Increasing Coding Opportunities Using Maximum-Weight Clique

Hootan Zhian, Masoud Sabaei, Nastooh Taheri Javan
Department of Computer and IT Engineering
Amirkabir University of Technology
Tehran, Iran
{hootan.zhian, sabaei, nastooh}@aut.ac.ir

Omid Tavallaie
Department of Computer and Electrical Engineering
Graduate University of Advanced Technology
Kerman, Iran
omid.tavallaie@ieee.org

***Abstract*—Network coding is used to improve the throughput of communication networks. In this technique, the intermediate nodes mix packets to increase the information content of each transmission. For each flow, a coding pattern is defined as a set of flows that can be coded together. Finding a suitable coding pattern is a challenge due to much complexity. In this paper, we propose an algorithm to find a suitable coding pattern in intermediate nodes by mapping this problem onto maximum-weight clique. Also, we described time complexity of our algorithm in details. Simulation results show that our proposed method can achieve better performance in terms of throughput and end-to-end delay by increasing coding opportunities.**

***Index Terms*— Coding Opportunity, Coding Pattern Wireless Networks, Network Coding.**

I. INTRODUCTION

Network coding is gaining popularity as a mechanism to increase the throughput of both wired and wireless networks. It was proposed for the first time in 2000 by Ahlswede et al [1]. In the case of wireless networks, network coding is performed by considering the broadcast nature of wireless communication in order to increase the information content per transmission. Fig.1 shows the impact of network coding on wireless networks. In this scenario, node *A* wants to send packet P_1 to node *B*, and node *B* wants to send packet P_2 to node *A*. These two nodes communicate with each other through node *C*. By considering network coding, node *C* can mix packets P_1 and P_2 together and then send packet $P_1 \oplus P_2$ to Nodes *A* and *B* by one transmission. Nodes *A* and *B* have packets P_1 and P_2 respectively. So they can decode their desired packets from $P_1 \oplus P_2$. In this scenario, we use three transmissions instead of four to deliver packets to their destinations.

In this paper we will focus on the problem of finding a suitable coding pattern. For each flow, we define a coding pattern as a set of flows that can be coded together and we try to find the patterns that have more flows. So, we attempt to increase the number of encoded packets in one transmission in order to reach higher throughput.

Fig. 1. An example of network coding

We show the impact of selecting coding pattern on throughput by an example in Fig. 2. Through node *R*, nodes *A*, *B*, *D*, *E* and *F* send packets of flows F_1, F_2, F_3, F_4 and F_5 to nodes *D*, *E*, *A*, *B* and *C* respectively. These packets are shown by P_1, P_2, P_3, P_4 and P_5 in Fig. 2. Dotted lines between nodes demonstrate the transmission range between them. Node *R*, can encode P_1 with two coding patterns before sending it:

1. $P_1 \oplus P_3 \oplus P_5$
2. $P_1 \oplus P_4$

The best coding pattern is the first one since it uses two transmissions instead of three. It encodes and sends packets ($P_1 \oplus P_3 \oplus P_5$) and ($P_2 \oplus P_4$) in two transmissions. If node R uses the second pattern, It needs three transmissions for sending packets ($P_1 \oplus P_4$), ($P_3 \oplus P_5$) and P_2. So, we can decrease the number of transmissions by choosing the pattern that has more flows.

In [2-4], the authors proposed methods based on the two-hop coding structure [5]. The approach of these studies for selecting patterns is as follows: each node puts packets of different flows in virtual queues based on their next hops. In coding process, each node uses round-robin or random to select queues and check its coding conditions with the packet

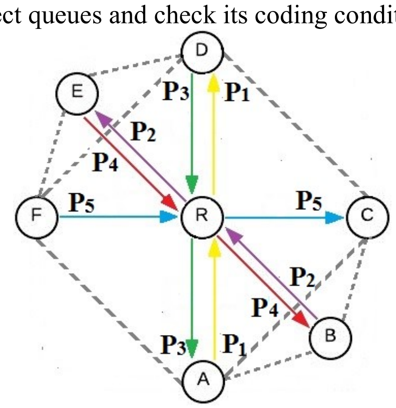


Fig. 2. Effect of selecting coding pattern on throughput.

