FENC: Fast and Efficient Opportunistic Network Coding in wireless networks

Peyman Pahlavani, Vali Derhami and Ali Mohammad Zareh Bidoki
Electrical and Computer Engineering Department, Yazd University, Yazd, IRAN
[e-mail: pahlavani@stu.yazduni.ac.ir, {vderhami, alizareh}@yazduni.ac.ir]
*Corresponding author: Vali Derhami

Received September 9, 2010; revised November 21, 2010; accepted December 23, 2010; published January 31, 2011

Abstract

Network coding is a newly developed technology that can cause considerable improvements in network throughput. COPE is the first network coding approach for wireless mesh networks and it is based on opportunistic Wireless Network Coding (WNC). It significantly improves throughput of multi-hop wireless networks utilizing network coding and broadcast features of wireless medium. In this paper we propose a new method, called FENC, for opportunistic WNC that improves the network throughput. In addition, its complexity is lower than other opportunistic WNC approaches. FENC utilizes division and conquer method to find an optimal network coding. The numerical results show that the proposed opportunistic algorithm improves the overall throughput as well as network coding approach.

Keywords: Network coding, wireless networks, division and conquer, throughput, complexity

This work was partially supported by Iranian Telecommunication Research Center (ITRC).

DOI: 10.3837/tiis.2011.01.003

1. Introduction

Network coding was first introduced in 2000 [1][2].We can broadly define network coding as allowing intermediate nodes in a network to either forward or combine the incoming independent flows. In this approach, each node that receives packets in each time slot from incoming edge can send combination (encode) of incoming packets to the outgoing edge (**Fig. 1**). This combination can be done in linear or non-linear manner.

COPE is the first network coding approach for wireless mesh networks [3]. Its implementation in TCP and UDP protocol was proposed in [4]. In this method, each wireless node listens to the neighbors and buffers overhearing packets. These packets later are used for decoding at receiver. Intermediate nodes first schedule a packet for sending then select sub set of incoming packets and apply XOR function to code with scheduled packet such that receiver can decode it. Receiver applies XOR function on received packets and then broadcast them.

Fig. 1. Network coding for wireless **(a)** Current approach and **(b)** Network coding approach.

To find the best encoded packets in COPE-wise approach, an intermediate node should find the best subset of incoming packets including most packets. The above mentioned method is very complex and leads to termination of packet deadlines, hence sender should send it again. Resending of packets decreases the overall throughput.

To the best of our knowledge, this paper is the first paper analyzing the complexity of cope-wise approaches and seeking to find a better coding decision. In this paper, we propose a new approach for network coding that finds the encoded packets with low complexity. We use division and conquer idea [5] to find network code. In order to find better encoded packets, the intermediate node partitions incoming packets into sets and finds best packets for encoding with scheduled packet in each set. Then it clusteres the sets so that each cluster includes two sets. For each cluster it combines packets selected for encoding in pervious step with selected packets of another set in the same cluster in the new set. For new sets algorithm again finds the best packets for encoding and clusters all sets and combines result of each cluster in new sets. This procedure goes on until a set remains. Ultimidatly, algorithm again finds the best packets for encoding upon the last set.

Simulation results show that the proposed algorithm increases throughput as compared with non–network code approach.

The paper is organized as follows. Section 2 discusses related work. An overview of the system model is given in Section 3. Section 4 presents our algorithms for network coding. Section 5 presents simulation results that demonstrate the advantages of the proposed algorithm over baseline schemes, in terms of sent data in intermediate node and application-level throughput. Finally, disscusion and conclusion remarks follows in Section 6.

2. Related Work

The network coding concept was introduced in articles [1][2]. They showed that mincut throughput of the network to each receiver could be achieved in multicast networks in which intermediate nodes do linear operations on incoming flows. The linearly combined packets can be used at the receivers to recover the original packets by solving a set of linear equations. This idea came from significant effort in several directions [6][7][8][9], including practical application of network coding, studying topologies beyond multicast, such as unicast [10][11][12] and broadcast scenarios. The broadcast nature of the wireless medium offers an opportunity for improving the throughput benefits of network coding [13][14].

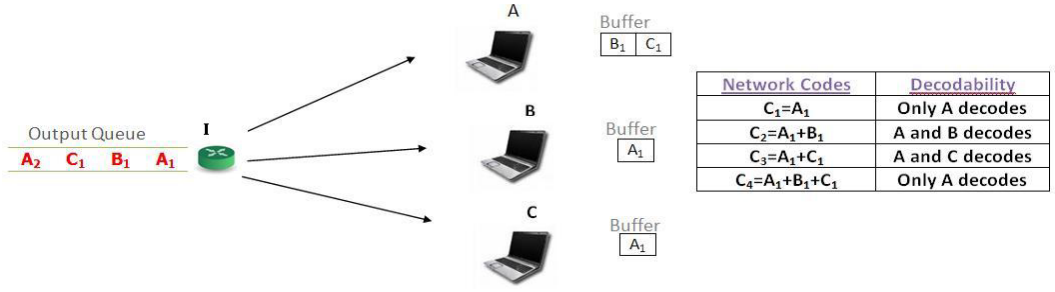
The recent work in [3][4] used the new ideas from the network coding in the context of wireless mesh networks. Authors in [3] implemented a pseudo-broadcast mechanism for IEEE 802.11 using opportunistic listening and a coding layer between IP and MAC. It is used to detect coding opportunities and XOR packets from different flows into a single transmission and increases the network throughput.

