

# Improving Unsegmented Network Coding for Opportunistic Routing in Wireless Mesh Network

Chen Chen<sup>†</sup>, Chao Dong<sup>†</sup>, Fan Wu<sup>‡</sup>, Hai Wang<sup>†</sup>, Laixian Peng<sup>†</sup>, Jingnan Nie<sup>†</sup>

<sup>†</sup>Institute of Communications Engineering, P.L.A. University of Science and Technology

<sup>‡</sup>Department of Computer Science and Engineering, Shanghai Key Laboratory of Scalable Computing and Systems, Shanghai Jiao Tong University  
{chchen2001, dch999, haiwang}@gmail.com, fwu@cs.sjtu.edu.cn

**Abstract**—Unsegmented network coding was incorporated into opportunistic routing to improve inefficient schedule of segmented network coding due to delayed feedback. However, most unsegmented network coding schemes fail to constrain the size of decoding window. Large decoding window not only challenges limited computational ability and decoding memory in practical environments, but also introduces large decoding delay, which is undesirable in delay sensitive applications. In this paper, we prove that the possibility of unacceptably large decoding window is innegligible, and verify the necessity of handling decoding window. To solve the issue, we argue that the number of unknown packets injected into the network must be strictly constrained at the source. Based on the idea, we improve unsegmented network coding scheme for opportunistic routing. Simulation results shows that the solution is robust to losses. In addition, to achieve optimal throughput, the redundancy factor should be selected larger than the reciprocal of end-to-end delivery ratio.

## I. INTRODUCTION

Traditional routing schemes in WMNs treat wireless links as point-to-point links. To deliver a packet from the source to the destination, the path along which the packet is handed off needs to determine in advance [1][2]. Apparently, traditional routing neglects the fact that nodes in WMNs communicate in the shared broadcasting wireless media. When links on the predetermined path turn poor, routing suffers severe losses. At the same time, packets transmitted in shared wireless media might be overheard by other nodes accidentally. Traditional routing regards these packets useless and simply discards those packets, though some of these packets may be useful for forwarding.

Opportunistic routing is put forward to exploit the broadcast nature of wireless media. To restrain redundant transmissions, a specific MAC layer schedule scheme is employed. The schedule, however, rigidly restrains flexibility of network [4]. The cooperation of network coding releases opportunistic routing from the MAC layer restraint while improves efficiency of transmission [7]. However, such scheme degrades in large scale networks due to inefficient schedule caused by heavily delayed feedback. Several solutions are proposed for improvement, of which representative ones are CodeOR [8], PipelineOR [9] and CCACK [10]. However, the problem is partially solved because of inherent shortage of segmented network coding they employ. Unsegmented network coding is proposed originally to compatibly protect packets from losses

in TCP Vegas [11]. Later, it is introduced into opportunistic routing and is proved to improve efficiency of schedule [12].

In existing unsegmented network coding schemes for reliable communication, the encoding window is slid strictly according to ACK on witness. In this case, the number of unseen packets injected into the network is constrained, but the number of unknown packets injected into the network is not. In random linear coding, the size of decoding window increases with the number of unknown packets combined together, which indicates that the size of decoding window is actually unconstrained. However, large decoding window is unacceptable in practical environments for three reasons. First, decoding a large chunk of packets one time challenges computational ability. The decoding computational complexity of random linear coding is  $O(k^3)$ , where  $k$  is the number of packets to be decoded [3]. Second, memory for decoding is limited in practical environments. Though total size of buffer could be somewhat large, memory assigned to one flow would be far less due to potentially numerous concurrent flows [14]. Third, large decoding window enlarges decoding delay, which is unqualified in delay sensitive applications. Thereof we come to the conclusion that the unsegmented network coding scheme must be improved.

In this paper, we investigate the probability distribution of the size of decode window for unsegmented network coding based opportunistic routing. We observe that the possibility of unacceptably large decoding window is innegligible, which to verify the necessity of handling decoding window. To constrain the size of decoding window, we argue that the number of unknown packets injected into network must be constrained at the source. Based on the idea, we improve unsegmented network coding scheme for opportunistic routing, and further implement it in Qualnet simulator. Our contributions are summarized as follows:

- We present the problem of unconstrained decoding window in unsegmented network coding for the first time. We further prove by theoretical analysis that the possibility of unacceptably large decoding window is innegligible.
- We argue that the number of unknown packets injected into network must be strictly constrained. Then the size of decoding window is constrained within the given bound.
- We conduct extensive simulation for analysis. Simulation

results show that to achieve optimal throughput, redundancy factor should be set larger than the reciprocal of end-to-end delivery ratio.

