Contents lists available at GrowingScience

Management Science Letters

homepage: www.GrowingScience.com/msl

New jitter-aware network coding approach for packet selection in wireless mesh network using Jacobsen algorithm

Rasoul Nikoee Saravani[a*] and Seyed Javad Mirabedini[b]

[a]*Department of Computer Engineering, Tehran Science and Research Branch, Islamic Azad University, Damavand , Iran*
[b]*Department of Computer Engineering, Central Tehran Branch, Islamic Azad University, Tehran, Iran*

CHRONICLE

Article history:
Received June 10, 2014
Accepted 20 November 2014
Available online
November 23 2014

Coding model
Network coding
Quality of service
Source routing algorithm
Wireless networks

ABSTRACT

Network coding (NC) is an approach for increasing the throughput of communication networks. NC has been derived from the idea of combining packets and forwarding them to achieve higher throughputs. From the very beginning, NC has attracted the attentions of scholars since they could improve the throughput of wired networks. Previous works have focused on the improvement of network throughput. Some methods add artificial delay to send packets. The methods are used in order to gain coding time and to increase throughput but they could not be used in immediate applications. However, early packets dispatch or adding delay to packets, for increasing coding time, can influence on the jitter of sent data. On the other hand, jitter is an important parameter in determining multimedia broadcasting time. Unlike previous methods, this paper provides a jitter-based coding algorithm using Jacobsen method. In the proposed method, the selection priority is based on packets jitter so that a packet with the minimum jitter is considered as the first priority for coding. Related computations are performed through Jacobson algorithm. According to simulation results, selecting a proper coding pattern for jitter can improve jitter by 30% while the throughput of this method shows no considerable change compared with that of available methods.

1. Introduction

Network coding (NC), introduced by Ahlsewede et al. (2000), is an approach with for increasing throughput of communication networks. From the very beginning, NC attracted the attentions of all researches who were trying to improve the use of wired and wireless networks resources. Before NC approach, the role of intermediate nodes, like switches and routers, was only to forward packets to their destinations. NC theory encourages the intermediate nodes to forward the packets and to combine packets with different sources before forwarding them, which in turn increases network throughput. Fig. 1 illustrates the process of network coding in wireless networks using a simple example.

*Corresponding author.
E-mail addresses: ra.nikoee@Gmail.com (R. Nikoee Saravani)

doi: 10.5267/j.msl.2014.11.009

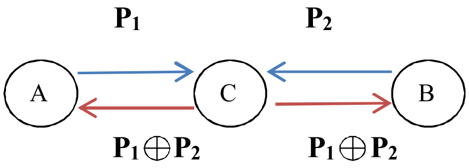


Fig. 1. An example of network coding

In this scenario, Node *A* tries to forward packet P_1 to node *B*. Node *B*, however, tries to forward packet P_2 to node *A*. The nodes are not in direct forwarding range to each other and they should use the relay node of *C*. Within the first time slot, node *A* forwards packet P_1 to node *C* and in the second time slot, node *B* forwards packet P_2 to node *C*. In the 3rd time slot, node *C* can combine P_1 and P_2 packets and forward packet $P_1 \oplus P_2$ to nodes *A* and *B*. Having packet P_1, node *A* can decode the considered packet from $P_1 \oplus P_2$. Similarly, having packet P_2, node *B* can decode the considered packet form $P_1 \oplus P_2$. COPE was the first network coding architecture in wireless networks that supported the idea of coding unicast traffics (with different originations) before forwarding them. COPE was a bridge between theoretical and practical opinions.

The brilliant results of COPE attracted the attentions of many scholars to network coding idea so that a number of them tried to improve COPE performance. In a method developed by Zhao and Médard (2010), after the random selection of every virtual queue, the initial packer does not select it for coding with the sent packet, as they believe that this loses the time for coding the initial packets of the virtual queue. In other method developed by Omiwade et al. (2008), the receivers can save temporarily the coded packets in order to capable of decoding the considered packets in future. Moreover, the other method developed by Dong et al. (2007), the coding process of COPE is used.