Authors in [15][16] present some COPE-wise methods that we call it Optimal Network Coding (OpNC) method. OpNC method is the same as COPE, but it chooses a code with maximum number of packets for XOR and uses other metric for selecting the best code among all possible combinations of network code which can be decodes in each packet destinations. Hence, OpNC chooses network code with maximum throughput. If there are several codes with the same maximum packet for XOR, one of them is selected at random. Since it should find all possible combination of network codes, the approach requires an exhaustive search for the best network code. The maximal number of combinations C_{max} is given by [16]:

$$C_{\max} = \sum_{i=1}^{k} \binom{k}{i} = 2^k - 1 \tag{1}$$

where K is the number of packet in queue. The intermediate node needs to calculate network code for $2^k - 1$ combinations. Then, it selects one of them with maximum packets which are decodable by receivers of packets. This can be significantly large computational overhead if the number of neighboring nodes is large. In the following, we propose a new approach that utilizes division and conquer method to find the best network code.

Our algorithm namely as, "Fast and Efficient Opportunistic Network Coding (FENC)", divides packets from incoming queue into sets and uses "OpNC" for each set and builds a tree structure. We will show that proposed algorithm has low complexity for selection of network code. not only does it improve throughput and the number of sent packets, but also it has low time order and decreases run time. In our approach, we use network coding method like COPE to increase throughput, but it seeks to select best candidate packets for combination. Our method is useful when there are large density of nodes and a lot of packets in intermediate queues.



Network Codes	Decodability
$C_1=A_1$	Only A decodes
$C_2=A_1+B_1$	A and B decodes
$C_3=A_1+C_1$	A and C decodes
$C_4=A_1+B_1+C_1$	Only A decodes

Fig. 2. Example of network coding, for a one-hop downlink scenario with three different streams.

3. System Model

Our solution is based on a networking model with cross topology in which there is an intermediate node surrounded by *K* neighboring nodes. Each neighbor sends packets to destination through intermediate node. We concentrate on the intermediate node. Our aim is selecting a subset of packets in output queue from the *K* neighboring nodes for which XORed packet is created and broadcasted. It is assumed that packets have been accumulated by the intermediate node and stored in output queue before the intermediate node transmission. Therefore, the source of the packet which is denoted by b_i, and destined to the node *i* from the intermediate node, is one of the neighboring nodes, *j* ($j \neq i$).

In XOR based WNC, each neighboring node recovers its packet using pre–knowledge on the received XORed packet. The simplest example for application of such WNC to the above networking model is when the intermediate node has bi–directional traffic flow, i.e., when the intermediate node has 2 packets: one destined to node *j* whose source is node *i*, the other destined to node *i* whose source is node *j*. Moreover, the opportunity for using WNC can be further increased if we utilize overhearing of each packet. Overhearing is defined as: when a packet is transmitted from each neighboring node to the intermediate node, this packet can also be received successfully by the other neighboring nodes with specific probability. One can be take advantage of such an overheard packet to increase the WNC opportunity for the intermediate node (see example blow). The intermediate node needs to know the overhearing situation of each neighbor in order to appropriately create a XOR packet which is decodable at the destination nodes. In this paper, we assume that such information can be obtained and controlled by a practical protocol, e.g., COPE the

proposed protocols in [3]. It is clear that the critical concept of the COPE-like design is to ensure that each node can get the status of neighbors' buffer to know that what packets the neighbors have. In practice, neighbors' buffer information is obtained through neighbors notifications with reception report and "guessing". Particularly, each node broadcasts reception reports listing the packets which it has stored. The problem with the reception reports is that these messages can be lost or, even more importantly, arrive too late for the coding purposes. Due to above mentioned points, each node may anticipate if a special packet has been received by a certain neighbor based on the delivery probability between that neighbor and the packet previous node, i.e., which neighbor node has sent the packet to the node. Moreover, the packet coding algorithm in COPE is based on the principle of never delaying packets whenever the wireless medium is available. The probability of successful overhearing depends significantly on the topology of neighboring nodes and the employed channel model among the nodes.

Example 1[15]: Consider the example shown in **Fig. 2**. The intermediate node *I* receives three independent streams, e.g., destined to its neighbors *A*, *B*, *C*. *I* maintains a FIFO queue that stores packets $\{A_1, A_2 \ldots\}$ destined to node *A*, $\{B_1, B_2, \ldots\}$ destined to node *B*, and $\{C_1, C_2, \ldots\}$ destined to node *C*. **Fig. 2**, also shows the contents of the buffers at each client: node *A* has overheard packets $\{B_1, C_1\}$ and nodes *B* and *C* have both overheard packet A_1, from previous transmissions. A_1 is the first packet from head of the queue and selected as the primary packet. Any packet in the output queue, other than A_1, can be chosen for combination with A_1, provided that the constructed network code has to be decoded at node *A*, i.e. A_1 can be retrieved. To satisfy this condition, combined packets which are used in the network code should already be available at node *A*. in other words, the decodability of a network code depends on the overheard packets at node *A*. Network codes like $c_1 = A_1$, $c_2 = A_1 \oplus B_1$, $c_3 = A_1 \oplus C_1$, and $c_4 = A_1 \oplus B_1 \oplus C_1$ can all be decoded by *A* and are eligible network codes, according. Node *A* can get packet A_1 from all four possible network codes. Codes c_2 and c_3 improve the throughput, It is clear that c_2 and c_3 are better codes than c_1 and c_4 in throughput because they have two packets which can decode in each packet destination. Hence OpNC algorithm chooses one of the c_1, c_2 randomly. This method can consider other metrics such as QOS metrics [15] to choose between c_1 and c_2.