The remainder of this paper is organized as follows. Section II reviews related works. Section III analyzes the probability distribution of size of decoding window of unsegmented network coding based opportunistic routing. Improvement of unsegmented network coding scheme for opportunistic routing is described in section IV. Section V presents simulation results. Section VI concludes this paper.

## II. RELATED WORKS

Traditional routing in WMNs follows the conventional wisdom in wired networks to determine the path from the source to the destination. However, WMNs differ from wired networks in unstable links and shared media. To exploit these characteristics, ExOR takes advantage of opportunistic transmission on long-distance but lossy wireless links to improve throughput of unicast [4]. ExOR takes the ETX metric to select forwarders. To reduce redundant transmissions of multiple copies of a same packet, ExOR employs a specific MAC layer schedule based on synchronization of reception state of neighbors. However, such schedule increases complexity, decreases channel utilization and also restrains design of network. To further reduce redundancy, it is advised to select forwarders from nodes near the shortest path as forwarders [15].

Ahlsvede *et al.* presented the seminal work on network coding in [5]. It is proved that significant gains on throughput [16-19], delay [20] and reliability [21] can be achieved through network coding. MORE takes advantage of random linear coding to eliminate the need of MAC layer schedule since equally informative coded packets eliminate the need of redundant transmission of a same packet [7]. To perform random linear coding, MORE partitions original packets into segments of equal size and employs a simple stop-wait method to schedule segments. However, MORE exhibits poor efficiency in large scale networks, since delayed ACK makes the schedule inefficient. CodeOR utilizes a slide window to transmit multiple segments simultaneously, which reduces gaps between segments [8]. CodeOR also employs point-to-point ACK to accelerate schedule. In PipelineOR, forwarder initiates an ACK once it collects enough packets to decode a segment [9]. Therefore, the source may move to the next segment before an ACK is received from the destination. MIXIT performs network coding on symbol level [22]. MIXIT utilizes PHY's confidence to infer whether a bit is erroneous, and can take advantage of correct bits of corrupted packets. Both MORE and its variants employ segmented network coding, which degenerate in large scale networks.

In [11], an end-to-end scheme is proposed to employ unsegmented network coding to improve throughput in TCP Vegas. It is proved that reliability is guaranteed if the slide window is slid according to ACK on witness. To combat losses, totally  $\lfloor NR \rfloor$  coded packets will be transmitted if  $N$  packets are to be transmitted, where  $R > 1$ . SlideOR later incorporates unsegmented network coding into opportunistic routing to improve

inefficient schedule in segmented network coding [12]. Since packets in overlapping slide windows are encoded together, a coded packet from one window may be useful for decoding packets from another window. To accelerate dissemination of information, SlideOR employs an aggressive window strategy, which artificially creates difference of information between upstream node and downstream node. However, the cost is unguaranteed reliability. Existing unsegmented network coding schemes fail to constrain the size of decoding window, which is practically inapplicable.

## III. UNCONSTRAINED DECODING WINDOW

In this section, we analyze the probability distribution of the size of decoding window in unsegmented network coding based opportunistic routing. We are interested in the possibility of large decoding window, which is practically inadmissible.

Unsegmented network coding based opportunistic routing can be briefly described as follows. The source maintains a slide window size of  $W_{enc}$  for packets being transmitted, which can be represented by lower side sequence number  $l$  and upper side sequence number  $h$ . The source transmits random linear coded packets from the encoding window. The destination feedbacks an ACK on witnessing a new packet. Each ACK contains the minimum sequence number of unseen packets, denoted by  $s$ . Window operation at the source can be decoupled into lower side operation and upper side operation, which we say advancing and increasing window for convenience respectively. The source advances encoding window from  $[l, h]$  to  $[s, h]$  when it receives an ACK and  $s > l$ . The source increases encoding window to transmit new packets. We refer to the rate of increasing window as  $r_{win}$  and rate of transmission as  $r_{xmt}$ . To combat losses, redundant transmission is required, i.e.,  $r_{xmt}/r_{win} > 1$ .  $R = r_{xmt}/r_{win}$  is called redundancy factor. The destination decodes some packets once the number of seen packets  $N_{seen}$ , and the number of unknown packets sensed by it,  $N_{undec}$ , are equal.

