TELKOMNIKA, Vol. 11, No. 2, February 2013, pp. 774~782
ISSN: 2302-4046

The Application Research of Randomized Network Coding in ForCES Router

Ming Gao*[1], ShiJu Li[1], Bangzhi Xiao[2]
[1]Department of Information Science & Electronic Engineering, ZheJiang University
[2]College of Information & Electronic Engineering , Zhejiang GongShang University, Hangzhou, P.R.China
Corresponding author, e-mail E-mail: gaoming@mail.zjgsu.edu.cn*, leesj@isee.zju.edu.cn ,
xiaobangzhi@pop.zjgsu.edu.cn

Abstract

ForCES router(ForTER) is an open experimental platform for network research of new routing algorithm，packet processing which is based on ForCES. By introducing and designing six new LFBs to ForTER, this paper propose a method of implementation of multicast IP router with network coding enabled. These LFBs undertake coding functions such as packets coding, labeling, buffering and decoding. We build an IP multicast network with ForTERs, which including the classic shortest path tree constructed by the reverse path from the source node to each sink, then evaluate the completeness of throughput, performance overhead and CPU utilization.

Keywords: *network coding, ForTER, ForCES, LFB*

1. Introduction

The network coding brings about a new idea that the middle network nodes can supply packet-content processing function against traditional transmitting mode. By compressing information, overthrows the well-known viewpoint that a single bit cannot be compressed in network nodes, so that the network performance is greatly improved [1]. However the harsh reality is that most network coding researches can only be completed with simulation for the lack of an appropriate platform to integrate the current achievement of network coding research with real network devices (such as routers). Under the circumstances, this paper develops an open experimental platform called ForTER.

By introducing some new ForCES (Forwarding and Control Element Separation) LFBs (Logical Function Block) into the ForCES Model [2], the ForTER implements a network coding enabled ForCES IP multicast router. As the evidence, [3] shows that ForCES is an open architecture of network device and its character of modularity, open and programmability can adapt to new network service via flexible programming and reconfiguration in the CE (Control Element).

Since the job of setting standards of ForCES is finished by IETF, many organizations accelerate their steps to develop the prototypes of ForCES. Under the ForCES framework [4] use these standards to specify IntServ's processing [5] proposes routing protocol modules derived from ForCES FE (Forwarding Element). Further [6] designs a distributed router to customize the functions on line by dynamically adding and deleting soft-module. Then M. Hidell [7] defines the ForZ protocol, based on ForCES architecture they use netlink as the communication mechanism between CE and FE to implement a router. Recently, the ITL lab of ZJGSU [8] released their ForCES Middle Software, which has immensely reduced the difficulty of realization of ForCES devices.

This paper focuses on the feasibility study of ForCES combined with network coding. To achieve the goal above, we utilize some current research results to our packets coding, labeling and decoding etc. About the problem of packet coding and decoding, firstly Li and Yeung et al [9] proved using linear network coding can maximize the multicast capacity in 2003. They believed that the information function of the node can be a simple linear combination in Galois field, such as $f(y_1, y_2, y_3) = \gamma_1 y_1 + \gamma_2 y_2 + \gamma_3 y_3$. Similarly, the decoding of the sink node can also use the linear calculation. Based on Li and Yeung's work, Sanders and Jaggi [10][[11]

Received August 21, 2012; Revised December 29, 2012; Accepted January 13, 2013

continued to prove that, in the acyclic network, looking for the coding and decoding coefficient can be a polynomial time algorithm, and Erez and Fedor [12] are committed to find the coefficients in cyclic networks. To meet the prerequisite of a large enough domain, Ho et al. [13][14] proposed the linear random network coding, this method is based on the strategy of a linear random network coding vector, the intermediate node randomly selected a combination map to links, these mapping relationships are mutually independent, and the benefits of doing this is to make the transmission matrix be full rank in a higher probability and ensure the decoding of the sink nodes to be successful [15]. Then Jaggi et al [11] further showed clearly that decoding success rate is 0.996 when $| F | = 2^{16}$ or $| E | = 2^{8}$, Chou et al [16] pointed out that $| F | = 2^{8}$ is enough to practical applications. In practical applications, the Microsoft P2P file sharing system avalanche is using linear random network coding strategy.