4. FENC: Fast and Efficient Opportunistic Network Coding

The code construction problem is concerned with finding candidate codes so as to guarantee decodability by the target node. The code selection problem implies selecting a code among the candidate codes. The selction policy is obtained and the policy optimizesthroughput.

The proposed mehtod is called FENC. FENC uses the division and conquer idea. A division and conquer work through recursively breaking down a problem into two or more sub-problems of the same (or related) type, and it continues until these become simple enough to be solved directly. The solutions to the sub-problems are then combined to obtain solution for the original problem [5]. An early example of a division-and-conquer

algorithm with multiple sub-problems is Gauss's 1805 description of what is now called the Fast Fourier transform (FFT) algorithm [17]. Also This solution is used in high speed network switching [18].

4.1 FENC Description

Suppose the number of packets in Output queue is N. The FENC divides N packets to n sets. Each set includes m packets. FENC considers these sets as primary sets:

$$n = \frac{N}{\log_2 N}, m = \log 2N \tag{2}$$

For each set FENC runs OpNC to find the best combination of packets for encoding. Then, it clusters the sets and each cluster includes two sets. For each cluster FENC combines packets selected for combination in pervious step (OpNC) with selected packets of another set in same cluster in the new set. Then, the pervious steps run for new sets (which are created from clustering) until one set remains. Finally, OpNC runs on last set to get final combination. The following code shows the algorithm steps.

FENC algorithms steps:

I. Divides packets into sets.
II. Run the OpNC on each set and determine selected packets.
III. Cluster sets in two-member cluster.
IV. Combine selected packets on each set with selected packets with other set in the same cluster and put them in the new set.
V. If number of new sets are unique, run the OpNC algorithm on the last set otherwise go to step (II) for new sets.

In FENC algorithm, each primary set contains about $\log_2 N$ packets. The algorithm is iterative and in each iteration sets are combined together two by two and the new sets are made for the next iteration. Therfore, the number of sets in each iteration is half of the previous iteration. To better understand, we can consider the algorithm like a tree that the node in each level are the iteration set. Algorithm builds binary tree and each node in the tree is algorithm sets. The leaves of the tree is primary sets and root of the tree is final answer, and each level of tree construct of selected packets in the blow level.

Example 2: In **Fig. 3**, intermediate node wants to send packets from output queue to 8 neighbors. FENC assumes that intermediate node has pre-knowledge of the neighbor buffers. Considering Eq. (2), we have 3 primary sets, because N=8. Each set has 3 packets. Primary sets are $\{A_1, B_1, C_1\}$, $\{D_1, E_1, F_1\}$, $\{G_1, H_1\}$. In **Fig. 4**, you can see these sets in leaves of tree .OpNC runs for primary sets. The results of OpNC for primary sets are $\{A_1, B_1\}$, $\{D_1\}$, $\{G_1, H_1\}$. Then FENC clusters sets thus that each cluster includes two sets and combines the result of binary sets with each other in new sets. New sets are shown in **Fig. 4** in level two of tree. Therefore, we have two new sets $\{A_1, B_1,$ and $D_1\}$, $\{G_1, H_1\}$. For this sets FENC runs OpNC again and combines the obtaned result in new sets. Therefore, we

have one new set. The new set in root of tree is shown in **Fig. 4** along with these packets {A_1, B_1, G_1, and H1}. FENC runs OpNC for this last set and gets the result for coding. In **Fig. 4**, one can see algorithm steps in this example. Each node is a set. Leaves show primary sets and root shows final set. It can be seen that the selected combination of each node in each level is larger than children sets. In each node, OpNC runs on set. Packets for primary sets can be chosen randomly from output queue. When FENC chooses packets for primary sets randomly, each set has equal chance for selection. FENC can choose packets in the same sets in which destination node of packets are close to each other. This hueristic selection increases coding opportunity for that underlying set, because overhearing of packets for nodes that are near to each other is high.

