = \frac{1}{M_i} \sum_{j=1}^{M_i} (W(p_{ij+1}) - W(p_{ij})) \tag{21}$$

Combining (20) and (21), we obtain

$$AVPWD(i) = \frac{1}{M_i} \sum_{j=1}^{M_i} [(OutTime(p_{ij+1}) - InTime(p_{ij+1})) - (OutTime(p_{ij}) - InTime(p_{ij}))] \quad (22)$$

After establishing the coding-aware route, the network enters a network jitter training phase which is described as follows:

Step 1: The coding node initiates the threshold as 0. The source begins to send packets along the new selected path and the coding node makes an efficient “transmit or wait” decision for packets of both flows according to the queue-length based threshold policy. When the data transmission is stable, in a fixed sample interval t , the coding node records $InTime(p_{0j})$ and $OutTime(p_{0j})$ for every successfully transmitted packet p_{0j} and calculates the number of successfully transmitted packets M_0 . According to (22), the coding node can obtain AVPWD(0);

Step 2: The coding node set threshold i to 1. In a fixed sample interval t , the coding node records $InTime(p_{1j})$ and $OutTime(p_{1j})$ for every successfully transmitted packet p_{1j} and calculates the number of successfully transmitted packets M_1 . The coding node calculates AVPWD(1) according to (22) and increases i by 1. Repeat above process, the coding node can obtain AVPWD(2), AVPWD(3), AVPWD(4)...AVPWD(min($K - 1, 11$));

Step 3: After obtaining above values of AVPWD(0), AVPWD(1) ... AVPWD(min($K - 1, 11$)), we can find the minimum value. The corresponding threshold is the optimal threshold for the network.

Remark: Due to the fact that the length of the searching range is at most 12, searching the optimal threshold of the network can be completed quickly.

4.2. Data Transmission

The value of selected optimal threshold is assumed to be L^* in the training phase.

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 in routing discovery. 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 packets and broadcasts the encoded packet to its neighboring nodes. If there is only one type of packet, this node inserts it into the queue and checks the queue length, which it is denoted by L .

If $L > L^*$, the packet is transmitted without network coding, otherwise waiting.

In order to describe clearly the behavior of intermediate nodes in the data transmission phase, the related flow chart is showed in Figure 6.

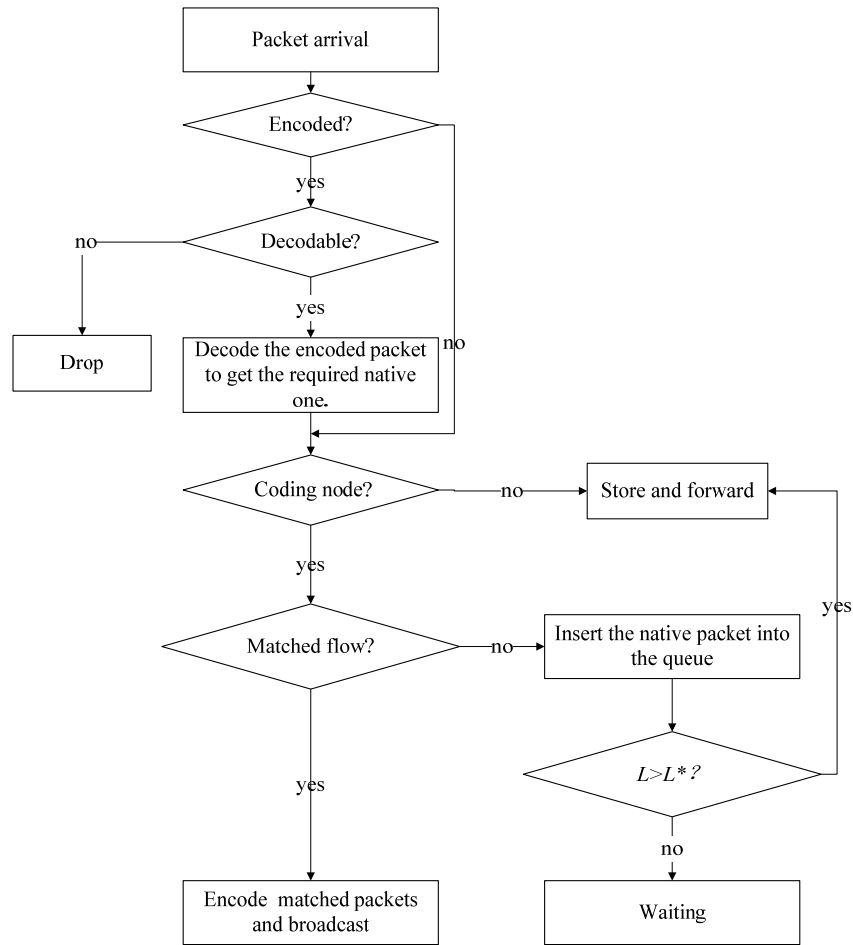


Figure 6. Flow chart for data transmission of intermediate nodes in our proposed algorithm.