2. Model Formulation

As we all know that the biggest characteristic of network coding against traditional router is that the former have the capability of packet processing. In order to make ForTER to support network coding function, we introduce six classes of network coding relevant LFB into ForTER. Of course the premise is that ForTER must have the ability to generate multicast network topology, for this point, ForTER should be designed to support PIM-SM multicast protocol [17].

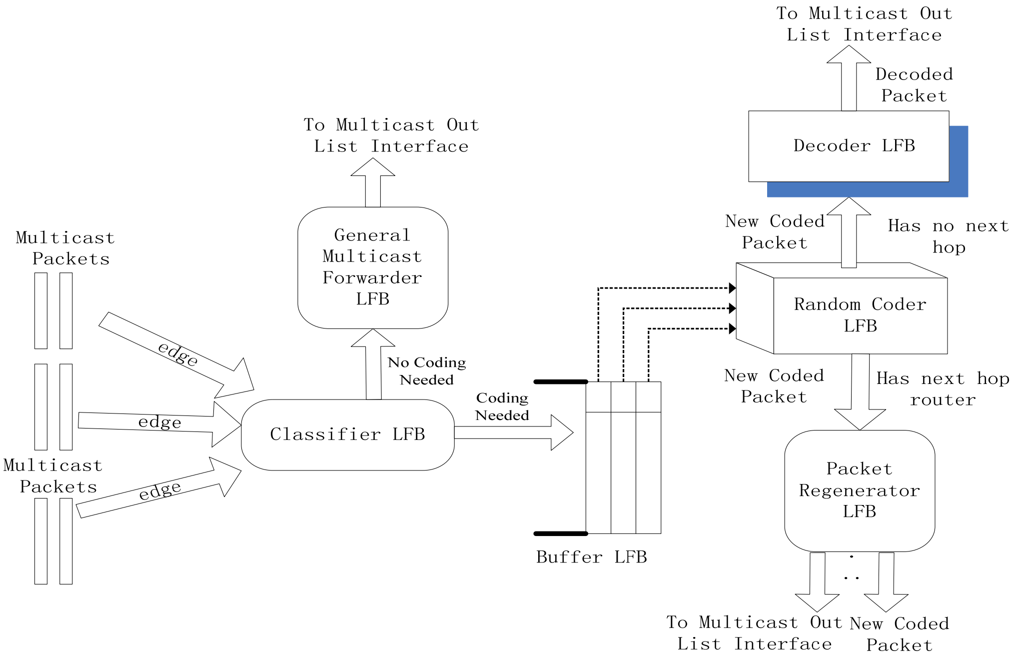

Figure1. The Working Path of Six LFBs

The six classes of network coding relevant LFB mentioned above, they are Classifier LFB, General Multicast Forwarder LFB, Buffer LFB, Random Coder LFB, Decoder LFB and Packet Regenerator LFB. The working path of the six LFBs is shown in

Figure1. An incoming multicast packet will be identified as coding or uncoding by Classifier LFB according to the field of protocol in IP header. For the coding packet, Classifier LFB will label it (see Chapter 3.2) and inform CE for further processing, otherwise the packet just to be stored-and-forwarded by General Multicast Forward LFB. For the case of the former, the informed CE initiates Buffer LFB and queues it.

In the environment of conventional network, packets are transmitted orderly by sequence. However, packet loss, congestion, traffic competition and cycle networks are existing anytime and anywhere, the numbers of packets per unit time on link are different greatly. Buffer

LFB tries to make the coding packet to be independent of other packets with the mechanism of queuing. Every once in a while at set intervals, the Buffer LFB will schedule and deliver the coding packet to Random Coder LFB for further coding processing (see Chapter 3.3).

Random Coder LFB is designed for coding operation of packet, the coefficients of coding are uniformly, independently and randomly selected from a Galois field F, which can be seen in charpter 3.1. According to the number of incoming coding packet, Random Coder LFB supplies two coding modes: for single coding packet, simple multiplication is taken, for multiple coding packets; linear combination is taken to generate a new coded packet.