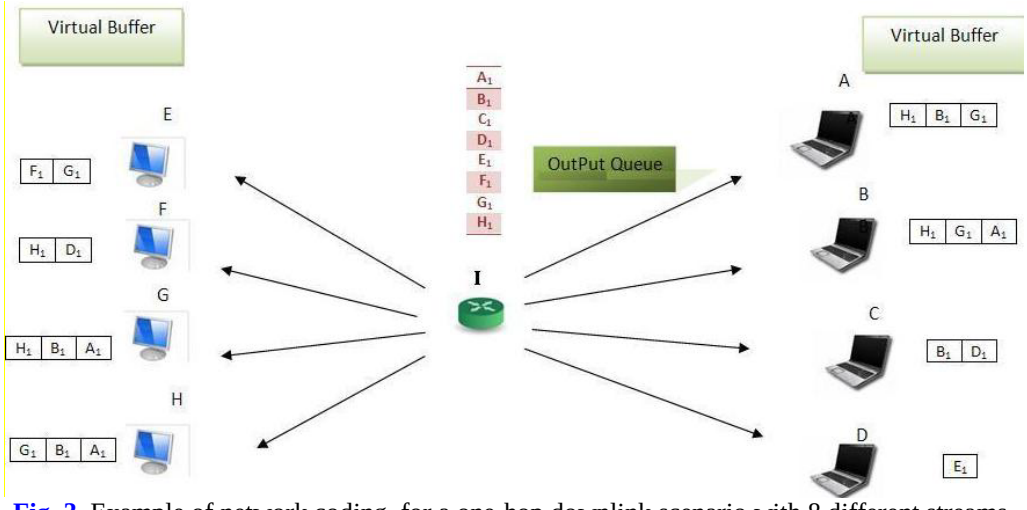


Fig. 3. Example of network coding, for a one-hop downlink scenario with 8 different streams.

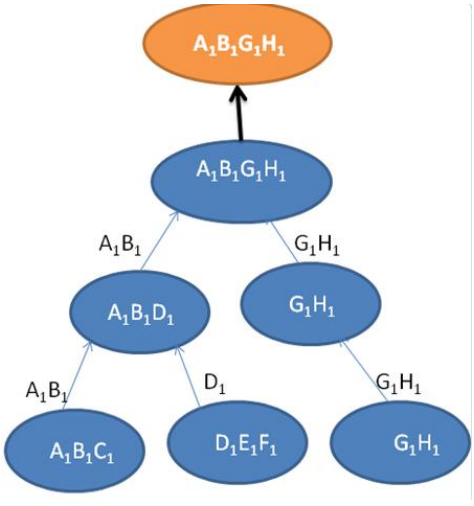


Fig. 4. The FENC algorithm steps in example 2.

To increas the chance to find optimal combination in large scale networks, FENC can be done more than once on output queue in intermediate node. In the above mentioned idea, FENC algorithm is run in each phase and results of the phase are considered as a set for the next the phase. This set and sets from output queue are considered in next phase. FENC is run till algorithm finds appropriate combination. In this paper,FENC runs one time on output queue.

4.2 Algorithm Complexity

FENC complexity calculates by multiplying number of sets in tree by complexity of running OpNC upon each set. First we find number of sets in tree. As far as Eq. (2) is considered, there are n primary sets which are considered as leaves of a tree. In the worst case, when tree is full, there are most sets (nodes) in the tree. Hence, the numbers of sets in each level are half of sets in down level (see **Fig. 5**).

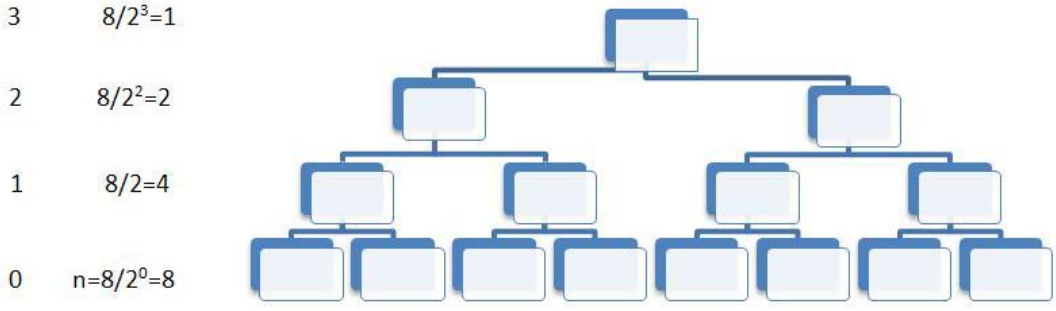


Fig. 5. A full binary tree.

If the numbers of sets (nodes) are denoted by S, then we have:

$$S = n + \frac{n}{2} + \frac{n}{2^2} + ... + \frac{n}{2^k} \tag{3}$$

There is one root in each tree, hence:

$$\frac{n}{2^k} = 1 \tag{4}$$

$$n = 2^k \tag{5}$$

$$k = \log_2 n \tag{6}$$

$$S = n + \frac{n}{2} + \frac{n}{2^2} + ... + 1 \tag{7}$$

Then, Eq. (7) can be solved simply as:

$$S = 2n - 1 \tag{8}$$

From Eq. (2), we have $n = \frac{N}{\log_2 N}$, hence:

$$S = 2\left(\frac{N}{\log_2 N}\right) - 1 \tag{9}$$

In order to find complexity of algorithm we have to find complexity of running OpNC upon each set. From Eq. (2) primary sets (leaves nodes) have $\log_2 N$ packets. We assume that the best combination of packets have no more than $\log_2 N$ packets. Therefore, others sets produced by combining of two primary sets have at most 2($\log_2 N$) packets in each set (nodes in tree). That is why, all of sets have no more than 2($\log_2 N$) packets, because the maximum combination in each set that found with OpNC, is ($\log_2 N$). As far as Eq. (1) is concerned, the complexity of OpNC is $O(2^k)$, whenever there are k neighbors. So the complexity of finding the best combination of packet for each set in worst case is:

$$2^{2(\log_2 N)} = N + 4 \tag{10}$$

Therefore, complexity of FENC calculates by multiplying number of sets in tree by complexity of running OpNC upon each set:

$$(N+4) \times (2\left(\frac{N}{\log_2 N}\right) - 1) \tag{11}$$

$$2\left(\frac{(N+4) \times N}{\log_2 N}\right) - N - 4 \cong O\left(\frac{N^2}{\log_2 N}\right) \tag{12}$$

It is obvious that $O\left(\frac{N^2}{\log_2 N}\right)$ has low complexity than $O(2^N)$.