In this section a single hop network with a pair of source and destination is considered, and ACK is assumed to be lossless and immediate. Note that both assumptions help to control decoding window. If multi-hop and imperfect ACK are considered, decoding will be put off and decoding window will be larger. We further assume that the Galois field for coding is sufficiently large, so that linear dependence can be ignored. Under this assumption, each coded packet that contains unknown packets increases  $N_{seen}$  by one. A slotted network is considered for simplicity, nevertheless the results hold for unslotted networks. Let  $p_{xmt}$  be the probability of transmitting a packet at the source,  $p_{win}$  be the probability of increasing the encoding window at source, and  $p_{rcv}$  be the transmission probability. Redundancy factor  $R = r_{xmt}/r_{win} = p_{win}/p_{xmt}$ .

Communication between the source and the destination can be regarded as a random process with regard to  $N_{undec}$  and  $N_{seen}$ , which can be depicted by a 2-dimension markov chain with regard to  $\langle N_{undec}, N_{seen} \rangle$ . We focus on the random process from the starting state, to the first decoding state. Subsequent decodings are omitted for similarity. In other

words, the markov chain starts at the state  $\langle 0, 0 \rangle$  and terminates at the states  $\langle N, N \rangle (N \neq 0)$ . To simplify model, we trick to equate  $N_{undec}$  to the number of packets regarded undecoded at the source. The modification dose not change the state transition since the two variables are synchronized at each successful transmission, and decoding occurs only at successful transmission.

We first define three transition probability in table I. Here  $p_1$  represents that the source increases encoding window by one while the destination dose not witness new packets in a slot.  $p_2$  represents that the source increases encoding window by one and the destination witnesses one new packet.  $p_3$  represents that encoding window is not increased while the destination witnesses one new packet.

TABLE I  
DEFINITION OF TRANSITION PROBABILITY

probability	value	state transition
$p_1$	$p_{win}(1-p_{xmt}p_{rcv})$	$\langle N_{undec}, N_{seen} \rangle \rightarrow \langle N_{undec} + 1, N_{seen} \rangle$
$p_2$	$p_{win}p_{xmt}p_{rcv}$	$\langle N_{undec}, N_{seen} \rangle \rightarrow \langle N_{undec} + 1, N_{seen} + 1 \rangle$
$p_3$	$(1-p_{win})p_{xmt}p_{rcv}$	$\langle N_{undec}, N_{seen} \rangle \rightarrow \langle N_{undec}, N_{seen} + 1 \rangle$

The markov chain can be decoupled into 8 parts with different transition patterns.

1)  $N_{undec} = 0, N_{seen} = 0$

The markov chain starts at this state. The state can transfer states  $\langle 1, 0 \rangle$  and  $\langle 1, 1 \rangle$ .

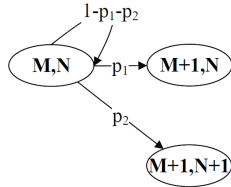


Fig. 1. state 1:  $N_{undec} = 0, N_{seen} = 0$

2)  $N_{undec} - N_{seen} = W_{enc}, N_{seen} = 0$

3)  $N_{undec} - N_{seen} = W_{enc}, N_{seen} > 0$

The 2nd and 3rd states represents that encoding window constrains the number of unseen packets. Therefore, these states never transfer to state  $\langle N_{undec} + 1, N_{seen} \rangle$ .

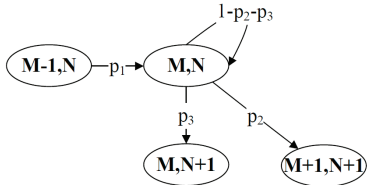


Fig. 2. state 2:  $N_{undec} - N_{seen} = W_{enc}, N_{seen} = 0$

4)  $0 < N_{undec} - N_{seen} < W_{enc}, N_{seen} = 0$

5)  $1 < N_{undec} - N_{seen} < W_{enc}, N_{seen} > 0$

6)  $N_{undec} - N_{seen} = 1, N_{seen} > 0$

In the 4th to 6th states, constraint by the encoding window is not reached, both  $N_{undec}$  and  $N_{seen}$  may increase.