4.3. Performance Analysis

Using the symbols defined in Section 4.1, we give two theorems in the following.

Theorem 1: The introduced network jitter training phase can have a very small influence on the network performance of coding aware routing in the long run.

Proof: In the network jitter training phase, the coding node successfully transmits $M_0 + M_1 + \dots + M_{\min(K-1,11)}$ packets. The transmission time of network jitter training phase is $(\min(K-1,11))t$. In the following data transmission phase, the corresponding transmission time T_{data} is very large as compared with the value of $(\min(K-1,11))t$ in the long run. Therefore, when the transmission rate is fixed, the number of transmitted packet M_{data} is very large as compared with the

value of $M_0 + M_1 + \dots + M_{\min(K-1,11)}$. $\frac{M_0 + M_1 + \dots + M_{\min(K-1,11)}}{M_{data}}$ is very small. Therefore, we can obtain

that the packets transmitted in network jitter training phase can induce very small influence on the calculation of corresponding network performance parameter. Further, we can prove the theorem.

Theorem 2: The BLJCAR policy performs better in jitter performance than the regular ONC policy.

Proof: Note that in ONC policy, the coding node never delay packets. If there are no coding opportunities, our node does not wait for the arrival of a matching codeable packet. The ONC policy is a special instance of the queue-length based threshold policy, where the threshold is 0. In ONC policy,

there is no network jitter training phase. Thus, the time of network jitter training phase can be seen as 0. According to Theorem 1, the jitter performance of ONC policy can be characterized by $AVPWD(0)$. In the BLJCAR policy, the jitter performance can be characterized by $\min(AVPWD(0), AVPWD(1) \dots AVPWD(\min(K-1, 11)))$. Thus, the theorem is proved.

Remark: In fact, the fixed sample interval t can be very small. A few tenth of a second is enough. Network jitter training phase can be completed in several seconds. According to theorem 1 and 2, the long-run jitter performance can be improved at cost of several seconds.

5. Performance Evaluation

To evaluate the performance of BLJCAR, we implement the BLJCAR scheme embedded in coding-aware routing using NS-2, and compare it with the ONC scheme based on coding aware routing mechanism, which is represented in [6–12]. In the above two coding aware routing mechanisms, the routing metric design includes the coding opportunity related to local topology and the path length. This routing metric design is the common mechanism used in most present coding aware routing protocols.

In our simulation, we consider the wireless topology in Figure 2. There are two flows in this topology. For flow 1, the source node is node 1 and the destination node is node 4. For flow 2, the source node is node 4 and the destination node is node 3. We use 802.11 at the MAC layer with the Two Ray Ground 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. Nodes are not energy limited. We vary the offered load of both flows to test the performance of the two mechanisms in network congestion. For a more thorough understanding of the performance of BLJCAR, it is interesting to study the capability of BLJCAR to cope with the longer communication paths. To achieve this, we change the destination node of flow 1. In experiments 4, 5 and 6, the destination node of flow 1 is changed from node 4 to node 6. We assume the network is static so that the paths of both flows do not change during the simulation. Without loss of generality, flow 1 is considered as the specific real-time flow.

5.1. Experiment 1

In this experiment, both flows have the same transmission rate. The rate of both flows varies from 280 kbps to 560 kbps with step of 20 kbps. Figure 7 shows the jitter performance of specific real-time flow, network throughput, average end-to-end delay and packet loss ratio due to both mechanisms *versus* offered load.

From Figure 7a, it is seen that BLJCAR can improve the jitter performance substantially as compared with the ONC policy because it considers the queue state of coding node in packet coding algorithm and could better optimize the trade-off of effect of packet waiting delay and network coding on jitter, resulting in improved long-term jitter benefits.

From Figure 7b,d, it is observed that BLJCAR can improve the network throughput and packet loss rate as compared with the ONC policy because it delays packets to wait for the arrival of a future coding opportunity, which allows more data transmissions in each time unit.

From Figure 7c, it also shows that BLJCAR scheme outperforms the ONC policy in average end-to-end delay because the effective queue management mechanism of BLJCAR can result in

improved jitter benefits, the transmission competition is greatly reduced and the queuing delay of packets is reduced.

Figure 7 shows that BLJCAR is suitable for the single rate case.