978-1-4799-0383-2/13/$31.00 ©2013 IEEE

that must be sent. Although coding structure in these approaches is limited within a two-hop region. [5-8] proposed coding aware routing algorithms. These methods have been developed in order to omit [2-4] boundaries. In [5], coding pattern selection is modeled to complete sub-graph and each flow is placed in only one coding pattern. Round robin and random approaches are used in [6]–[8] in order to find the set of packets that can be encoded with each other.

Many researchers tried to analyze the performance of network coding [2], [9] and [10]. In COPE[2] and [9], the encoding number is unbounded. Whereas in [10], there is an upper bound for packets that have ability to be encoded. In [11], the problem of finding the biggest coding pattern is modeled to *stable sets* which is a NP-hard problem. In [13], it is proved that finding maximum complete sub-graph is a NP-complete problem.

In this paper, we map the problem of selecting suitable coding pattern to the problem of selecting maximum weight-clique. Also we focus on selecting a suitable pattern to encode each flow in encoder nodes. Our aim is enhancing the network throughput. For reaching this, encoder nodes will try to select longer patterns in order to encode more packets together. In order to specify coding pattern of input flow, we make a new graph similar to coding graph in [5] and dedicate a weight to each vertex. We select the best coding pattern by using a maximum weight-clique algorithm. The proposed algorithm can be applied in multi-hop or two-hop coding structure as [5] and [2] respectively. In implementation phase, we compare our approach with other methods and we show that it can improve performance in terms of throughput and end-to-end delay.

The rest of this paper is organized as follows. Section II briefly discusses related works in network coding and problem of finding the coding pattern. A brief background about maximum-weight clique and vectors scalar product is mentioned in section III. We discuss the detail and complexity of our algorithm in section VI and V. Simulation results are provided in section IV and finally we conclude our paper in section VII.

II. Related Works

Network coding firstly proposed by Ahlswede and his assistants in 2000[1] as a way to increase throughput especially in wireless networks. The approach of [2-4] studies in selecting pattern is as follows: packets belong to each flow in encoder node are put in a virtual queue based on their next hop. In this approach, only one packet is selected from each queue and its coding condition is studied. Each node uses a random permutation for selecting the queues and checking the coding conditions. These tasks will be repeated for each transmission. There are coding assumptions in [2-4] methods. For coding N packets together, following conditions is expected. P_i is the probability that a next hop node has heard the packet i and P_D indicates the probability that a native packet can be decoded. i.e.,

$$P_D=P_1\times P_2\times\ldots\times P_{n-1}. \quad (1)$$

If P_D for all N next hop nodes be greater than 0.8, the required condition for Xor operation of nth packet is ready.

Coding aware routing algorithms have been developed in order to ending COPE boundaries. These algorithms choose a route that has more coding conditions. The methods proposed in [5]–[8] use RR and random techniques to find coding pattern.

DCAR (Distributed Coding Aware Routing protocol) is proposed by li and his assistants in 2010. This protocol is able to discover all routes and identify all coding opportunities on the routes. This protocol can distinguish the routes with higher coding chance in order to solve COPE boundaries.

[5] Uses Round Robin to find a coding pattern. By receiving a new flow, a coding graph is drawn. Every vertex of this graph is equivalent to a flow and the edge between two nodes shows the coding condition between flows. After creating the graph, the available vertices in coding graph are randomly chosen and maximum complete sub-graph is considered as the coding pattern. This will be performed over and over until there is no vertex in coding graph. In [5], it is not shown how to find a complete sub graph and complexity of coding graph is not discussed. As we mentioned, in this method each flow only can participate in one coding pattern and a random strategy is used for selecting each vertex which can lead to lose chances.

The necessary conditions to code the packets in multi-hop structure are determined in [5-8]. We use following symbols for conceptual and official definitions:

- n shows a node
- $NS(n)$ shows a set of one-hop neighbors of node n
- FL shows a flow
- $n \in FL$ shows that node n is on the route of FL
- $U(n, FL)$ shows a set of upstream nodes of node n in the route of FL
- $D(n, FL)$ shows a set of downstream nodes of node n in the route of FL

If Fl_1 and Fl_2 intersect each other in a node likes M, the required condition for coding Fl_1 and Fl_2 in node M is shown as below.