7)  $N_{undec} = N_{seen}, N_{undec} = 1$

8)  $N_{undec} = N_{seen}, N_{undec} > 1$

The markov chain terminates at the 7th and the 8th state,

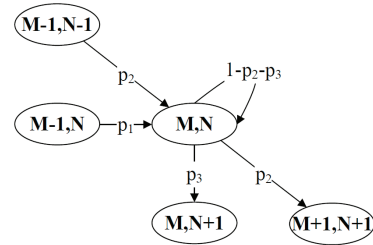


Fig. 3. state 3:  $N_{undec} - N_{seen} = W_{enc}, N_{seen} > 0$

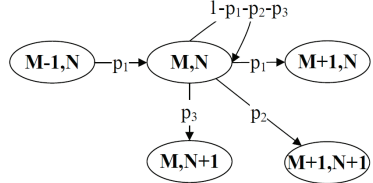


Fig. 4. state 4:  $0 < N_{undec} - N_{seen} < W_{enc}, N_{seen} = 0$

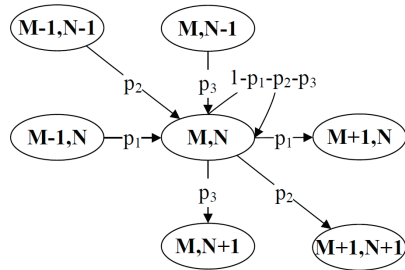


Fig. 5. state 5:  $1 < N_{undec} - N_{seen} < W_{enc}, N_{seen} > 0$

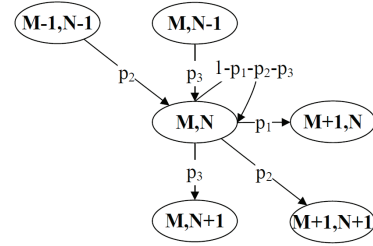


Fig. 6. state 6:  $N_{undec} - N_{seen} = 1, N_{seen} > 0$

where the number of unknown packets and the number of seen packets are equal at the destination.

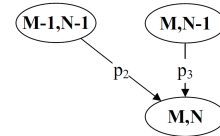


Fig. 7. state 7:  $N_{undec} = N_{seen}, N_{undec} = 1$



Fig. 8. state 8:  $N_{undec} = N_{seen}, N_{undec} > 1$

The markov chain is nonreversing, and can easily be solved by dynamic programming. Distribution of the size of decoding window can thus be calculated.

Let's take into account a maximum size of decoding window that is admissible. We say the decoding is unacceptable if

the size of decoding window exceeds the bound. In practice, the size of maximum decoding window would generally be small, as discussed in section I. Let's take 60 as the maximum size of the decoding window. As far as we know, none work employs larger decoding window. Fig. 9 and Fig. 10 show the probability unacceptable decoding window with  $R = 1.25$  and  $R = 1.5$  respectively when  $p_{rcv} = 0.8$ . It is shown that either smaller  $W_{enc}$  or larger  $R$  decreases the probability of unacceptable decoding window. With smaller  $W_{enc}$ , however, slide window scheme degrades. Note that with  $R = 1.25$ , redundancy and losses accurately counteract each other on average. [11] uses such  $R$  to protect packets from losses. In Fig. 9, approximately 7% decoding window exceeds the constraint with  $W_{enc} = 30$ . With  $R = 1.5$  as shown in Fig. 10 the probability drops to 2%, however, the cost is that 20% more packets might be sent.