Previous works have mainly focused on throughput improvement (Katti et al., 2006). Available methods, e.g. Huang et al. (2008), added artificial delays to packets in order to gain time for coding and to increase throughput while the methods cannot be adopted in many immediate applications. However, early sending of packets or adding delays to them, aimed at gaining more time for coding, can influence on the jitter of sent data. On the other hand, jitter is an important parameter in determining multimedia broadcast time. To solve the problem, solutions such as Chen et al. (2012) make a tradeoff between throughput and jitter. In other method developed by Zhang and Qian (2011), delay is added to packets in order to gain time for coding the packets but this method tries to adopt PNCP model in order to control the applied delay.

Ahlswede et al. (2000) introduced DNPGD coding model for delay decrease purposes. In this model, the total delay applied to a forwarding packet is used in coding process. This method works as follows. First, the expected delay for each flow packet is calculated and then, the flow delay of every packet is calculated. Then, the calculated delays are compared with each other in order to decide that whether more delay can be added to the packet in order to create more coding time.

In this paper, we concentrate on introducing a coding method for improving jitter parameter. Unlike previous methods with the focus on throughput, in the proposed method, the selection priority of packet is based on jitter so that a packet with the minimum jitter is the first priority for coding. Related calculations are carried out by Jacobsen algorithm. Fig. 2 illustrates the influence of jitter average-based coding algorithm selection. In this figure, it is assumed that nodes *A, B* and *C* can forward packets P_1, P_2 and P_3 to *D, E* and *F*, respectively via relay node *R*. Node *R* can code packet P_1 with two coding models:

1- $P_1 \oplus P_2$
2- $P_1 \oplus P_3$

To select coding model, node R calculates the current delay of both packets. Then, based on the calculated delay, it calculates the current jitter average of both flows to which packets P_2 and P_3 belong. The packet with the minimum effect on average jitter will be considered as the preferred packet for coding and node R will use that coding model. Before discussing about the proposed algorithm, we study the process of calculating jitter.

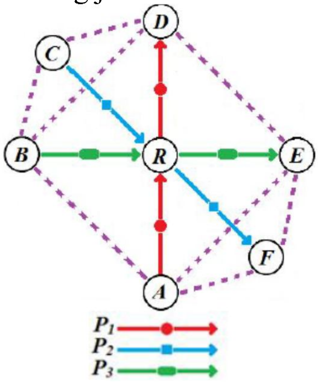


Fig. 2. An example for jitter-based model selection

The structure of this paper is as follows. Section 2 discusses the primary concepts of network coding (NC), quality of service (QoS) and reviews previous works on the selection of coding model and QoC-aware coding. In section 3, the proposed algorithm is explained in detail. In addition, the way of calculating distributed end to end delay in every node is discusses and then jitter (variation of delay) is calculated. Section 4 evaluates proposed methods and discusses results. Section 5 concludes the paper and presents suggestions for further studies on this field.

2. Related Works

Today, Internet offers many services to its clients but suffers from great disadvantages. Despite the fact that network routers can receive and process input packets, they cannot assure the safe delivery of the packets to destinations. Regarding the ever-increasing use of Internet service especially its importance as a tool of developing global trade, many attempts are being accomplished to maintain QoS in Internet. IETF introduces different models and mechanisms in order to assure the level of QoS requested by clients. You may find some of the most important models such as RSVP (resource reservation protocol), differentiated service, MPLS, traffic engineering and constraint-based routing. There are a number of parameters including throughput, delay, jitter and loss used to determine QoS in network. During queuing and buffer delay, the sequenced packets of the same traffic flow experience different delays, which create jitter. In addition, jitter is created for another reason: Different packets of a traffic flow may be sent to destination after routing via different physical routes. This creates jitter in the delivered packets to destinations.