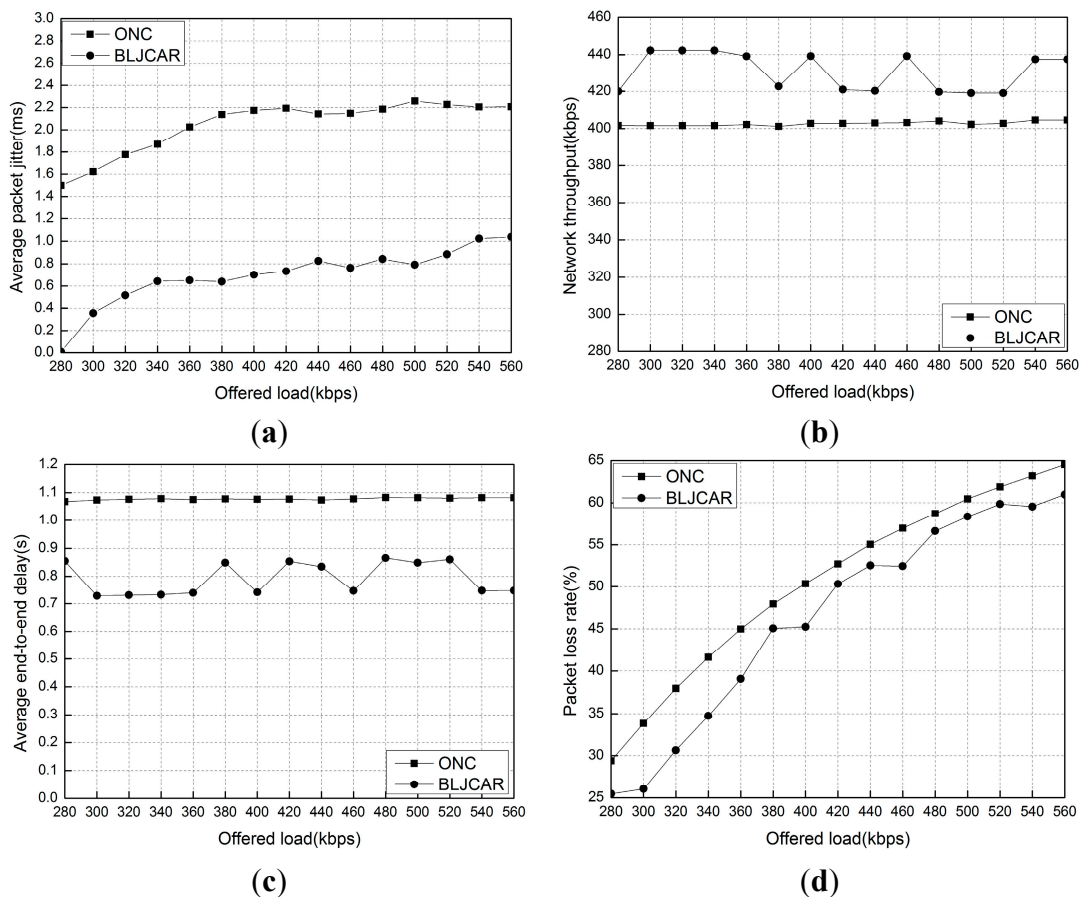


Figure 7. (a) Average packet jitter *versus* offered load; (b) Network throughput *versus* offered load; (c) Average end-to-end delay *versus* offered load; (d) Packet loss ratio *versus* offered load.

5.2. Experiment 2

In this experiment, the traffic load of flow 1 is twice the load of flow 2. We vary the rate of flow 2 from 200 kbps to 340 kbps with step of 10 kbps to test the jitter performance of flow 1, network throughput, average end-to-end delay and packet loss ratio of the two mechanisms. The results are shown in Figure 8.

From Figure 8, it is observed that BLJCAR performs better than the ONC policy in jitter, network throughput, average end-to-end delay and packet loss ratio by considering the queue state of coding node in packet coding algorithm and the effective queue management mechanism. The effective queue management mechanism of BLJCAR can result in improved jitter benefits, alleviates congestion, reduces the queuing delay of packets and allows more data transmissions in each time unit.

Figure 8 shows that BLJCAR is suitable for the multiple rate case.