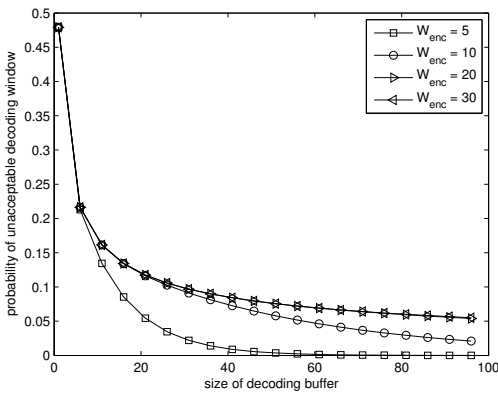


Fig. 9. probability of unacceptable decoding window with  $R = 1.25$

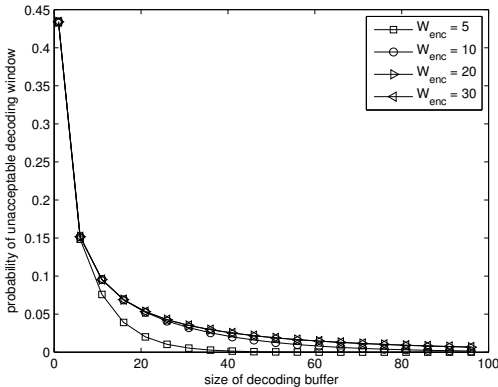


Fig. 10. probability of unacceptable decoding window with  $R = 1.5$

Our analysis shows that with reasonable size of encoding window and redundancy factor, the possibility of unacceptable decoding window is innegligible. The possibility will be even higher if multi-hop and imperfect ACK are considered. Therefore the necessity of handling decoding window is verified.

#### IV. IMPROVED UNSEGMENTED NETWORK CODING SCHEME

In this section we improve unsegmented network coding scheme for opportunistic routing with the purpose of handling

decoding window. The key idea of our solution is to constrain the number of unknown packets injected into network at the source. With the new constraint, the size of the decoding window never exceeds the given constraint. We denote the constraint of decoding window by  $W_{dec}$ . The size of encoding window is denoted by  $W_{enc}$ .

Our solution can be decoupled into four parts, which are encoding and decoding algorithm, ACK, window operations and redundancy policy.

A) *Encoding and Decoding Algorithm*: Assume that packets from a data stream are attached with consecutive sequence numbers, and the  $i$ -th packet can be represented by  $P_i$ . When the source want to transmit a packet, it randomly selects an encoding vector from a Galois field to linearly encode all packets in the slide window, i.e.,  $\sum_{i=l}^h \alpha_i P_i$ , where  $(\alpha_l, \alpha_{l+1}, \dots, \alpha_h)$  is the encoding vector, and  $l$  and  $h$  indicate the range of sequence number of packets in the slide window. Before the coded packet is transmitted, the coding vector is attached to the coded packet.

On receiving a coded packet, the forwarder checks whether the packets is innovative, i.e., whether the packet is linearly independent from previously received packets. Innovative ones are buffered while non-innovative ones are discarded. To forward, the forwarder reencodes all coded packets in its buffer similar with the source, attaches new coding vector and broadcasts the coded packet. Note that buffered packets with too many outdated original packets can be removed from buffer [12], so that the memory burden of forwarder can be alleviated.

The destination buffers innovative ones similarly with the forwarders. Once the number of seen packets catches up with the number of unknown packets, the destination can decode all unknown packets by Gauss elimination and simple matrix inversion [6].

B) *ACK*: Unsegmented network coding based opportunistic routing uses ACK on witness to advance encoding window. Each ACK contains the minimum sequence number of unseen packets,  $s$ . In addition, the maximum sequence number of yet decoded packets,  $d$ , is also contained in ACK. The destination transmits an ACK when it witnesses or decodes any packet. To avoid deadlock where the constraint is reached and ACK gets lost, redundant ACK will also be sent if certain number of innovative packets are continuously received.

C) *Window Operations*: Window operations include advancing and increasing encoding window, which relate to lower side and upper side of slide window. The source records and dynamically updates the number of seen packets and the number of unknown packets according to ACK, denoted by  $N_{seen}$  and  $N_{undec}$ . Consider a encoding window at  $[l, h]$ . When the source receives an ACK with  $s$  and  $d$ , the source advances the encoding window to  $[s, h]$  if  $l < s < h$ . The source increases the encoding window to transmit more packets. Before that, the source checks whether the encoding window is full and whether the number of unknown packets injected into the network reaches the given constraint. Encoding window is allowed to be increased only when  $N_{undec} - N_{seen} < W_{enc}$

and  $N_{undec} < W_{dec}$ .