2.1 Network Coding (NC)

Wireless networks have throughput and scalability limitations as they are susceptible to noise and have a distributed nature. Coding idea, proposed by Ahlswede et al. (2000) shows how NC can promote the throughput of networks, especially wireless networks. Katty et al. (2008) introduced the first practical coding system for multi-step wireless networks. This idea combines unicast flows packet to increase throughput. COPE idea consists of three fundamental phases: opportunistic hearing, opportunistic coding and learning neighborhood status. This method uses a two-stepped structure for coding and decoding purposes where the coded packets should be decoded one step beyond the coder node. The following figure illustrates model selection in COPE method. First, the packets of each flow within coder node are placed in virtual queues based on their next step. In this model, every queue has only one packet. During the process of selecting coding pattern, packets are picked up from the start point of every queue and their coding condition with the encoded set is evaluated. Every node uses a random permutation for selecting queue and evaluating the coding condition of related packets. This process is repeated for every forwarding step. COPE coding assumes the followings: In order to code N packets with each other the following conditions should be met:

P_i is the extent of probability to which the next step node may hear packet i.

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

The probability of decoding the main packet in the next step node is equal to the probability of hearing previous *n-1* packets coded with the considered packet. If after the *Xor* of the nth packet the PD value for all *n* nodes of the next step is greater than a threshold value (G>0.8), the necessary condition for the Xor of the nth packet will be met. The brilliant results of COPE have attracted the attentions of scholars to NC so that a group of them tried to improve COPE performance (Dong et al., 2007). According to Dong et al. (2007), following the random selection of every virtual queue, its first packet is not selected for coding forwarding packet as they believe that this loses the time required for coding the first packets of virtual queues. For the methods developed by Zhao and Médard (2010) and Omiwade et al. (2008), receivers can temporarily save the coded packets in order to become able to decode a considered packet in future and the coding process of COPE is used.

Previous works have focused mainly on throughput improvement (Katti et al., 2008). Some methods like Huang et al. (2008) add artificial delays to packets to gain time for coding and to increase throughput while the methods cannot be used in many immediate applications such as the applications needed to support QoS parameters including delay. To solve the problem, a number of solutions make a tradeoff between throughput and jitter (e.g. Ao et al., 2012). Dong et al. (2013) adds delays to packets in order to increase coding time but it tries to control the added delay by introducing PNCP model.

This paper discusses two main problems of network coding:
1- Which time coding should be used?
2-Which coding structure should be used?

The authors of this paper believe that coding is not always a proper solution. This method addresses the first problem based on two parameters: the status of queue and the preference of forwarding packet. The second problem is covered by COPE coding where relay node makes a trade-off between throughput and jitter and decides that whether or not to add a delay. This paper makes coding decisions based on buffer condition and the class of forwarding packet class. Yeow et al. (2009) introduced a coding model titled as DNPDG in order to reduce delay. This model uses the total sum of the current delay added to the sent packet in the coding process. In fact, this method determines the expected delay for every flow packets and calculates current delay of every packet. Then, it compares results and decides that whether to add additional delays to the sent packet in order to gain more time for coding. Yeo et al. (2009) focused on minimizing delay for broadcast flows by using network coding. Eryilmaz et al. (2006) studied the gain in delay performance resulting from network coding. Ying et al. (2009) demonstrated that coding achieves the optimal delay-throughput tradeoff in mobile ad-hoc networks. These works only consider the performance of average delay and do not address strict per-packet delay bounds.

3. The Proposed Method

This section discusses the proposed model in detail. Unlike available methods, this algorithm uses network coding to reduce jitter. Before discussing coding selection, the procedure of calculating delay and jitter is reviewed.