1. There is a node like dsn_1 where $D(M, Fl_1) \in dsn_1$, $usn_2 \in U(M, Fl_2)$ and $dsn_1 \in NS(usn_2)$ (2.a)
   or
   $dsn_1 \in U(M, Fl_2)$ (2.b)
2. There is a node like dsn_2 where $D(M, Fl_2) \in dsn_2$, $usn_2 \in U(M, Fl_1)$ and $dsn_2 \in NS(usn_1)$ (3.a)
   or
   $dsn_2 \in U(M, Fl_1)$ (3.b)

There are a lot of works related to analysis of network coding. In [9] performance of COPE is studied. In this study, number of nodes that can be replaced next to the relay node is supposed to be infinite. For instance, in considered scenario in Fig. 3.a, transmitted packets at the head of each diagonal as I can be heard by all nodes, except that destination node J. So, encoder node can send infinite packets in one transmission. [10] Determines an upper bound for packets that have the ability of coding. Encoding number or number of coded packets in one transmission is the main parameter which is evaluated in this paper.

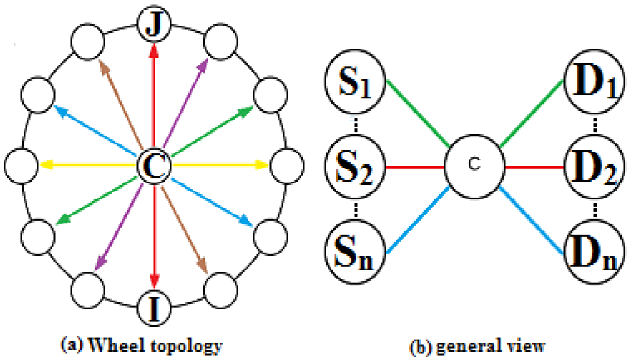


Fig. 3. (a) Wheel topology and (b) and general view

The considered upper bound of throughput (based on the scenario in Fig. 3.b) is calculated $n \times B/(n+1)$ where B is the value of shared channel bandwidth and n is the number of flows that can be coded together. In this paper, the number of coded packets is limited to $O((r/\delta)^2)$. This upper bound can be applied for all coding structure, that reliable transmission range is shown by r and interval between a reliable and unreliable transmission is shown by δ. In Fig. 3.a, upper bound of codable packets in one transmission is limited to $\Pi/arcos\ (\frac{r}{r+\delta})$. Network coding analysis in [11] and [13] is based on a wireless network model with erasure channel. Source nodes transmit packets to their receivers via the relay node. They ignore the seriousness of packet loss. In [14], Baochun et al. bring the theoretical benefits of network coding to practical systems. For example, peer-to-peer network application may be considered as the most promising scenario for network coding.

III. Background

In this section, we briefly describe background about maximum-weight clique and usage of vectors scalar product.

A. Maximum Weight Clique

A clique is a subset of a simple graph like G where there is an edge between every pairs of vertices. In the domain of graph theory, the problem of finding the biggest complete sub-graph is called maximum clique. Each vertex has a weight and the clique that has the total maximum weight is called "maximum-weight clique". In [5], the problem of finding the biggest coding pattern is mapped to finding maximum clique. In [12] is showed that finding maximum clique is a NP-Complete problem. In maximum weight clique problem, a positive value is added to every vertex but it does not lead to reduce complexity. So, the maximum weight clique is another NP-Complete problem [15].

B. Vectors Scalar Product

The dot product which is also called inner product, is denoted with the symbol$(\bullet)$. If all the elements of u and v vectors are only zero and one, then

$$0 \le \vec{u} \bullet \vec{v} \le \min(|\vec{v}|, |\vec{u}|) \le n \quad (4)$$

If we had u vector, we can maximize $\vec{u} \bullet \vec{v}$ by defining v as (5).

$$v(i) = \begin{cases} 1 & , \ u(i) = 1 \\ 0 \text{ or } 1, & u(i) = 0 \end{cases} \quad (5)$$

We use this concept for prioritizing and weighting the flows in the process of selecting the coding pattern.

IV. Proposed Algorithm

In this section, we describe briefly the system and how to map problem of finding the suitable coding pattern to maximum-weight clique and we propose our algorithm to find the suitable coding pattern for input flows.