D) *Redundancy Policy*: The source transmits coded packets at a higher rate than the rate of increasing encoding window to combat losses. The two rates are represented by  $r_{xmt}$  and  $r_{win}$  respectively. Redundancy factor is thus  $R = r_{xmt}/r_{win}$ . In slotted networks, for example, we can manipulate the probability of increasing encoding window and the probability of transmission for the purpose of redundancy. For other networks without slot structure, redundancy policy is similar, except that probability distribution of intervals between transmissions and between increasing encoding window is continuous.

## V. PERFORMANCE EVALUATION

### A. Simulation Setup

We use the Qualnet simulator to evaluate performance of our solution. A multi-hop network is constructed with network topology shown in Fig. 11. The variable on each link denotes the delivery ratio on the link. The 802.11b is used in the PHY layer. TDMA is used in the MAC layer. As an instance, we equate one hop delivery ratio to  $p_{1h}$ , two hop delivery ratio to  $p_{2h}$  and three hop delivery ratio to  $p_{3h}$ . We let  $p_{2h} = p_{1h} * \alpha$  and  $p_{3h} = p_{2h} * \beta$ , where  $0 < \alpha, \beta < 1$ . Average end-to-end delivery ratio,  $p_{e2e}$ , can thus be calculated. We define  $R^* = 1/p_{e2e}$  and relative redundancy factor  $R' = R/R^*$ .  $W_{enc}$  and  $W_{dec}$  are set to 30 and 60 respectively. We vary  $p_{1h}$  and get the following results on average of 10 runs.

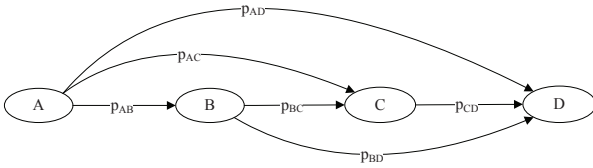


Fig. 11. network topology

### B. Simulation Results

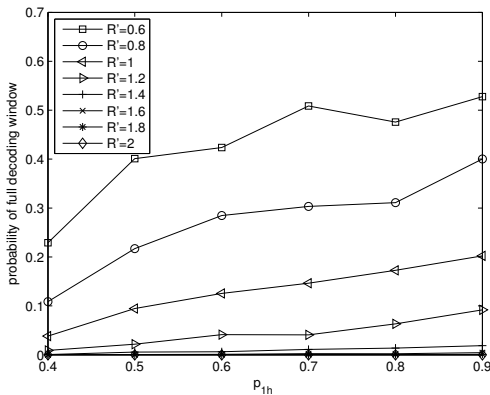


Fig. 12. probability of full decoding window

Fig. 12 shows the probability of full decoding window, which is equal to the probability of unacceptable decoding window if the constraint on the number of unknown packets is absent. It is interesting that with fixed redundancy factor, the

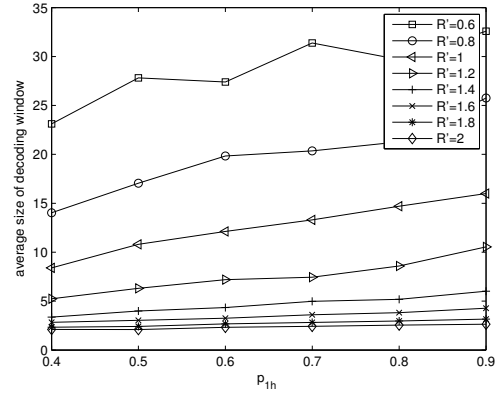


Fig. 13. average size of decoding window

probability decreases in lossy environments. This is because opportunistically forwarded packets possess higher reliability than ACK, and the difference of reliability between packets and ACK may hamper early decoding. In the case of lossy environment, the difference of reliability between coded packet and ACK decreases, which promotes earlier decoding. It can also be noticed that the value in this figure is higher compared with results presented in section III. For example when  $p_{1h} = 0.8$  and  $R' = 1$ , the probability is 15% in Fig. 12 while the probability is 7% in Fig. 9. The reason is that multi-hop topology and imperfect ACK put off decoding, which increases the size of decoding window. This figure shows that unacceptable decoding window exists with innegligible possibility, and verifies again the necessity of handling the decoding window. Average size of decoding window is shown in Fig. 13, which is consistent with Fig. 12.