3.1 Calculation of Delay

The following symbols are used in the process of calculating a delay:

A: stands for a node available in network
P_i: stands for the ith received packet in an intermediate node like *a*
Ta (Pi): stands for the time at which P_i is delivered to the intermediate node
Current-Time: stands for the current time of a node like *a*
Db(Pi): stands for the delay added to the p_i packet in the intermediate node buffer

$$D_b(P_i) = Current_{Time} - T_a(P_i) \tag{2}$$

TX(Pi): stands for forward delay
R: stands for forward rate
L: stands for the volume of forwarding packet

$$T_x(P_i) = \frac{L}{R} \tag{3}$$

Tp(Pi): stands for broadcast delay
D: stands for distance between receiver and transmitter
V: stands for light velocity

$$T_P(P_i) = \frac{D}{V} \tag{4}$$

Time-Stamp(pi): stands for the accumulated delay of packet P_i to the packet delay in each step. This parameter is calculated by the following relation:

$$\text{Time\_Stamp}(P_i) = \text{Time\_Stamp}(P_i) + D_b(P_i) + T_x(P_i) + T_P(P_i) \tag{5}$$

3.2 Calculation of Jitter

Jitter is used in different applications and it is calculated in various ways. Jacobson's method is a method for calculating the jitter (Eryilmaz et al., 2006; Qiong & Mills, 2001). It has the following three states:

1- SRTT (delay) : send-receive total time of a packet
2- RTTVar (jitter) send-receive total time average in an interval
3- RTO response time which is used to forward the packet for the second time in the event of receiving no ack

In Jacobson's method, SRTT is derived from Eq. (5). In the following equation, RTT is the SRTT derived from the nth packet and a=1/8. Eq. (6) is used to determine the jitter of SRTT. Jacobson's method uses only forward and broadcast delays to calculate the jitter. In other words, it does not take queue and processing delays into account.

$$SRTT_{(new)} = a \times SRTT_{(old)} + (1 - a) \times RTT \tag{6}$$
$$RTTvar_{(new)} = (1 - b) \times RTTvar_{(old)} + b \times |SRRT - RTT| \tag{7}$$

In Eq. (7), a=1/4.

Jitter is calculated by Eq. (8). RTTvar$_{(\text{new})}$ shows the jitter of a packet before forwarding to the next step node. This equation has been derived from Jacobson’s method and it differs with the Jacobson’s method in that the proposed method only needs half of RTT.

$$RTTvar_{(new)} = b \times RTTvar_{(old)} + \frac{(1-b)|\, RTT - SRTT|}{2} \tag{8}$$

In Eq. (7), b=1/8. $\overline{\overline{RTTvar}}_{(\text{new})}$ shows the effect of a new packet delay on jitter. This is derived from Eq. (9). This parameter shows the effect of the nth packer on jitter at forwarding time (whether it is reached to queue start point itself or is coded by the packet at the start point of the queue).

$$\overline{\overline{RTTvar}}_{(new)} = \frac{\left|RTTvar_{(new)} - RTTvar_{(old)}\right|}{RTTvar_{(old)}} \tag{9}$$

3.3 Coding Structure

Similar to COPE method, the coder nodes of this algorithm place the packets of every flow in virtual queues with respect to the next step. Upon the entrance of a new flow, the coder node determines the virtual queue in which the packets of this flow should be placed. To determine coding pattern, the node works in this manner:

1- Coder node should seek for the packets belonging to the flows of *temp* array. Each virtual queue should be assessed until the first packet, which can be coded with the forward packet, is found.
2- Calculate $\overline{\overline{RTTvar}}_{(new)}$ for the candidate packet
3- Repeat steps 2 and 3 for other virtual queues
4- Select the packet with the minimum $\overline{\overline{RTTvar}}_{(new)}$.
5- Go to step 2 and repeat all steps until there is no packet to add to the forwarding set in the queue