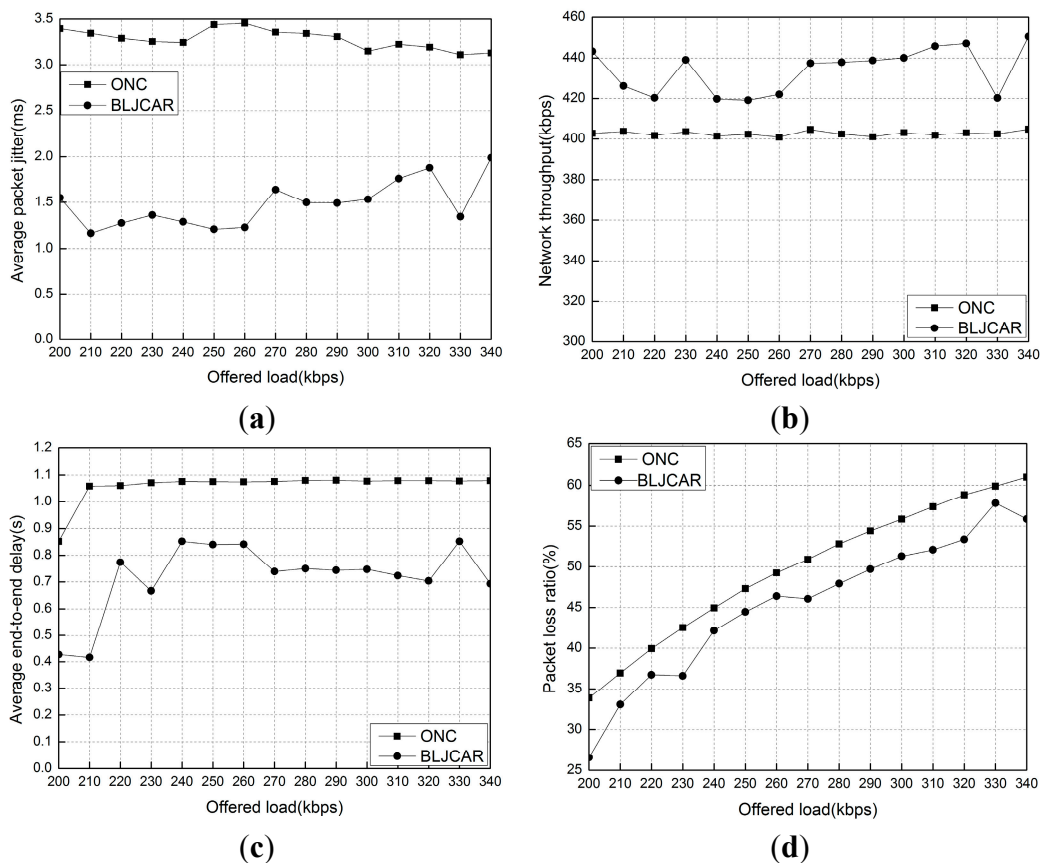


Figure 8. (a) Average packet jitter *versus* offered load; (b) Network throughput *versus* offered load; (c) Average end-to-end delay *versus* offered load; (d) Packet loss ratio *versus* offered load.

5.3. Experiment 3

For a more thorough understanding of the performance of BLJCAR in multiple rate case, the rate of one of both flows is fixed. In this experiment, the rate of flow 1 is fixed to 300 kbps and the rate of flow 2 varies from 300 kbps to 600 kbps with step of 20 kbps. Figure 9a–d shows the jitter performance of flow 1, network throughput, average end-to-end delay and packet loss ratio, respectively.

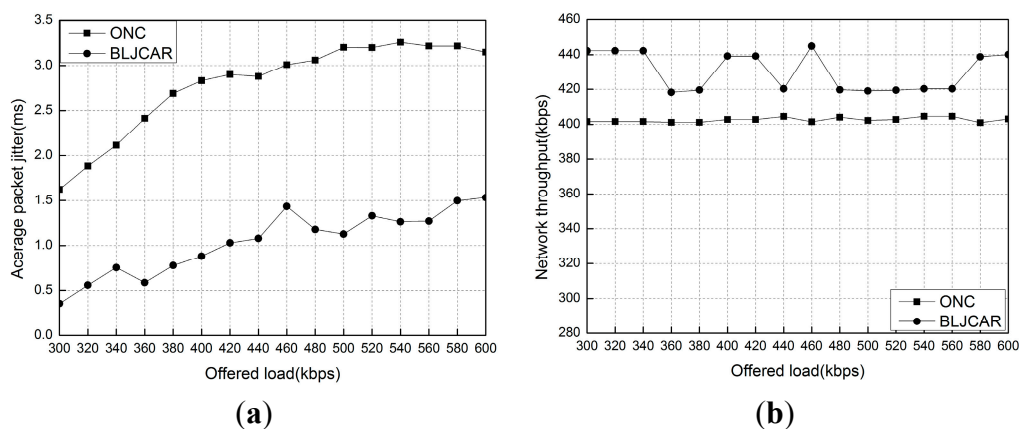


Figure 9. Cont.