5. Performance Evaluation

5.1 Simulation Setup

We use the NS-2.30 simulation environment [19] to implement the proposed algorithm and the baseline schemes. Simulation setup includes the underlying topologies, traffic scenarios, the MAC model, the wireless channel model, and the baseline algorithms used for comparison.

Baseline Algorithms for Comparison:

Our algorithm is compared with two algorithms as a baseline. OpNC described previously and No Network Coding (NoNC). NoNC algorithm is a FIFO output queue with no network coding.

Simulation Topologies:

Cross topology is used from simulation. In this topology, multiple crossing flows at an intermediate node are considered (see **Fig. 6**). The pairs of nodes *A*, *C* and *B*, *D* communicate over an intermediate node *I*, *e.g.*, *A* transmits to *C* and *C* transmits to *A* via *I*.

A single channel is used for both uplink and downlink transmissions. In this scenario, each node buffers a packet that has just been transmitted as well as all overheard packets. An illustrative example of cross topology is shown in **Fig. 6**. The intermediate node makes decisions on network coding and scheduling. It is assumed that nodes are placed on a circle with center *I* and radius 200 meters.

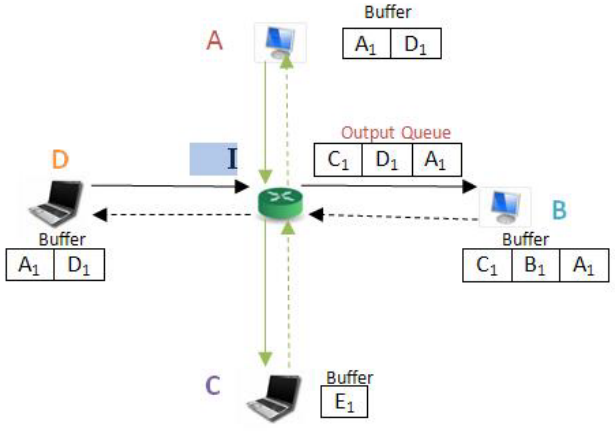


Fig. 6. Two-hop cross topology.

Grid Topology**:** In this topology, we consider the wireless mesh network (WMN) shown in **Fig. 7**. Nodes are distributed over a 300m×300m terrain according to a grid topology: the area is divided into 9 cells of equal size, 20 nodes are divided into 2 or 3 node sets randomly, and each set is assigned to a different cell. Nodes in a set are randomly placed within their assigned grid. Depending on the location of the receiving node *R*, sender transmitts packets directly (one-hop) or routed (two hops). If both intermediate node and *R* are either in the same cell or in neighboring cells, there will be a one-hop transmission; otherwise a node in the cell between *I* and *R* is selected as an intermediate hop. if there are more than one neighboring cells, one is selected randomly.

Wireless Channel Model:

Two-ray path loss model is a propagation path loss model using free space path loss for near sight and plan earth path loss for far sight.

MAC Model:

IEEE 802.11 is used in the MAC layer, with the subsequent modifications needed for network coding. First, to obtain the network coding benefits, we need a broadcast medium, which is hidden by the 802.11 protocol. Like to [1], we used the pseudo-broadcasting mechanism: packets are XORed in a single unicast packet, an XOR header is added for all nodes that should receive that packet, and the MAC address is set to the address of one of the receivers. A receiver knows whether a packet is destined to it from the MAC address or the XOR header. For simplicity, we use omniscient coding, described in [20] which employ a central component to provide each node with immediate and exact information on the packets known to every individual node. While this can be implemented in a simulator, it

would not be possible on real devices. Thus omniscient coding allows for perfect coding decisions, without the delay or overhead of reception reports or the risk of guessing wrong.

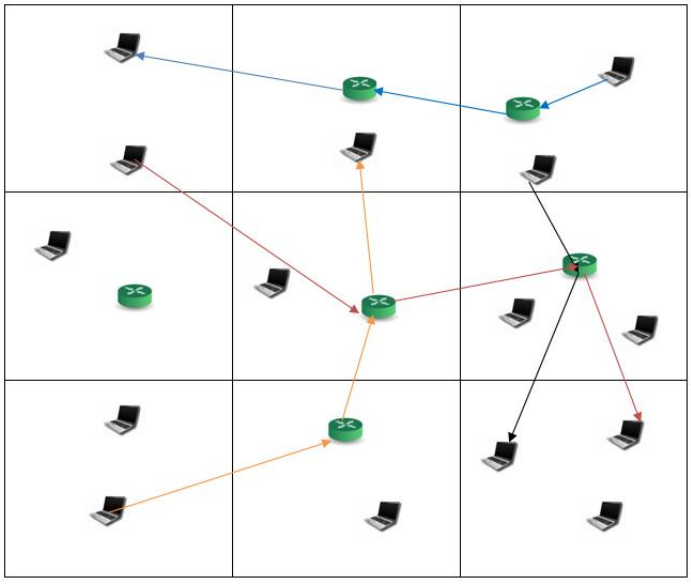


Fig. 7. Multi-hop grid topology.