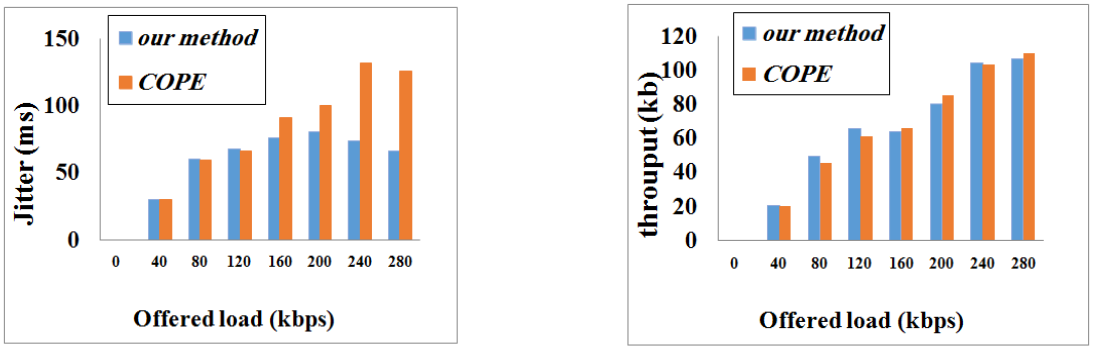
Fig. 3 shows the steps of this algorithm.

```
1.   Pick packet $p$ at the head of the output queue.
2.   Natives = {$p$}
3.   nexthops = {$nexthops$ ($p$)}
4.   candidate_packet=null
5.   $\overline{\overline{RTTvar}}$_candidate_packet=0
6.   for Neighbor $i$ = 1 to $M$ do
7.       Pick packet $pi$, the head of virtual queue $Q(i$, which queue)
8.       if $\forall n \in$ nexthops $\cup\{i\}$, Pr[$n$ can decode $p \oplus pi$] then
9.           if $\overline{\overline{RTTvar}}$_candidate_packet < Calculate_delay(pi) then
10.                  $\overline{\overline{RTTvar}}$_candidate_packet=  Calculate_$\overline{\overline{RTTvar}}$(pi)
11.                candidate_packet=pi
12.          end if
13.      end if
14.          $p = p \oplus$  candidate_packet
15.          Natives = Natives ∪{ candidate_packet }
16.          nexthops = nexhops ∪{$i$}
17.      temp=temp∩Ca(flow-id of  candidate_packet)
18.  end for
19.  if temp!=0   then     at least  one entry of temp isn’t 0
20.      if  candidate_packet !=null  then atleast one packet is exsit to code with forwarding packet
21.          Go to line 5
22.      end if
23.  end if
24.  return $p$
```

Fig. 3. Coding structure of proposed algorithm

4 Simulation

We have simulated the proposed algorithm in TOSSIM simulator to evaluate the performance of the algorithm and compared it with random selection pattern in COPE (Katti et al., 2008). [802,11] has been used as MAC layer. We considered network coding over wireless mesh networks where intermediate nodes (wireless mesh routers) were able to forward packets to other intermediate nodes and clients. Random topology with 100 nodes was used for simulation scenario. In this scenario, 12 flows with a load of 20 kbps were created, randomly. In the beginning of simulation process, there was only one flow. Then, for every 15 seconds, a random flow load was added to the set. This scenario used COPE coding structure to code packets. The performance of this algorithm was evaluated considering throughput and jitter parameters. Fig. 4 shows the simulation results of the first scenario. Fig. 4-1 shows throughput in the first scenario. According to this figure, our proposed method applies fewer changes to packets as it considers packet jitter as a priority. Fig. 4.2 compares the throughputs of our proposed method and COPE. It is evident that there was no significant difference in throughput between both methods. The reason is that packets were selected in random in COPE method.



(1) (2)

Fig. 4. simulation results of the proposed method: (1) jitter, (2) throughput

5 Conclusion

NC was introduced as a solution for improving the performance of wired networks. Till now, different solutions have been presented to improve NC performance. Previous works have focused on throughput improvement. On the other hand, many attempts have been accomplished to maintain QoS in Internet. However, the assurance of QoS including jitter is an essential component in multimedia applications. Unlike previous works, this paper used NC to control jitter. According to simulation results of the proposed method, this algorithm used fewer changes to packets compared with the initial proposed methods. According to simulation results, selecting a proper coding pattern for jitter can improve jitter by 30% while the throughput of this method shows no considerable change compared with that of available methods. The possibility of taking advantage of this method makes it suitable for adjusting the playback of video streams in the network. Future work could include combination of two parameters of delay and delay variation as noted before. In addition, the selection of coding pattern may be based on criteria such as interference control, load balancing and reduction of the number of lost packets (Packet Loss).