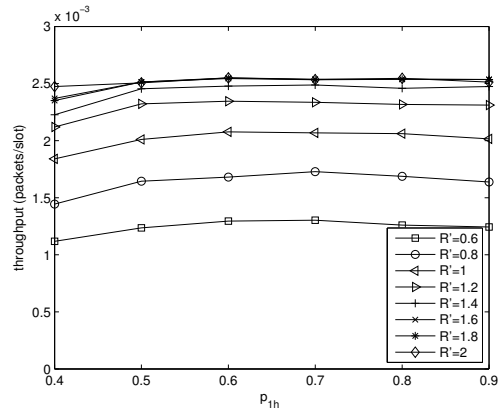


Fig. 14. average end-to-end throughput

Fig. 14 presents average end-to-end throughput which is defined as the number of packets decoded by destination per slot. Obviously, with larger redundancy factor, higher throughput can be achieved. An upper bound of throughput can be seen in the figure. If we continue to increase  $R'$  when  $R' > 1.4$ , the increment of throughput would be minor. It is also shown that with fixed redundancy factor the throughput maintains regardless of higher loss ratio, which shows robustness of the solution in lossy environments.

Fig. 15 focuses on the decoding delay while ignoring other

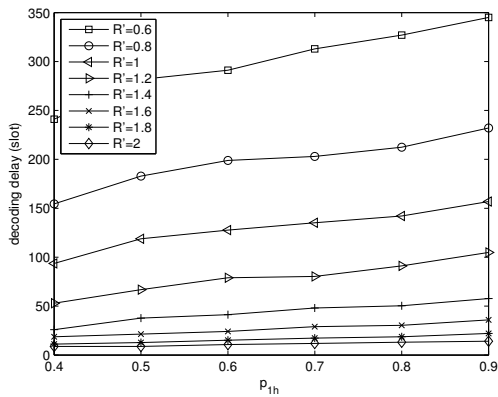


Fig. 15. decoding delay

parts of delay, which helps to concentratively study the impact on delay from network coding. It is shown that, decoding delay increases with the decrease of redundancy factor. The reason is that, with smaller redundancy factor, decoding window is more likely to grow up as shown in fig 13, and larger decoding delay is involved. Note that what Fig. 15 presents is average decoding Delay, and the maximum would even be larger.

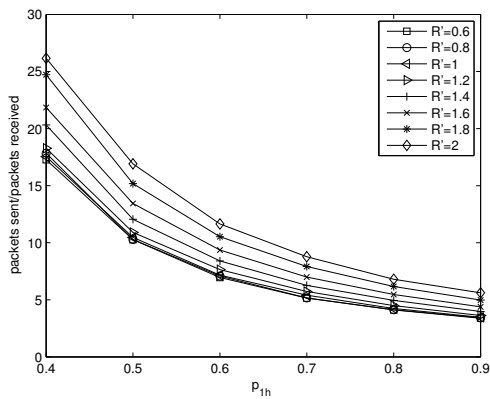


Fig. 16. redundancy

Overheads are shown in Fig. 16, which is defined as the number of packets sent by the source divided by the number of packets decoded by the destination. The overheads can also be interpreted as the redundancy factor in terms of amount of packets. Higher redundancy factor introduces higher overheads. With  $R' < 1.2$ , variation of overheads is tiny. The reason is that the given  $R'$  might be insufficient for decoding before the decoding window reaches the constraint as shown in Fig. 12, and further transmissions are required.

## VI. CONCLUSION

This paper shows that the possibility of unacceptably large decoding window could be innegligible in unsegmented network coding based opportunistic routing. To deal with the problem, we argue that the number of unknown packets injected into the network should be strictly constrained. Based on the idea, we improve unsegmented network coding scheme for opportunistic routing and implement the solution in Qualnet simulator. Simulation results show robustness of the protocol

in lossy environments, and provide guides for selecting redundancy factor.

## VII. ACKNOWLEDGEMENT

This work is supported by National 973 project of China under Grant No.2009CB3020402, National Natural Science Foundation of China under Grant No.61103224 and No.61170236, and Natural Science Foundation of Jiangsu Province of China under Grant No.BK2011118.