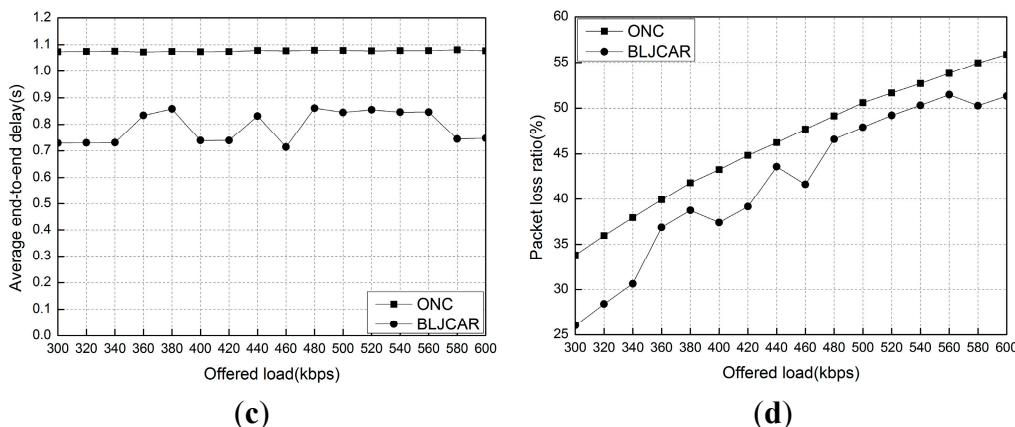


Figure 9. (a) Average packet jitter *versus* offered load; (b) Network throughput *versus* offered load; (c) Average end-to-end delay *versus* offered load; (d) Packet loss ratio *versus* offered load.

From Figure 9, it is seen that BLJCAR can improve the jitter, network throughput, average end-to-end delay and packet loss ratio as compared with the ONC policy.

5.4. Experiment 4

In this experiment, both flows have the same transmission rate. The rate of both flows varies from 400 kbps to 550 kbps with step of 10 kbps. Figure 10 shows the jitter performance of specific real-time flow, network throughput, average end-to-end delay and packet loss ratio due to both mechanisms *versus* offered load.

From Figure 10a, it is seen that BLJCAR can improve the jitter performance as compared with the ONC policy because it considers the queue state of coding node in packet coding algorithm and could better optimize the trade-off of effect of packet waiting delay and network coding on jitter, resulting in improved long-term jitter benefits.

From Figure 10b,d, it is observed that BLJCAR can improve the network throughput and packet loss rate as compared with the ONC policy because it delays packets to wait for the arrival of a future coding opportunity, which allows more data transmissions in each time unit.

From Figure 10c, it also shows that BLJCAR scheme outperforms the ONC policy in average end-to-end delay because the effective queue management mechanism of BLJCAR can result in improved jitter benefits, the transmission competition is greatly reduced and the queuing delay of packets is reduced.

Figure 10 shows that the capability of BLJCAR to cope with the longer communication paths is also fine for the single rate case.

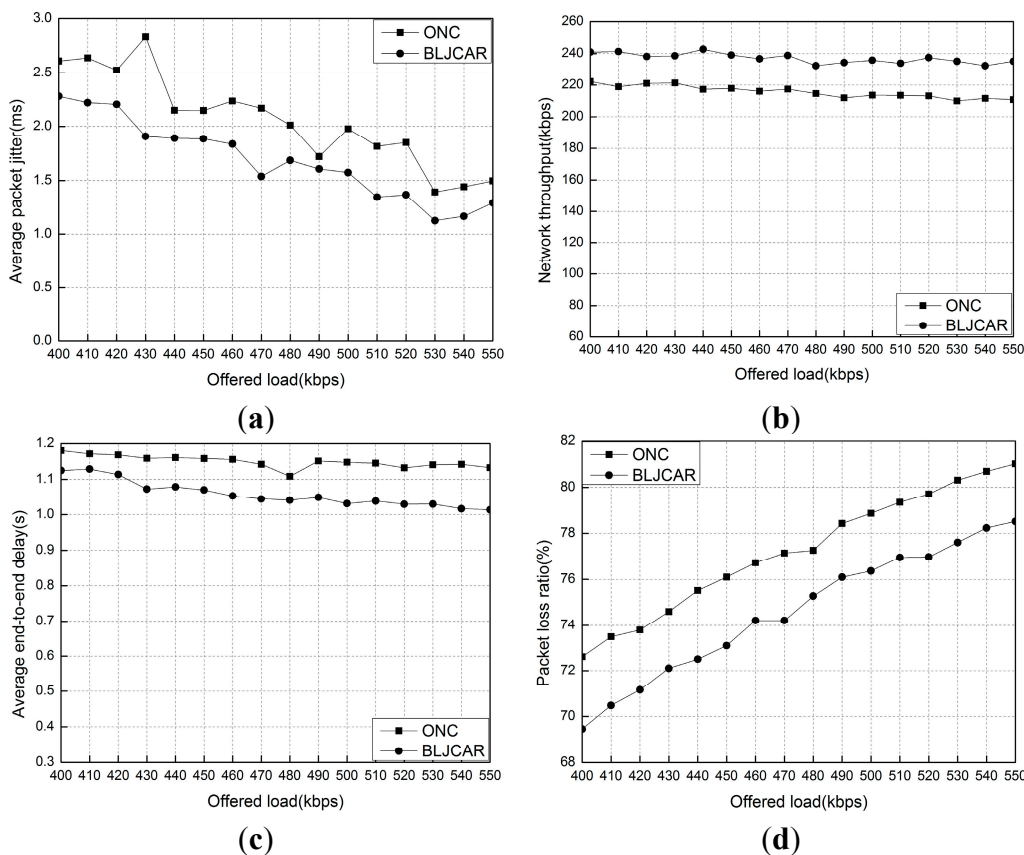


Figure 10. (a) Average packet jitter *versus* offered load; (b) Network throughput *versus* offered load; (c) Average end-to-end delay *versus* offered load; (d) Packet loss ratio *versus* offered load.