Coding operation will reduce the number of packets in intermediate nodes partly, so we need to regenerate the coding packets and send the duplication to the respective destination. In fact, General Multicast Forwarder LFB has the ability as well, it can be used to replace Packet Regenerator LFB. After the encoded packets have been received by General Multicast Forwarder LFB, it will look up the multicast out list interface according to metadata that produced by upstream LFB, then based on the number of output port copy the packets and encapsulate the 2-layer header (for the format see Figure 3) and then send them out.

Decoder LFB utilizes Gaussian elimination method to decode the code vector and get the original data information. Before it started, the system needs to judge whether the node is the last-hop node in advance. If to be so, execute the decoding operation. Because encoded packet carries the global coding vector, each will generate an additional overhead of h symbols, if $h = 40$, $| F | = 2^8$, the overhead of carrying the coding vector is 40/1400 = 2.86 %(see chapter 3.2). The purpose is that the global code vector is carried by the encoded data packet itself, it also means that the sink node does not have to know the coding function and the network topology. As long as the number of received packet is not less than h, and the global code vectors are linearly independent [18], the sink node will be decoded accurately (see Chapter 3.4).

3. Design of Coding Scheme

This chapter will illustrate how the six LFBs mentioned above work properly with coding nature. Because of the advantages of the distributed feature of random network coding algorithm, we choose random network coding to be ForTER's coding construction algorithm and refine the decomposition of random network coding function. Random network coding is a distributed coded system; every packet will get a random coded coefficient after passing through a node and have a coding operation [19]. It means that even only one packet passes through an intermediate node, coding operation is needed. However from the viewpoint of network coding, a single packet does not need the coding operation but simply to be copied or forwarded. So in ForTER, it will have coding operation only if there are two or more. On the other hand, when there is only one multicast packet goes through intermediate node, what ForTER needs to do is just copy and forward the packet.

The core idea of the network coding is the processing procedure of packets that entering the intermediate node. ForTER as an intermediate node in the network, it's coding-functions such as copying, forwarding and linear processing etc. need to be designed carefully. These functions will embeded in the six LFBs which introduced in chapter 2, mainly Classifier LFB for labeling, General Multicast Forwarder LFB for forwarding, Buffer LFB for buffering, Random Coder LFB for coding, Decoder LFB for decoding, Packet Regenerator LFB for copying. Here we only state four functions of coding, labeling, buffering and decoding, because other functions of copying and forwarding are the same as tradition.

3.1. Coding Packets

Using ForTER, we can build a multicast network G = (V, E, C), here V specially refer to the set of ForTER nodes, E is the set of directed links (or edges), C gives the capacity of each link, and the link in the graph has a unit capacity, so $\mathrm{C(e)} = 1, \mathrm{e} \in \mathrm{E(\ v)}$. This means that each link transfer a symbol per unit interval and every symbol come from the elements of Galois field F.

For any node of G, let E+ indicating the set of output links, E- the set of input links. To note that in this paper when we mention link e, it may imply $e \in E^+(v)$, also $e' \in E^-(v)$. Now we assuming the G has single source $s \in V$, multiple sinks $t \in T$ and h is the multicast capacity. The h is the minimum cut of (S, T), namely the maximum flow. When the source s sends multicast messages $x_1, x_2, x_3, \dots, x_h$ to the sink T, the information arrives at a ForTER and the symbols transmitting on e (e∈E^+(v)) can be expressed by vector $y(e)$ which comes from the Galois field F. Let $y'(e)$ to be the symbolic vector of e' (e' ∈E^-(v)), then y(e) can be expressed with the linear combination of $y'(e')$, concluding the equation as follows:

$$y(e) = \sum_{e'} \beta_{e'}(e') y(e') \tag{1}$$

$\beta_{e'}(e')$ is the coefficient of the linear combination that randomly selected from the node's local coding vector of $\beta(e)$ and the length of $\beta(e)$ is determined by the number of the links that entering the node. So the local coding vector of link e describes the network operating characteristics of node.