A. System Overview

We consider network coding over wireless mesh networks where intermediate nodes (wireless mesh routers) are able to forward packets to other intermediate nodes and clients. In this paper, we propose an algorithm that can be used at the intermediate nodes to increase coding opportunities and throughput. Nodes can use either multi-hop or two hop coding structure to check coding condition of each pair of flows. This idea has been recently applied in 802.11 based on multi-hop wireless networks while DSR is used as a routing protocol.

B. Map the Problem to Find Maximum Weight Clique

In order to find the suitable coding pattern in each node, we modeled the problem in the following. Node a keeps the list of all flows that passes from it in the set *fls*. Moreover, each node keeps a matrix which is called *CA*. The number of rows and columns in this matrix is equal to the number of flows that have passed the node. In fact, there is a row and column for each passing flow from this node. The ith row of matrix *CA* is shown by $CA(i)$ and it indicates the flows that can be coded with flow i. $CA(i,j)=1$ if flow i can be coded with flow j or $i=j$. The problem of finding the coding pattern for new flows is mapped to the problem of finding the biggest subset which is called *results.* We define *results* as the biggest subsets from $CA(i)$ which all the members can be coded together. In this problem, if we consider flows as vertices, there is an edge between two vertices if they can be coded together. $CA(i) \bullet CA(j)$ shows that how much ith and jth flows are similar to each other in regard of coding capability. We assign a weight to each vertex based on the weighting function that is described in section C.

C. Algorithms Details

The following steps will be performed in order to find a suitable coding pattern when a new flow is received.

- The new flow's coding condition is checked with current flows. The row and column of the new flow are set in Matrix *CA* by zero and one values.
- A weight is assigned to each flow that is able to en-code with the new flow. This weight is obtained through inner product of two flows in Matrix *CA*. Formally,

$$w(i) = \begin{cases} CA[i] \bullet CA[n] & \text{if } CA[i,n] = 1 \\ 0 & else \end{cases} \quad (6)$$

- A Coding graph similar to the graph that proposed in [5] will be established but it contains the only verti-ces that can be coded with the new flow.

- In final step, by using mentioned algorithm in [16], we find the maximum-weight clique in the coding graph and vertices in this sub graph is considered as the coding pattern of new flow.

For detailed information about these steps a psudocode have been provided in Fig. 4.

V. Computational Complexity

The proposed algorithm can be applied in multi-hop coding and two-hop coding structures. Firstly, the total computational complexity of finding the suitable pattern in the proposed method is determined and then computational complexity of multi-hop or two-hop structures is specified.

```
Finding Pattern Procedure
// Codingarray[i][j]   coding condition between flow i and j
// fls[x] is list of flows  that traverse form node
Pick  new flow  F_new
for i=0 to N-1 do   //F_new is Nth flow that will traverse form node
      if F_new can encode to  fls[i]  then
            Codingarray[N][i]=1
            Codingarray[i][N]=1
      else
            Codingarray[N][i]=0
            Codingarray[i][N]=0
      end if
end for
Codingarray[N][N]=1
Weight _Of_ Vertex  [N]  //set of flows that can encode  with each other and  F_new
for j=0 to N-1 do
      if Codingarray[N][i] !=0 then
            Wight _Of_ Vertex[i]= Codingarray[N]•Codingarray[i]
      else
            Wight _Of_ Vertex[i]= 0
      end if
      Create coding graph(Codingarray[n],Wight _Of_ Vertex)
      find maximum weight clique()
end for
```

Fig. 4. Calculation of coding pattern in second method

A. Algorithm Computational Complexity Analysis

Computational complexity of our algorithm is as follows:

- Updating Nth row and column of matrix CA has order of $O(N \times \textit{check coding condition})$.
- Scalar product of two row of matrix CA has order of $O(N)$ and computational complexity of assigning weights to all flows that can be coded is $O(N^2)$.
- Finding the biggest maximum-weight clique has order of $O(N^2)$.
- So computational complexity of algorithm can be calculated as $O(N^2 + N \times \textit{check coding condition} + N^2)$

As it mentioned, we can apply the proposed algorithm in two-hop and multi-hop structures. So, computational complexity of each algorithm is related to the coding structure. In this section we study the computational complexity of multi-hop and two-hop coding structures.