5.5. Experiment 5

In this experiment, the traffic load of flow 1 is twice the load of flow 2. We vary the rate of flow 2 from 270 kbps to 410 kbps with step of 10 kbps to test the jitter performance of flow 1, network throughput, average end-to-end delay and packet loss ratio of the two mechanisms. The results are shown in Figure 11.

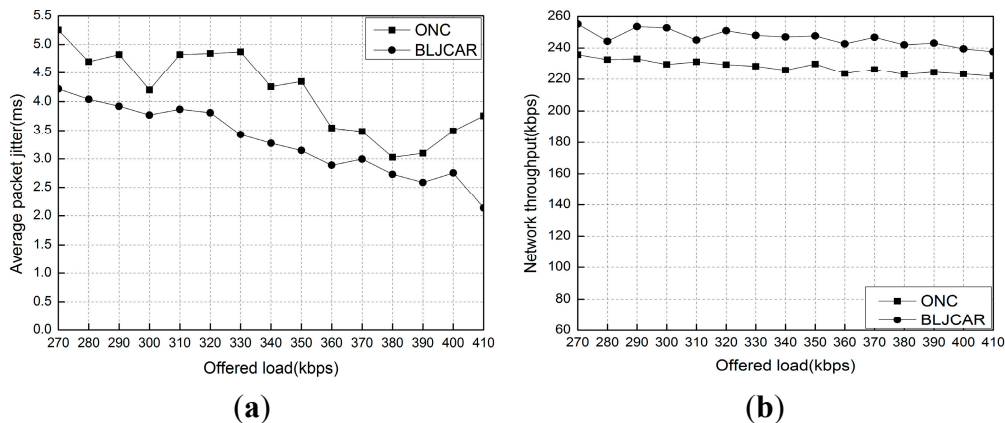


Figure 11. Cont.

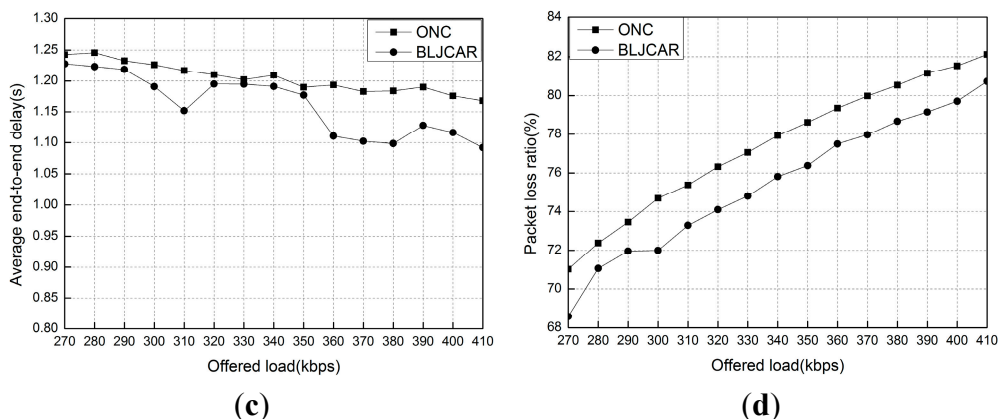


Figure 11. (a) Average packet jitter *versus* offered load; (b) Network throughput *versus* offered load; (c) Average end-to-end delay *versus* offered load; (d) Packet loss ratio *versus* offered load.

From Figure 11, it is observed that BLJCAR performs better than the ONC policy in jitter, network throughput, average end-to-end delay and packet loss ratio by considering the queue state of coding node in packet coding algorithm and the effective queue management mechanism. The effective queue management mechanism of BLJCAR can result in improved jitter benefits, alleviates congestion, reduces the queuing delay of packets and allows more data transmissions in each time unit.

Figure 11 shows that the capability of BLJCAR to cope with the longer communication paths is also fine for the multiple rate case.

5.6. Experiment 6

For a more thorough understanding of the performance of BLJCAR with longer communication path in multiple rate case, the rate of one of both flows is fixed. In this experiment, the rate of flow 1 is fixed to 400 kbps and the rate of flow 2 varies from 400 kbps to 550 kbps with step of 10 kbps. Figure 12a–d shows the jitter performance of flow 1, network throughput, average end-to-end delay and packet loss ratio, respectively.