The e's global coding vector $g_e(e)$ is the product of e' local coding vector $\beta_{e'}(e)$ and global coding vector $g_{e'}(e)$ that entering into node v.

$$g_e(e) = \beta_{e'}(e) g_{e'}(e) \tag{2}$$

For the purpose of the homogeneous of symbols, assuming that there are h virtual paths entering into the source node S, $e'_1, e'_2, \dots, e'_h$, so that the symbols $f(e'_1), f(e'_2), \dots, f(e'_h)$ in these virtual paths are equivalent to $x_1, x_2, x_3, \dots, x_h$ which can be illustrated to some extent in Figure 2. Concluding from the theory introduced above, the encoded symbol $y(e)$ is a linear combination of $x_1, x_2, x_3, \dots, x_h$ sent by source node. That is:

$$y(e) = \sum_{i=1}^{h} g_i(e)\, x_i \tag{3}$$

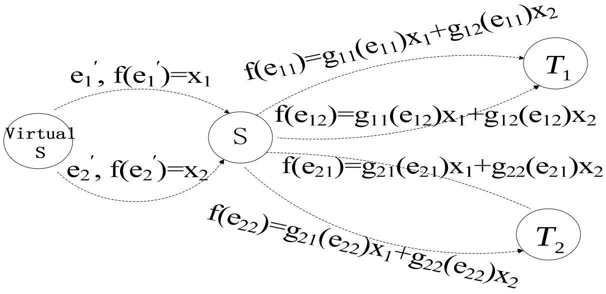

Figure 2. A Simple Model of Single Source-Multiple Sinks

Coding coefficients of the linear combination of the global coding vector are determined by $g(e) = [g_1(e), \dots, g_h(e)]$. As shown in Figure 2, a simple description of a source node S of the single channel, T1 and T2 are the sink nodes, x_1 and x_2 are the messages to be transmitted by the source S. For the homogeneous of the symbol, virtualizing a node which has two virtual paths into S. $f(e_{11}), f(e_{12})$ are the messages to be transmitted to T1, and $f(e_{21})$, $f(e_{22})$ is the information to T2. The global coding vector clearly presents the processing of the packet passing a node. Similarly with extending, supposing that there are h paths $e_1, e_2, e_3, \dots, e_h$ go into the sink node $t \in T$, then the coded packets the sink node received are available to the following formula:

$$\begin{bmatrix} y(e_1) \\ \vdots \\ y(e_h) \end{bmatrix} = \begin{bmatrix} g_1(e_1) & \cdots & g_1(e_h) \\ \vdots & \ddots & \vdots \\ g_h(e_1) & \cdots & g_h(e_h) \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_h \end{bmatrix} \tag{4}$$

3.2. Labeling Packets

In actual networks, symbol stream is transferred orderly on the links, and IP packets are made up of these symbol streams. If the MTU of the network data packet is 1400 bits and the Galois field | F | = 2^{16}, then each of the IP packet in the network can carry a number of bits N = 1400, else if the Galois field | F | = 2^{8}, each IP packet can carry a 700 bit data packet [11]. Thus, each packet in the network can be considered as the vector of the encoded symbols, such as $f(e) = [f_1(e), \dots, f_N(e)]$. Similarly, the source symbols x_i may also be regarded as a vector of symbols, such as $x_i = [x_{i,1}, \dots x_{i,N}]$. In that way, we can apply the algebraic relationship into an IP packet, if there is a downstream link e on source node carrying IP packet, the IP packet is the linear combination of source information $x_1, x_2, x_3, \dots, x_h$. If e is the downstream link of a non-source node, then IP packet can be represented by the linear combination of the data packets of upstream link e', namely,

$$f(e) = \sum_{e'} \beta_{e'}(e')y(e') = \sum_{i=1}^{h} g_i(e)\, x_i \tag{5}$$

It can also be rewritten and represented by the equation 6.

$$\begin{bmatrix} f(e_1) \\ \vdots \\ f(e_h) \end{bmatrix} = \begin{bmatrix} g_1(e_1) & \cdots & g_1(e_h) \\ \vdots & \ddots & \vdots \\ g_h(e_1) & \cdots & g_h(e_h) \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_h \end{bmatrix} \tag{6}$$