5.2 Simulation Results

The algorithms are evaluated with transmitted coded data and throughput in intermediate node during simulation. **Fig. 8** and **Fig. 9** show transmitted and coded data for different coding algorithms in two topologies. It can be seen that with transmitted and coded data, OpNC is larger than other algorithms. Because this algorithm selects the best combination of packets to transmit that have most packets to code in intermediate node which can be decoded in receivers. Therefore, the number of transmitted packets will be increased in each transmission in intermediate node, thereby total throughput will increase. FENC uses network coding to increase transmitted packet in each transmission either. FENC tries to find the best combination packets as well as the OpNC. It can be seen that, FENC transmits coded data as well as OpNC's and act better than NoNC's. The reason for this advantage is that for each transmission, FENC codes some packets together and more receivers can decode this coded packet to recover desire packets.

In **Fig. 10** and **Fig. 11** we compare OpNC, FENC and NoNC approaches in term of the total throughput. Similar to **Fig. 8** and **Fig. 9** OpNC has the best throughout and FENC is near to OpNC. But NoNC approach has smaller throughput. Because OpNC and FENC use broadcast medium and send coded packet to more destinations in each transmission and each destination recovers packet destined to it from coded packet. Hence, sent packets and consequently received packets will increase and improve the overall throughput of system

in all network coding approaches. However, NoNC sends one packet to one destination in each transmission.

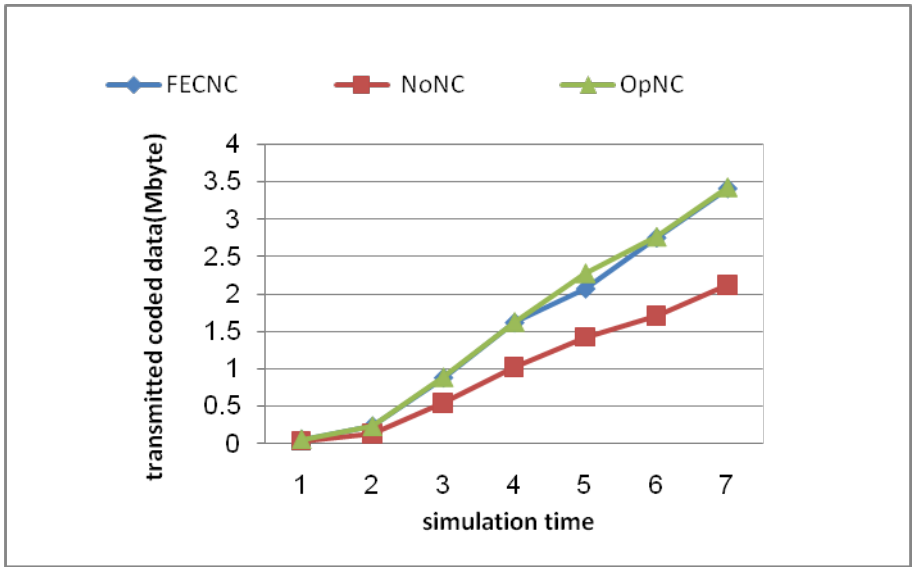


Fig. 8. Transmitted code data in intermediate node for XOR–based WNC with different coding algorithm in cross topology.

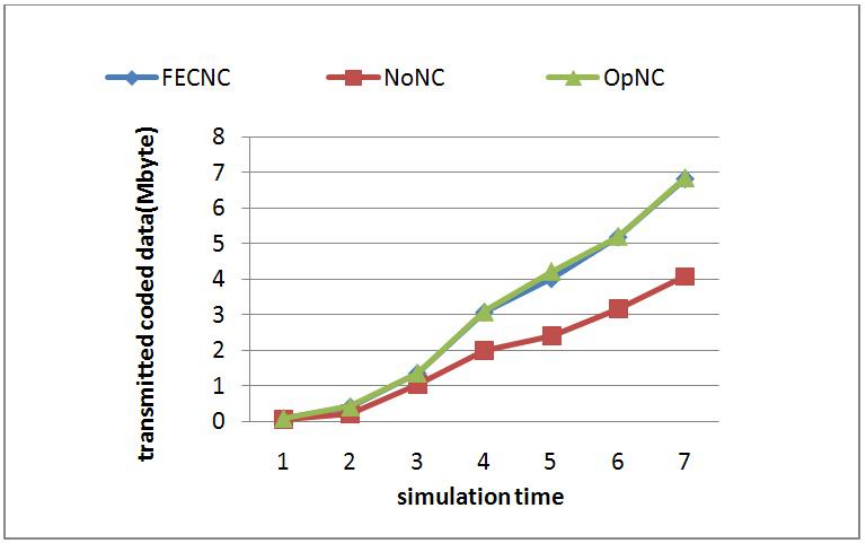


Fig. 9. Transmitted code data in intermediate node for XOR–based WNC with different coding algorithm in grid topology.