## REFERENCES

- [1] C. Perkins, E. Royer, and S. Das, Ad hoc On-Demand Distance Vector (AODV) Routing, RFC 3561, Category: Experimental, Jul. 2003.
- [2] T. Clausen and P. Jacquet, Optimized Link State Routing (OLSR) Protocol, RFC 3626, Category: Experimental, Oct. 2003.
- [3] Petar Maymounkov, Nicholas J. A. Harvey, and Desmond S. Lun, Methods for Efficient Network Coding, in Proc. of 44th Allerton Conference, 2006.
- [4] S. Biswas and R. Morris, ExOR: Opportunistic Multi-Hop Routing for Wireless Networks, in Proc. of ACM SIGCOMM, 2005.
- [5] R. Ahlswede, N. Cai, Li, S.Y.R. and R. W. Yeung, Network information flow, IEEE Trans. on Information Theory, vol. 46, no. 4, pp. 1204-1216, 2000.
- [6] T. Ho, M. Médard, R. Koetter, D. R. Karger, M. Effros, J. Shi, and B. Leong, A random linear network coding approach to multicast, IEEE Trans. on Information Theory, vol. 52, no. 10, pp. 4413-4430, 2006.
- [7] S. Chachulski, M. Jennings, S. Katti, and D. Katabi, Trading Structure for Randomness in Wireless Opportunistic Routing, in Proc. of ACM SIGCOMM, 2007.
- [8] Y. Lin, B. Li, and B. Liang, CodeOR: Opportunistic Routing in Wireless Mesh Networks with Segmented Network Coding, in Proc. of IEEE ICNP, 2008.
- [9] Y. Lin, C. Huang, and J. Huang, PipelineOR: A Pipelined Opportunistic Routing Protocol with Network Coding in Wireless Mesh Networks, in Proc. of IEEE VTC, 2010.
- [10] D. Koutsonikolas, C. C. Wang, and Y. C. Hu, CCAK: Efficient network coding based opportunistic rotting through cumulative coded acknowledgments, in Proc. of IEEE INFOCOM, 2010.
- [11] J. K. Sundararajan, D. Shah, M. Medard, M. Mitzenmacher, and J. Barros, Network Coding Meets TCP, in Proc. of IEEE INFOCOM, 2009.
- [12] Y. Lin, B. Liang, and B. Li, SlideOR: Online Opportunistic Network Coding in Wireless Mesh Networks, in Proc. of IEEE INFOCOM, 2010.
- [13] J. K. Sundararajan, D. Shah, and M. Medard, ARQ for Network Coding, in Proc. of IEEE International Symposium on Information Theory, 2008.
- [14] Y. Wang, H. Lu, P. Hong, and K. Xue, Practical Wireless Network Coding with Constrained Decoding Buffers, in Proc. of IEEE International Symposium on Network Coding, 2010.
- [15] E. Rozner, J. Seshadri, Y. Mehta, and L. Qiu, Simple opportunistic routing protocol for wireless mesh networks, in Proc. of WiMesh, 2006.
- [16] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard, and J. Crowcroft, XORs in the air: practical wireless network coding, in Proc. ACM SIGCOMM, 2006.
- [17] J. S. Park, and M. Gerla, D. S. Lun, Y. Yi, and M. Medard, CodecCast: a network-coding-based ad hoc multicast protocol, IEEE Journal on Wireless Communication, vol. 13, no. 5, pp. 76-81, 2006.
- [18] D. E. Lucani, M. Stojanovic, and M. Médard, Random linear network coding for time division duplexing: when to stop talking and start listening, in Proc. of IEEE INFOCOM, 2009.
- [19] H. Seferoglu, A. Markopoulou, and K. Ramakrishnan, I<sup>2</sup>NC: Intra-and Inter-Session Network Coding for Unicast Flows in Wireless Networks, in Proc. of IEEE INFOCOM, 2011.
- [20] A. Eryilmaz, A. Ozdaglar, M. Medard, On delay performance gains from network coding, in Proc. of IEEE 40th Annual Conference on Information Sciences and Systems, 2006, pp. 864-870.
- [21] F. C. Kuo, K. Tan, X. Y. Li, J. Zhang, and X. Fu, XOR rescue: exploiting network coding in lossy wireless networks, in Proc. of IEEE SECON, 2009.
- [22] S. Katti and D. Katabi, MIXIT: The Network Meets the Wireless Channel, in Proc. of the Sixth ACM Workshop on Hot Topics in Networks, 2007.