B. Computational Complexity of Checking the Coding Condition in Two-hop Structure

In defining the coding structure in COPE method, two conditions are expected for checking the coding condition of two flows, $flow_1$ and $flow_2$:

1. The probability of hearing packets of $flow_2$ in the next hop of $flow_1$ is more than 0.8.
2. The probability of hearing packets of $flow_1$ in the next hop of $flow_2$ is more than 0.8. Obviously, computational complexity of checking the coding condition of two flows in COPE coding structure equals to $O(1)$.

C. Computational Complexity of Checking Coding Condition in Multi-hop Structure

Each node that is in the route of a flow, for coding procedure must keep the ID of other nodes in the route and their neighbors. In order to check the coding condition of a flow, the maximum number of required ID can be obtained by

$$(TTL-1)\times Max\_Deg + TTL \tag{7}$$

where Max_Deg is the maximum degree of every nodes in network and TTL is the upper bound of the route length that a flow can traverse. $(TTL-1)\times Max\_Deg$ is the number of nodes that listen packets of a flow. We use Fig. 5 to show a way that we can calculate computational complexity of checking two flows condition. f_1 and f_2 cross each other in a node like M. Distance between node M and source of f_1 is k while distance between node M and destination of f_1 is $TTL-k$. Also, distances between node M and source and destination of f_2 are k' and $TTL-k'$ steps respectively where

$$1 \le k, k' \le TTL-1 \tag{8}$$

we define set $NUS(M, flow)$ as

$$NUS(M, flow) = \{x \mid \forall y \in U(M, flow), x \in N(y)\} \tag{9}$$

For meeting multi-hop coding condition in (2.a), set $Condition_1(f_1, f_2, M)$ must not be empty. i.e.

$$Condition_1(f_1, f_2, M) = \{x \mid x \in NUS(M, f_2), \\ x \in D(M, f_1)\} \ne \phi \tag{10}$$

Also, to meet multi-hop coding condition (2.b), set $Condition_2(f_1, f_2, M)$ must not be empty. i.e.,

$$Condition_2(f_1, f_2, M) = \{x \mid x \in D(M, f_1), \\ x \in U(M, f_2)\} \ne \phi \tag{11}$$

To calculate $Condition_1(f_1, f_2, M)$, we sort two sets $NUS(M, f_2)$ and $D(M, f_1)$ by using merge sort with order of $O(n\log(n))$. In addition, we use dynamic programming to find first common element in two sorted sets S_1 and S_2 with order of $O(|S_1|+|S_2|)$. Then by using (12), the computational complexity of $Condition_1(f_1, f_2, M)$ is calculated by (13).

$$|NUS(M, f_2)| \le k' \times Max\_Deg \tag{12}$$

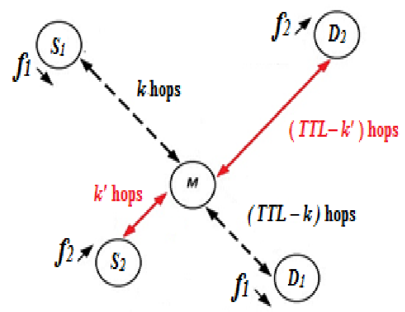


Fig. 5. Checking coding condition of two flows in multi-hop coding structure.

$$O((TTL-k)\times\log(TTL-k)+(Max\_Deg\times k')\times \log(Max\_Deg\times k')) \quad (13)$$

Parameters $Max\_Deg \times k' \times log(Max\_Deg \times k')$ and $(TTL - k) \times log(TTL - k)$ show sorting computational complexity of sets $D(M, f_1)$ and $NUS(M, f_2)$ respectively. To calculate set $Condition_2(f_1, f_2, M)$, first we sort two sets $U(M, f_2)$ and $D(M, f_1)$ then by using dynamic programming, we calculate first common element between these two sets. Computational complexity of checking coding condition for f_1 is calculated by (14). By performing the same tasks for f_2, computational complexity of coding condition for this flow is calculated by (15). So, computational complexity of checking coding condition equals (16).