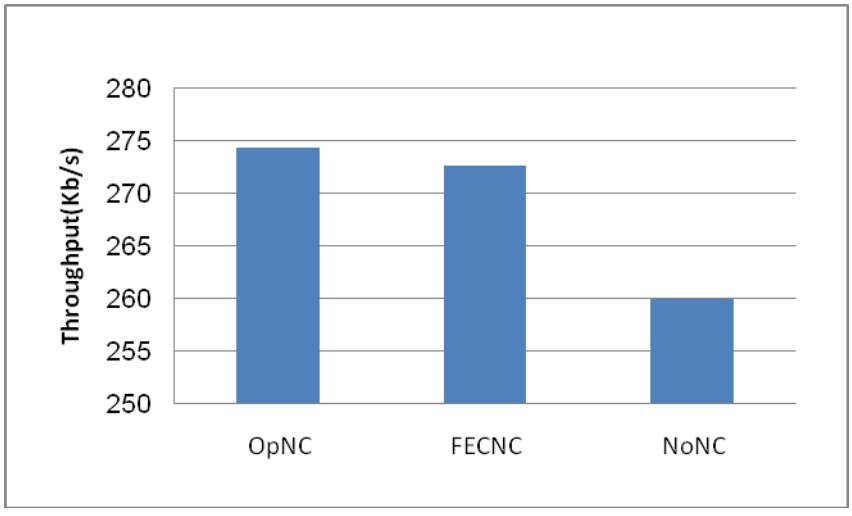


Fig. 10. System throughput for XOR–based WNC with different coding algorithm in cross topology.

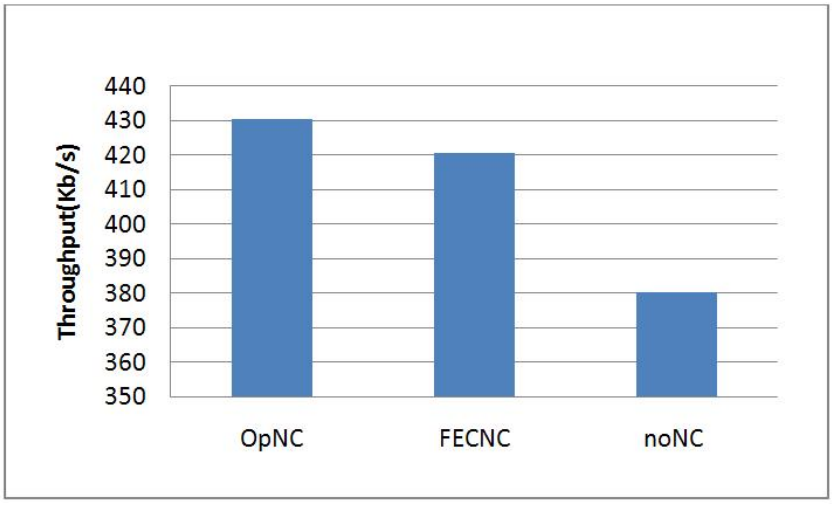


Fig. 11. System throughput for XOR–based WNC with different coding algorithm in grid topology.

6. Discussion and Conclusions

In the simulations, we showed that the throughput and coded data obtained by the proposed opportunistic algorithm (FENC) is comparable with OpNC approach.

The throughput depends on the number of packets selected for combination. FENC uses network coding and therefore it has more packet for combination. In order to increase the chance for FENC to select the best combination with more packets, it could select packets whose destination is close in the same primary set. Because probability of over hearing for such nodes are high.

When, there are many received packets in output queue, OpNC algorithm requires an exhaustive search to find the best combination. Therefore, the intermediate node has to spend much time to find the best combination. For example if there are 10 packets for different destinations in intermediate's output queue, OpNC searches to find the best combination, in addition it needs to produces 1023 different combinations. This high volum of combination leads to redused scheduling rate in ouput queue. Therefore, since receiving packets rate in output queue is high, in high node–density networks such as mesh networks, packet queue will be overflowed and lots of the packets will be dropped. On the contrary, FENC does not need to produce all combinations, therefore it is fast and can schedule more packets compared with OpNC at the same time.

In real-time traffic (such as video traffic) when packets wait lot of times in queue to schedule for sending, packets deadline will expire and sender should send them again or discard them. Therefore, when we use the algorithms with high complexity, lots of packets, will be dropped because of their deadlines reach. Consequently, real-time quality of service will drastically decrease.

In this paper, we proposed a practical opportunistic algorithm for WNC, called FENC. Essentially, our approach utilizes both ideas: networks coding for increasing throughput and division and conquer approach to decrease complexity. We proved that the proposed approach has lower complexity than OpNC. In addition, numerical results showed that the proposed scheduling can improve the overall throughput performance of WNC compared with the non–opportunistic approaches. Moreover, the obtained throughput in our approach is so close to OpNC throughput. Therefore, it can be concluded that the packets selected for combinations with FENC is so near to the best combinations in OpNC.