Because of the intermediate nodes selectively do linear processing on influent packets, after processing, their parameters definitely to be changed. So how can we make the sink node to know these changes, so the packets must carry this changing information by themselves. A good way is to mark these packets with labels. For this situation, we can enable the ForTER adding a prefix of global coding vector to packets and this global coding vector is belong to the link e which the packet is being on. As for how to labeling packet to obtain $f'(e) = [g(e), f(e)]$, we indeed place the prefix at the head of packet, specifically—adding i-th unit vector u_i as the prefix to i-th vector x_i .According to equation 1 and 2, $f'(e)$ can be expressed as follows:

$$f'(e) = \sum_{e'} \beta_{e'}[g(e'), y(e')] = \sum_{i=1}^{h} g_i(e)\, [u_i, x_i] \tag{7}$$

With matrix expression, another equivalent is fomulated as equation 8:

$$f'(e) = \begin{bmatrix} g_1(e_1) & \cdots & g_h(e_1) f_1(e_1) & \cdots & f_h(e_1) \\ \vdots & \ddots & \vdots \quad \vdots & \ddots & \vdots \\ g_1(e_h) & \cdots & g_h(e_h) f_1(e_h) & \cdots & f(e_h) \end{bmatrix} = G_t \begin{bmatrix} 1 & \cdots & 0 \; x_{1,1} & \cdots & x_{1,N} \\ \vdots & \ddots & \vdots \quad \vdots & \ddots & \vdots \\ 0 & \cdots & 1 \; x_{h,1} & \cdots & x_{h,N} \end{bmatrix}$$

Where:

$$G_t = \begin{bmatrix} g_1(e_1) & \cdots & g_h(e_1) \\ \vdots & \ddots & \vdots \\ g_1(e_h) & \cdots & g_h(e_h) \end{bmatrix} \tag{8}$$

Obviously saying that carrying global coding vector in packets make the overhead to be increased, such as | F | = 1400, h =40, then the overhead is 40/1400 = 2.86%. However the advantage is accessed: when packet received, the sink node does not need to know the topology of the entire network and on the basis of preceding packet operation to calculate the transition matrix G_t of network. It is very important in a rapidly changing network, because node failure, link failure and packet loss is likely to happen. Labeling packet makes decoding operation robust, as long as the network maintain the minimum Cut ≥ h, the sink node will be able to decode the received packet, and if minimum Cut ≤ h, the decoding fails.

3.3. Buffering Packets

The buffer module of ForTER architecture is mainly used to store packets that need to be coded, a timer is set in the buffer area in the meantime. For example, the size of the buffer module we can set as 2, the buffer maintains two coding queues. When a packet enters the coding queue buffer and it is not yet full, so the buffer will wait for a period of time, and this time is controlled by the timer. If the buffer queue is full before timeout, the buffer module will randomly put packets to the coding module for coding. And if the timer expires, the buffer queue is not yet full, the packets are not going to be processed but directly forwarded out.

It is not enough to merely provide the capability of linear random coding and labeling in the application of network coding to ForTER. In the sophisticated network, packets are transmitted according to sequence, factors of packet loss, congestion, traffic competition etc. will affect the throughput of link. Also to say is that the source information must be persistent flow of packets. If so, then we can package h packets into a data block. Now we make the hypothesis of existing a group of packets that related to the source information $x_{mh+1} \dots, x_{mh+h}, x_{mh+h}$, m is the ID of data block, here named to be generation ID. In order to facilitate tracking packets in the same generation, each coded packet is labeled with an generation ID , as shown in Figure 2.

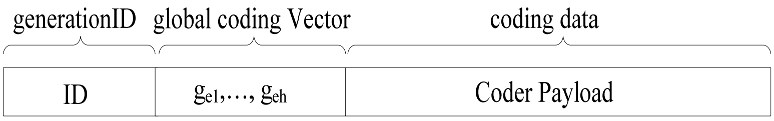

Figure 3. The Format of Coded Packet

Packets in data block that belong to the same generation are available to be synchronized via buffer. In the below buffer model (Figure 4), when packets arrive ForTER from different input links, packets of the same generation will be sent to a single buffer, and once the tick of output is fired, these packets will be sent to the coding module for random coding.