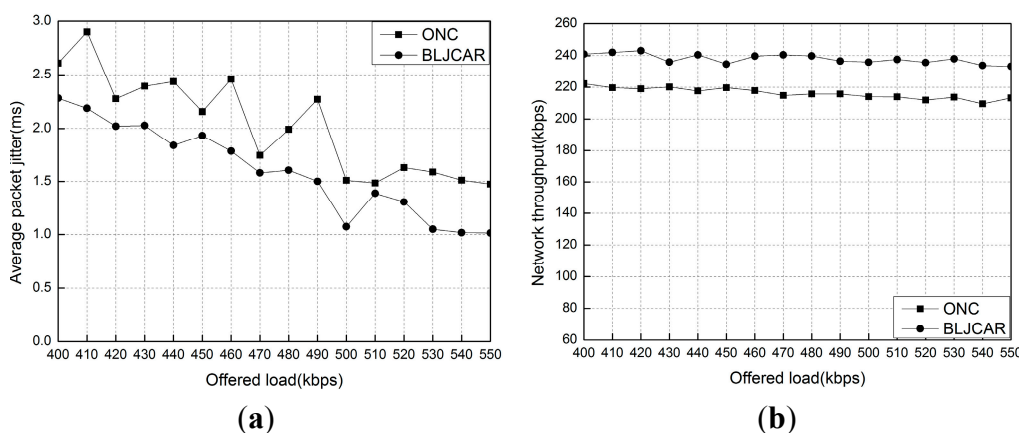


Figure 12. Cont.

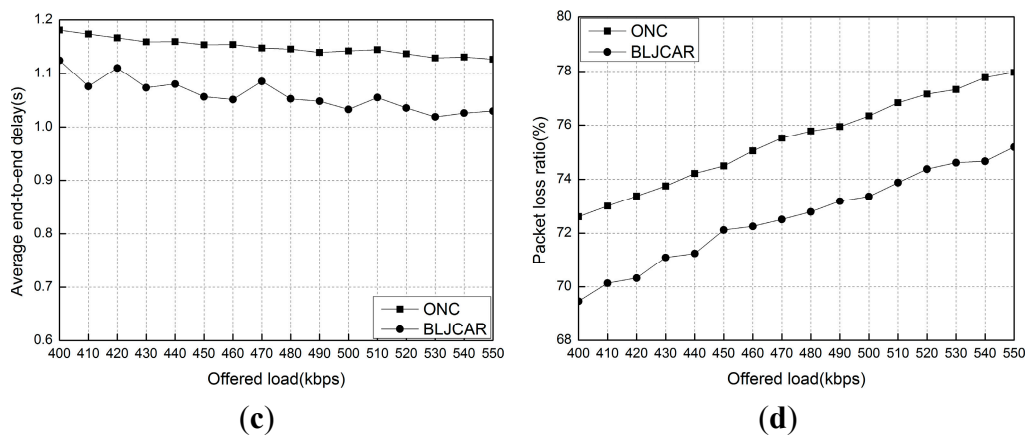


Figure 12. (a) Average packet jitter *versus* offered load; (b) Network throughput *versus* offered load; (c) Average end-to-end delay *versus* offered load; (d) Packet loss ratio *versus* offered load.

From Figure 12, it is seen that BLJCAR can improve the jitter, network throughput, average end-to-end delay and packet loss ratio as compared with the ONC policy.

6. Conclusions

In this paper, we present BLJCAR, a low jitter wireless transmission mechanism applied to network coding-aware routing algorithm. It solves the problem of “whether coding” at the coding node according to the queue-length based threshold policy. BLJCAR introduces the network jitter training phase after routing discovery and before data transmission to search the optimal threshold of the queue-length control policy. Because the regular ONC scheme can be seen as a special instance of the queue-length based threshold policy where the threshold is 0, the packet jitter can be significantly improved as compared with the ONC policy in different cases. Simulation results demonstrate the improved jitter performance by our scheme. Simulation results also show that our scheme can achieve higher network throughput, lower packet delivery delay and lower packet loss ratio. 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 BLJCAR and apply it in video transmission system. We are also exploring the problem of delay-constrained and jitter-optimized traffic allocation with network coding for multipath routing in multi-hop wireless network.

Acknowledgments

This work is supported by National Natural Science Foundation of China (No.61372069), National Defense Pre-research Foundation, Scientific Research Foundation for the Returned Overseas Chinese Scholars, State Education Ministry (JY0600090102), “111” project (No.B08038) and the Fundamental Research Funds for the Central Universities.

Author Contributions

Cunbo Lu contributed to the design of the proposed scheme, the setup of the simulation environment, the simulation data collection, writing of the primary draft of the paper. Liangtian Wan instructed his

research activities, and then contributed to refinement of the proposed scheme, the simulation result, and the advancement of the paper.

Conflicts of Interest

The authors declare no conflict of interest.