Acknowledgement

The authors would like to thank the anonymous referees for constructive comments on earlier version of this paper.

References

Ahlswede, R., Cai, N., Li, S. Y., & Yeung, R. W. (2000). Network information flow. *Information Theory, IEEE Transactions on*, *46*(4), 1204-1216.

Ao, W. C., Chen, P. Y., & Chen, K. C. (2012). Rate–Reliability–Delay Tradeoff of Multipath Transmission Using Network Coding. *Vehicular Technology, IEEE Transactions on*, *61*(5), 2336-2342.

Chen, P. Y., Ao, W. C., & Chen, K. C. (2012). Rate-Delay Enhanced Multipath Transmission Scheme via Network Coding in Multihop Networks. *IEEE Communications Letters*, *16*(3), 281-283.

Dong, Q., Wu, J., Hu, W., & Crowcroft, J. (2007, September). Practical network coding in wireless networks. In *Proceedings of the 13th annual ACM international conference on Mobile computing and networking* (pp. 306-309). ACM.

Dong, Z., Shen, S., & Li, Y. (2013). Dynamic Network Coding with Delay-guarantee in Multi-hop Wireless Networks. *Journal of Computational Information Systems*, *9*(4), 1529-1538.

Eryilmaz, A., Ozdaglar, A., & Medard, M. (2006, March). On delay performance gains from network coding. In *Proc. CISS* (Vol. 6, pp. 864-870).

Huang, Y., Ghaderi, M., Towsley, D., & Gong, W. (2008, June). TCP performance in coded wireless mesh networks. In *Sensor, Mesh and Ad Hoc Communications and Networks, 2008. SECON'08. 5th Annual IEEE Communications Society Conference on* (pp. 179-187). IEEE.

Katti, S., Rahul, H., Hu, W., Katabi, D., Médard, M., & Crowcroft, J. (2006, September). XORs in the air: practical wireless network coding. *ACM SIGCOMM Computer Communication Review,* 36(4), 243-254.

Omiwade, S., Zheng, R., & Hua, C. (2008, June). Practical localized network coding in wireless mesh networks. In *Sensor, Mesh and Ad Hoc Communications and Networks, 2008. SECON'08. 5th Annual IEEE Communications Society Conference on* (pp. 332-340). IEEE.

Li, Q., & Mills, D. L. (2001). Jitter-based delay-boundary prediction of wide-area networks. *Networking, IEEE/ACM Transactions on*, *9*(5), 578-590.

Wei, X., Zhao, L., Xi, J., & Wang, Q. (2009, September). Network coding aware routing protocol for lossy wireless networks. In *Wireless Communications, Networking and Mobile Computing, 2009. WiCom'09. 5th International Conference on* (pp. 1-4). IEEE.

Yeow, W. L., Hoang, A. T., & Tham, C. K. (2009, April). Minimizing delay for multicast-streaming in wireless networks with network coding. In *INFOCOM 2009, IEEE* (pp. 190-198). IEEE.

Ying, L., Yang, S., & Srikant, R. (2007, April). Coding achieves the optimal delay-throughput trade-off in mobile ad-hoc networks: Two-dimensional iid mobility model with fast mobiles. In *Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks and Workshops, 2007. WiOpt 2007. 5th International Symposium on* (pp. 1-10). IEEE.

Zhao, F., & Médard, M. (2010, January). On analyzing and improving COPE performance. In *Information Theory and Applications Workshop (ITA), 2010* (pp. 1-6). IEEE.

Zhang, R., & Qian, Q. (2011, January). Probabilistic network coding with priority over wireless networks. In *Measuring Technology and Mechatronics Automation (ICMTMA), 2011 Third International Conference on* (Vol. 2, pp. 918-921). IEEE.