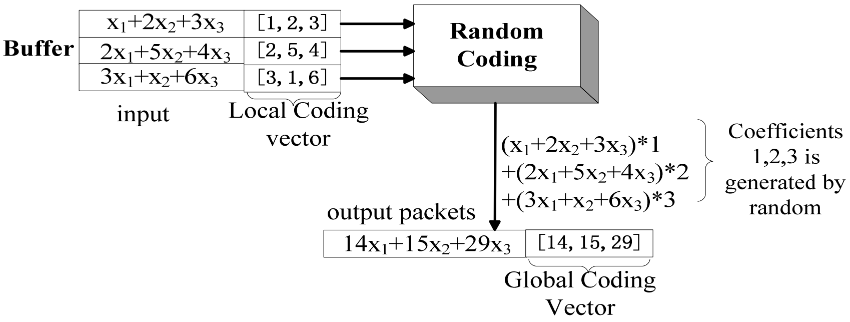

Figure 4. Packet's Processing of Input and Output from Buffer to Coding

In the linear random coding network the coding model consists of three parts: Source, Coding and Destination. The output information of Source is made up of data blocks, $X = [X_1, X_2, \dots, X_h]$ and all of them are random and independent process message. In Figure 4, h=3, X_1=[1,2,3], X_2=[2,5,4], X_3=[3,1,6].

Definition 1: Matrix SM is the Source matrix, indicating the connection relationship between the source and the ForTER multicast network.

$$SM_{i,j} = \begin{cases} a_{i,ej}, x_i = X(\text{tail}(e_j), P) \\ 0, \qquad \text{else} \end{cases} \tag{9}$$

For example, SM in Figure 4 is $\begin{bmatrix} 1 & 2 & 3 \\ 2 & 5 & 4 \\ 3 & 1 & 6 \end{bmatrix}$.

The Coding part implements the function of multicast packets' forwarding and coding. Utilizing PIM-SM protocol obtains multicast tree (S, G). For link Path, the output of node v is defined as v=tail (P). Otherwise, the input is defined as v=head (P), each link P carries a random process Y (P), which defined by equation 11.

Definition 2: Matrix FM is an adjacency matrix which represents the relationship between links in ForTER multicast network

$$FM_{i,j} = \begin{cases} f_{ei,ej}, \text{head}(e_i) = \text{tail}(e_j) \\ 0, \quad \text{else} \end{cases} \tag{10}$$

Stochastic process on link e is shown by equation 11. It is a process of linear combination, namely random process in the multicast network coding segment is the addition of the product of the source matrix and information vector and the product random process and adjacent matrix.

$$Y(j) = \sum_i^n a_{i,j} X_i + \sum_k^n f_{l,j}\, Y(k) \tag{11}$$

Destination judges and encodes the received multicast coding packets. The new multicast coded packet at sink β is represented by Z (β, i).

Definition 3: DM is the destination matrix, to show the relationship between the ForCES multicast network and the sink nodes.

$$DM_{i,j} = \begin{cases} d_{ej,l,Zi} = Z(\text{head}(e_j), p) \\ 0, \quad \text{else} \end{cases} \tag{12}$$

The stochastic process of destination node (the last hop of ForTER) received is a product by random process and sink node matrix:

$$\gamma_{\beta,i} = \sum\nolimits_{\text{head}(l)=\beta} d_{\beta i,l}\, Y(p) \tag{13}$$

Definition 4: The transition matrix TM indicates the transfer matrix of entire network, defined as follows:

$$TM_{i\times ij} = [M(L_1)_{i\times i}, M(L_2)_{i\times i}, \dots, M(L_j)_{i\times i}]$$

where: $M = SM(I - FM)^{-1}DM^T$ and I is a unit matrix of $|E| \times |E|$ (14)

Matrix SM and matrix RM simply illustrate the linear relationship of the input and output of random process, but no essential contribution to the transition matrix. In order to find an impulse response of an input random process X (v, i) and output random process γ (v, i), we add all links to them. Due to link nodes are determined by $I + FM + FM^2 + FM^3 + \cdots$, F is a nilpotent matrix, so there will be a positive integer N to make FM^N is a zero matrix. So according to the Taylor's Series, we can get $(1 - FM)^{-1} = I + FM + FM^2 + FM^3 + \cdots$, then achieving the above equation.