References

1. Ahlswede, R.; Cai, N.; Li, S.Y.R.; Yeung, R.W. Network information flow. *IEEE Trans. Inf. Theory* **2000**, *46*, 1204–1216.
2. Liu, J.; Goeckel, D.; Towsley, D. Bounds on the gain of network coding and broadcasting in wireless networks. In Proceedings of 26th IEEE International Conference on Computer Communications (INFOCOM), Anchorage, AK, USA, 6–12 May 2007; pp. 724–732.
3. Le, J.; Lui, J.C.S.; Chiu, D.M. How many packets can we encode?-an analysis of practical wireless network coding. In Proceedings of 27th IEEE International Conference on Computer Communications (INFOCOM), Phoenix, AZ, USA, 13–18 April 2008; pp. 13–18.
4. YuGupta, S.; Vazquez-Castro, M.A. Physical Layer Network Coding Based on Integer Forcing Precoded Computer and Forward. *Future Internet* **2013**, *5*, 439–459.
5. Katti, S.; Rahul, H.; Hu, W.; Katabi, D.; Medard, M.; Crowcroft, J. XORs in the air: Practical wireless network coding. *IEEE/ACM Trans. Netw.* **2008**, *16*, 497–510.
6. Ni, B.; Santhapuri, N.; Zhong, Z.; Nelakuditi, S. Routing with opportunistically coded exchanges in wireless mesh networks. In Proceedings of Second IEEE Workshop on Wireless Mesh Networks (WiMesh), Reston, VA, USA, 25 September 2006; pp. 157–159.
7. Yan, Y.; Zhao, Z.; Zhang, B.T.; Mouftah, H.; Ma, J. Rate-adaptive coding-aware multiple path routing for wireless mesh networks. In Proceedings of the IEEE Global Communications Conference (GLOBECOM), New Orleans, LA, USA, 30 November–4 December 2008; pp. 1–5.
8. Le, J.; Lui, J.C.S.; Chiu, D.M. DCAR: Distributed coding-aware routing in wireless networks. *IEEE Trans. Mob. Comput.* **2010**, *9*, 596–608.
9. Peng, Y.; Yang, Y.; Lu, X.; Ding, X. Coding-aware routing for unicast sessions in multi-hop wireless networks. In Proceedings of the IEEE Global Communications Conference (GLOBECOM), Miami, FL, USA, 6–10 December 2010; pp. 1–5.
10. Yue, H.; Zhu, X.; Zhang, C.; Fang, Y. CPTT: A high-throughput coding-aware routing metric for multi-hop wireless networks. In Proceedings of the IEEE Global Communications Conference (GLOBECOM), Anaheim, CA, USA, 3–7 December 2012; pp. 5687–5692.
11. Wang, W.; Wu, W.; Guan, Q.; Wang, J. TCAR: A new network coding-aware routing mechanism based on local topology detection. *J. Cent. South Univ.* **2014**, *21*, 3178–3185.
12. Shao, X.; Wang, C.; Rao, Y. Network Coding Aware QoS Routing for Wireless Sensor Network. *J. Commun.* **2015**, *10*, 24–32.
13. Shao, X.; Wang, R.; Huang, H.; Sun, L. Load Balanced Coding Aware Multipath Routing for Wireless Mesh Networks. *Chin. J. Electron.* **2015**, *24*, 8–12.

14. Zhang, J.; Fan, P. Optimal Scheduling for Network Coding: Delay vs. Efficiency. In Proceedings of the IEEE Global Communications Conference (GLOBECOM), Miami, FL, USA, 6–10 December 2010; pp. 1–5.
15. Chen, W.; Letaief, K.B.; Cao, Z. Buffer-aware network coding for wireless networks. *IEEE/ACM Trans. Netw.* **2012**, *20*, 1389–1401.
16. Hsu, Y.P.; Abedini, N.; Ramasamy, S.; Gautam, N.; Sprintson, A.; Shakkttai, S. Opportunities for network coding: To wait or not to wait. In Proceedings of the IEEE International Symposium on Information Theory (ISIT), Saint Petersburg, Russia, 31 July–5 August 2011; pp. 791–795.
17. Ramasamy, S. Delay-aware Scheduling in Wireless Coding Networks: To Wait or Not to Wait. Master's Thesis, Texas A&M University, College Station, TX, USA, December 2010.
18. Mohapatra, A.; Gautam, N.; Shakkottai, S.; Sprintson, A. Network Coding Decisions for Wireless Transmissions with Delay Consideration. *IEEE Trans. Commun.* **2014**, *62*, 2965–2976.
19. Yen, L.-H.; Cheng, Y.-M. Clustering coefficient of wireless ad hoc networks and the quantity of hidden terminals. *IEEE Commun. Lett.* **2005**, *9*, 234–236.

© 2015 by the authors; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/4.0/>).