$$O(Max\_Deg\times k'\times\log(Max\_Deg\times k'))+ O((TTL-k)\times\log(TTL-k)) \quad (14)$$

$$O(Max\_Deg\times k\times\log(Max\_Deg\times k))+ O((TTL-k')\times\log(TTL-k')) \quad (15)$$

$$O(Max\_Deg\times TTL\times\log(Max\_Deg\times TTL)) \quad (16)$$

VI. Simulation Results

We simulated our algorithm in OMNET++ simulator to evaluate the performance of our algorithm and compare it with random selection pattern in COPE[2] and Round robin mechanism in DCAR[5]. 802.11 is used as MAC layer. We first use wheel topology, as Fig.6, where a central node(0) is surrounded by fifteen nodes (1-15) that are distributed along the cycle. We randomly add twelve flows with the rate of 25 kbps in 180 seconds. we have only one flow in the start of simulation and one flow is added in every fifteen seconds. Sources and destinations of flows are placed in the circle's diameter and all flows are relayed by node(0). In this scenario, we used the coding structure in COPE to encode packets together. In the simulation, we evaluated the performance of both algorithms in terms of average network throughput, average end to end delay and average instruction counts.

We plot the average throughput, average end to end delay, executed instructions counts and the length of coding pattern respectively in Fig. 7a, 7b, 7c and 7d. As it is clear, in Fig. 7.a, our approach offers highest throughput. The underlying reason is that our approach finds coding pattern with greater length. So, it increase coding opportunities at the node(0). In Fig 7.b, our algorithm has the lowest end to end delay Since node(0) does not wait for receiving a suitable packet when the wireless channel is available and there are no coding opportunities. As a result, selecting a coding pattern with higher length leads to less end to end delay in the network. Although, as it is shown in Fig. 7.c, the number of instructions increases in order to find the longest coding patterns and reach more encoding opportunities. Finally, Fig. 7.d indicates that our algorithm has the longest coding pattern for each flow.

In second scenario, we compare our algorithm with random and round robin mechanisms. 100 nodes are distributed randomly in the network. We use the same flows as the first scenario. The average throughput and end to end delay for all flows is plotted in Fig. 8a and 8b. Our algorithm has the highest throughput and the lowest end to end delay due to create more coding opportunities by finding the longest coding patterns.

VII. Conclusion and Future Works

One of the most important concepts that have a direct relationship with network coding performance is the problem of selecting a suitable coding pattern. In this paper, this problem is mapped onto maximum-weight clique. This algorithm firstly makes a coding graph for input flows. Then it assigns a weight to each vertex. Finally by finding the maximum-weight clique on this graph, a suitable coding pattern is selected for a new flow.

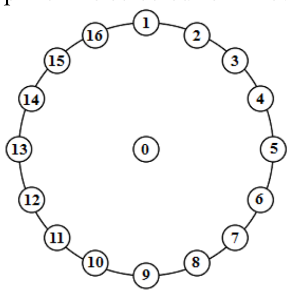
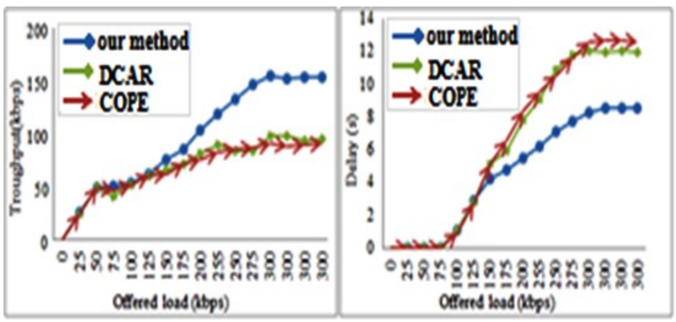
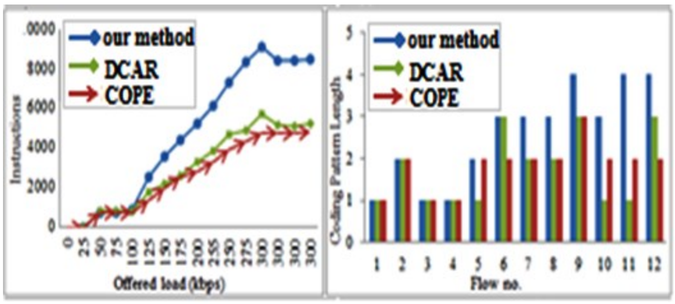