3.4. Decoding Packets

According equation 4, ForTER can get global coding vector matrix from packets of output links, i.e. the transition matrix. After an inversion and Gaussian elimination, we can recover the source $x_1, x_2, x_3, \dots, x_h$.

$$\begin{bmatrix} x_1 \\ \vdots \\ x_h \end{bmatrix} = G_t^{-1} \begin{bmatrix} y(e_1) \\ \vdots \\ y(e_h) \end{bmatrix} \tag{15}$$

The global coding vector matrix ForTER received is actually tends to be a lower triangular matrix, when a node receives more than k packets and it is able to decode the packets. For example, assuming that a node receives three packets, $x_1, 2x_1 + 2x_2, 3x_1 + 4x_2 + 3x_3$ where x_1, x_2, x_3 are the original data packets.

$$\begin{bmatrix} x_1 & 0 & 0 \\ 2x_1 & 2x_2 & 0 \\ 3x_1 & 4x_2 & 3x_3 \end{bmatrix} = \begin{bmatrix} b1 \\ b2 \\ b3 \end{bmatrix} \quad (16)$$

ForTER can decode x_1 first, and then according to $x_1$to decode $x_2$and x_3. Such solution is pre-decoded, it does with no need for all data packets have been received, but according to the received packet in accordance with a decoding, thus reducing the decoding delay.

4. Evaluation

To motivate the efficiency and robustness of the design, in this section we build an IP multicast network with ForTERs, which including the classic shortest path tree constructed by the reverse path from the source node to each sink ,then we evaluate both the ForTER's throughput , performance and CPU utilization.

4.1. Throughput

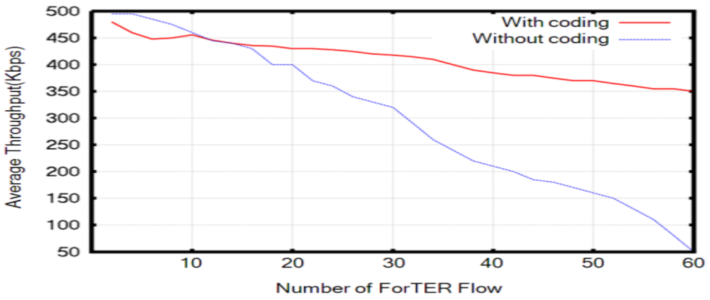

Figure5. Comparison of Throughput With The Effect of Flows

4.2. Performance Overhead

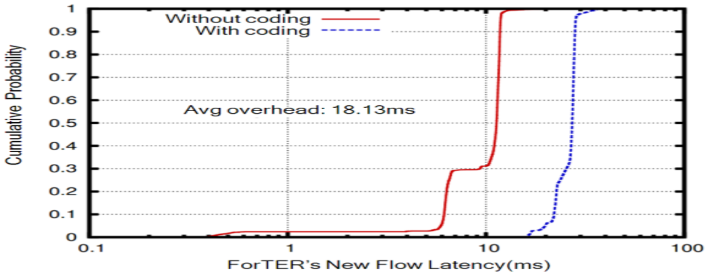

Figure 6. CDF of Network Coding Overhead for ForTER's New Flow

4.3. CPU Utilization

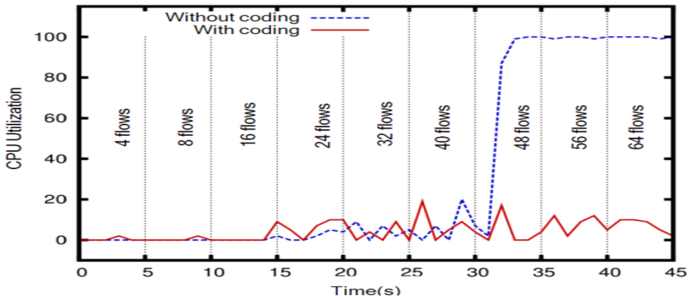

Figure 7. ForTER's New Flow Throttling Prevents a Malicious Attack from Saturating CPU

To quantify the promotion of throughput by network coding, we give two experiments that monitor the input of sink with and without coding enabled in ForTER network. In our experiment (

Figure5), increases flow number gradually (1->60), the result shows that the coding features can prevent the throughput from decreasing sharply specially from 12 flows on.