Studies to choose packets for primary sets in different topologies and also to run FENC more than one time to find better network code remains as a future work.

References

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

[2] S.-Y. R. Li, R. W. Yeung and N. Cai, "Linear network coding," *IEEE Transaction on Information, Theory*, vol. 49, no. 2, pp. 371–381, February, 2003.Article (CrossRef Link).

[3] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard and J. Crowcroft, "XORs in the Air: Practical Wireless Network Coding," *IEEE/ACM Transaction on Networking*, vol. 16, no. 3, pp.

497–510, June, 2008.Article (CrossRef Link).
[4] S. Katti, D. Katabi, W. Hu, H. Rahul and M. Medard, "The Importance of Being Opportunistic: Practical Network Coding For Wireless Environments," in *Proc. of 43rd Allerton Conference on Communication, Control, and Computing*, Monticello, September, 2005.
[5] G. Brassard and P. Bratley, "Fundamental of Algorithmics," Prentice-Hall, 1996.
[6] The network coding webpage. http://www.networkcoding.info/
[7] C. Fragouli. J. Widmer and J.Y. LeBoudec, "Network coding: an instant primer," in *Proc. of ACM SIGCOMM CCR*, vol. 36, no. 1, pp. 63-68, January, 2006.Article (CrossRef Link).
[8] P. A. Chou, Y. Wu and K. Jain, "Practical network coding," in *Proc. of 41st Annual Allerton Conference on Communication, Control, and Computing*, Monticello, October, 2003.
[9] P. A. Chou and Y. Wu, "Network coding for the Internet and wireless networks," *IEEE Signal Magazine*, vol. 24, no. 5, pp. 77-85, September, 2007.Article (CrossRef Link).
[10] Z. Li and B. Li, "Network coding: The case for multiple unicast sessions," in *Proc. of Allerton Conference*, Illinois, September, 2004.
[11] Y. Wu, P.A. Chou and S. Y. Kung, "Information exchange in wireless network coding and physical layer broadcast," in *Proc. of IEEE CISS*, Baltimore, March, 2005.
[12] T. Ho and R. Koetter, "Online incremental network coding for multiple unicasts," in *DIMACS WG on Network Coding*, Piscataway, January, 2005.
[13] S. Deb, M. Effros, T. Ho, D.R. Karger, R. Koetter, D.S. Lun, M. Medard and N. Ratnakar, "Network coding for wireless applications: A brief tutorial," in *Proc. of IWWAN*, London, May, 2005.
[14] A. Ramamoorthy, J. Shi and R. Wesel, "On the capacity of network coding for wireless networks," *IEEE Transaction on Information Theory*, vol. 51, no. 8, pp. 2878-2885, August, 2005 . Article (CrossRef Link).
[15] H. Seferoglu and A. Markopoulou, "Video-Aware Opportunistic Network Coding over Wireless Networks," *IEEE Journal on Selected Areas in Communications*," vol. 27, no. 5, pp. 1-16, June, 2009.Article (CrossRef Link).
[16] H. Yomo and P. Popovski, "Opportunistic Scheduling for Wireless Network Coding," *IEEE Transaction on Information Theory*, vol. 8, no. 6, pp. 2766-2770, June, 2007. Article (CrossRef Link).
[17] M. T.Heideman, D. H. Johnson and C. S. Burrus, "Gauss and the history of the fast Fourier transform," *IEEE ASSP Magazine*, vol. 4, no.1, pp. 14–21, October, 1984.Article (CrossRef Link).
[18] A.M. Zareh, V. Azhari and N. Yazdani, "A high speed Logarithmic Scheduling Algorithm for Input-Queued Switches," *Journal of Computer Communication in Elsevier*, vol. 31, no. 1, pp. 5-18, January, 2008. Article (CrossRef Link).
[19] The NS-2 network simulator. http://www.isi.edu/nsnam/ns/
[20] B. Scheuermann, W. Hu and J. Crowcroft, "Near-optimal co-ordinated coding in wireless multihop networks," in *Proc. of the ACM CoNEXT conference*, New York, December, 2007. Article (CrossRef Link).

Peyman Pahlavani born in Zanjan, Iran, received a B.S. degree in information technology from Institute for Advanced Studies in Basic Science (IASBS), Zanjan, Iran, in 2008. He also received MSc in Computer Network from Yazd University in 2010. His research interests include network coding and wireless networks.

Vali Derhami born in 1973 in Yazd, Iran, received the B.S. degree in control engineering from Esfahan University of Technology, Iran, in 1996. He received M.S. and Ph.D. degrees in control engineering from Tarbiat Modares university, Iran, in 1998 and 2007, respectively. Currently, he is a assistant professor in Computer and Electrical Engineering Department in Yazd University. His research interests are computer networks, machine learning, neural fuzzy systems, intelligent control, reinforcement learning, robotics, and information technology.

Ali Mohammad Zareh Bidoki got his B.S. degree in Computer Engineering Isfahan University of Technology in 1999. He also received MSc in Computer Architecture from University of Tehran in 2002. Furthermore he graduated as a Ph.D. in Computer Engineering at University of Tehran in 2009. His research interests include Web information retrieval, search engines and data mining. He has published more than 20 papers in international journals and conferences. Currently, he is an assistant professor in Yazd University.