Fig. 6:wheel topology

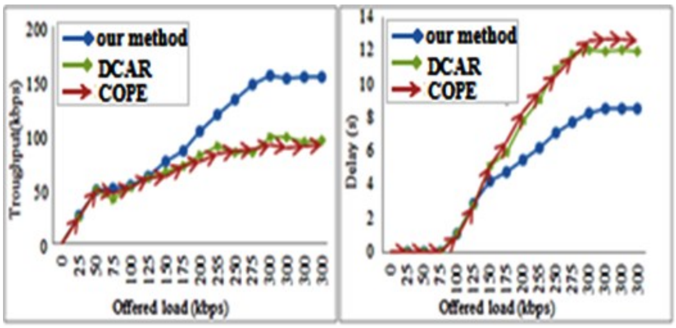


(a) (b)



(c) (d)

Fig. 7. Results from a wheel topology. (a) Average throughput. (b)Average end to end delay.(c) Average instruction count.(d) Pattern length of each flow.



(a) (b)

Fig. 8. Results from a random topology. (a) Average throughput. (b)Average end to end delay.

Simulation results show that our algorithm can increase the network throughput and decrease end to end delay as compared with existing mechanisms. In the future, we will extend our work and consider the importance of packets based on QoS requirements in the coding pattern selection.

References

[1] R. Ahlswede, N. Cai, S.-Y. Li, R. Yeung, "Network information flow," in IEEE Transaction on Information Theory. vol. 46, no. 4, pp. 1204–1216, 2000

[2] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard and J. Crowcroft, "XORs in the Air: Practical Wireless Network Coding," IEEE/ACM Transactions on Networking, pp. 497-510, 2008.

[3] F. Zhao, M. Medard, "On analyzing and improving COPE performance," in Proc. of ITA, San Diego, CA, Feb 2010.

[4] S. Omiwade, R. Zheng, C. Hua, "Practical localized network coding in wireless mesh networks," in Proc. 5th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON '08), June 2008.

[5] J .Le, JCS. Lui, D. M. Chiu, "DCAR: Distributed Coding-Aware Routing in Wireless Networks," IEEETRANSACTIONS ON MOBILE COMPUTING, vol. 9, no. 4, pp. 596-607, 2010.

[6] S. Sengupta, Sh. Rayanchu, S. Banerjee, "Network Coding-Aware Routing in Wireless Networks," IEEE/ACM Transaction on Networking, Vol. 18, No. 4, 2010.

[7] A. Khreishah, I. M. Khalil, J. Wu, "Distributed network coding-based opportunistic routing for multicast," in Proceedings of the thirteenth ACM MobiHoc, pp. 115-124, 2012.

[8] Z. Zhou, L. Zhou, "Network joint coding-aware routing for wireless ad hoc networks," in Proceedings of the IEEE international conference on wireless communications, networking and information security (WCNIS'10), pp. 17–21, June 2010.

[9] J. Liu, D. Goeckel, D. Towsley, "Bounds on the Gain of Network Coding and Broadcasting in Wireless Networks," Proceedings of IEEE INFOCOM'07 2007.

[10] J. Le, J. Lui, D. Chiu, "How Many Packets Can We Encode?—An Analysis of Practical Wireless Network Coding," Proc. IEEE INFOCOM, pp. 371-379, 2008.

[11] D. Traskov, M. M´edard, P. Sadeghi, R. Koetter, "Joint scheduling and instantaneously decodable network coding," in Proc. IEEE GLOBECOM, December 2009.

[12] M.R. Grey, D.S. Johnson, "Computers and Intractability: A Guide to the Theory NPcompleteness," Freeman, New York, 1979.

[13] T.Zhang, E. Chunlin, B. Wang, "Performance analysis on wireless unicast with network coding," in 14th International Conference on Advanced Communication Technology (ICACT 2012), February 2012.

[14] B. Li, D. Niu, "Random network coding in peer-to-peer networks: From theory to practice," in Proceedings of the IEEE, vol. 99, no. 3, pp. 513 –523, march 2011.

[15] D. Kumlander, "A new exact algorithm for the maximum-weight clique problem based on a heuristic vertex-coloring and a backtrack search," in Proceedings of The Forth International Conference on Engineering Computational Technology, pp. 137-138, Civil-Comp Press, 2004.

[16] P.R.J. Östergård, "A new algorithm for the maximum-weight clique problem," in Nordic Journal of Computing, Vol. 8. 424-436, 2001.