Adding an additional function of coding to flow adds overhead to the system of ForTER. However, as a result of our design, the coding does not add overhead to the flow. That is, with coding, packets are forwarded at full line rate. To quantify this overhead, we measure the time between sending the packet and receiving the new flow with and without the coding using libpcap. Our results (Figure 6) show that the coding increases new flow latency, i.e., the additional time from source to sink, by 18ms on average. For latency sensitive applications, e.g., web services in large data centers, 18ms may be too much overhead. How-ever, new flow adds 12ms latency on average even without the coding.

To quantify our ability to protect CPU resource, we show two experiments that monitor CPU-usage overtime of ForTER with and without coding enabled. In our experiment (Figure 7), increases flow step by step (4, 8, 16,…, 64), we show the 1-second-average ForTER's CPU utilization over time, and the coding features reduce the switch utilization from 100% to a conFigureurable amount and also note that without coding ForTER could handle less than 48 flow simultaneously without appreciable CPU load, but immediately goes to 100% load when the number of flow hits 48.

References

[1] R Ahlswede, N Cai, SY Li, R Yeung. *Network information flow*. IEEE Trans. on Information Theory. 2000; 46(4): 1204-1206.

[2] J Halpern, J Hadi Salim. *Forwarding and Control Element Separation (ForCES) Forwarding Element Model.* IETF RFC 5812. 2010.

[3] L Yang, R Dantu, T Anderson. *Forwarding and Control Element Separation (ForCES) Framework.* IETF RFC 3746. 2004.

[4] Christoph Schuba et.al. *Integrated network service processing using programmable network devices.* Technical Report From Sun Microsystems, Inc. Mountain View, 2010.

[5] Nick Feamster, Hari Balakrishnan et.al. *The Case for Separating Routing from Routers.* In Proc. ACM SIGCOMM. 2004; 5-12.

[6] Robert Haas, Toshiaki Suzuki. *Architecture of the Flexinet ForCES-based Control Point.* In Proc. 81th IETF meeting. Quebec. 2011.

[7] M Hidell, P Sjödin, O Hagsand. *Control and Forwarding Plane Interaction in Distributed Routers.* In Proc. 4th IFIP-TC6 Networking Conference Networking. 2005.

[8] Weiming Wang, Ligang Dong, Bin Zhuge, *ForTER-An Open Programmable Router Based on Forwarding and Control Element Separation*, In Proc. DCABES 2011, 2: 1069-1077.

[9] SY Li, RW Yueng, N Cai. Linear network coding. *IEEE Trans. on Information Theory*. 2003.

[10] P Sanders, S Egner, L Tolhuizen, *Polynomial time algorithms for network information flow*, In Proc. SPAA'03. 2003; 286-294.

[11] S Jaggi, P Sanders, et.al. *Polynomial time algorithms for multicast network code construction.* IEEE Trans. on Information Theory. 2005; 51(6): 1973-1982.

[12] E Erez, M Feder, *Efficient network codes for cyclic networks*. In Proc. ISIT'05. 2005; 1982-1986.

[13] T Ho, R Koetter, M Medard, D R Karger, M Effros. The benefits of coding over routing in a randomized setting. *IEEE International Symposium on Information Theory*. 2003.

[14] T Ho, R Koetter, M Medard, DR Karger, M Effros, J Shi, B Leong. *A random linear network coding approach to multicast.* IEEE Trans. on Information Theory. 2006; 52(10): 4413-4430.

[15] Samini Subramaniam, Su-Cheng Haw, Poo Kuan Hoong. *Bridging XML and Relational Databases: An Effective Mapping Scheme based on Persistent.* IJECE. 2012; 2(2); 239-246.

[16] PA Chou, Y Wu, K Jain. *Practical network coding*. In Proc. 41st Annual Allerton Conference on Comunication Control and Computing. 2003.

[17] Chuanhuang Li, Weiming Wang. *Solutions and Comparison of Performance to the Key Technologies of PIM-SM under ForCES Architecture's Router.* In Proc. 8th ICN, 2009.

[18] Jiangang Lu, Jie You, Qinmin Yang. *Nonlinear Model Predictive Controller Design for Identified Nonlinear Parameter Varying Model.* TELKOMNIKA. 2012; 10: 514-523.

[19] Nasaruddin, Melinda, Ellsa Fitria Sari. *A Model to Investigate Performance of Orthogonal Frequency Code Division Multiplexing.* TELKOMNIKA. 2012; 10: 579-585.