

Centrality-Based Network Coding Node Selection Mechanism for Improving Network Throughput

Tae-hwa Kim*, Hyungwoo Choi*, Hong-Shik Park*

* Department of Information & Communications Engineering, KAIST, Daejeon, Republic of Korea

thkim0@kaist.ac.kr, neocrack@kaist.ac.kr, parkhs@ee.kaist.ac.kr

Abstract— The problem of minimizing the number of coding nodes is caused by network coding overhead and is proved to be NP-hard. To resolve this issue, this paper proposes Centrality-based Network Coding Node Selection (CNCNS) that is the heuristic and distributed mechanism to minimize the number of network coding (NC) nodes without compromising the achievable network throughput. CNCNS iteratively analyses the node centrality and selects NC node in the specific area. Since CNCNS operates with distributed manner, it can dynamically adapt the network status with approximately minimizing network coding nodes. Especially, CNCNS adjusts the network performance of network throughput and reliability using control indicator. Simulation results show that the well selected network coding nodes can improve the network throughput and almost close to throughput performance of a system where all network nodes operate network coding.

Keywords— Network coding, Throughput, Centrality, Degree, Weight

I. INTRODUCTION

Before the appearance of network coding, the max-flow min-cut theorem¹ can't be established in the multicast communication scenario under the traditional routing. R.Ahlsweide et al. theoretically prove network coding can improve the network throughput through a combination of packets in the network in [1]. T.Ho et al. proposed Random Linear Network coding (RLNC) which is coding over routing for multicasting from several sources over a network in [2][3]. RLNC allows that network intermediate nodes generate coded packets which are random linear combinations from incoming packets with coefficients randomly chosen from a Galois field and transmit these coded packets through the all its outgoing links like as broadcasting transmission. Random property of RLNC makes it possible robust, distributed transmission of coded packets without communication with neighbors and higher decoding success probability. P.A. Chou et al. evolve the RLNC with encoding vector, buffer management and synchronization mechanism to adapt the network coding in practical network in [4].

Not only network coding is remarkable in improvement of throughput performance, but also according to [5] network coding has other advantages such as minimizing energy

consumption, minimizing delay through reduction of transmission number. And M. Langberg et al. show unreliable network can apply network coding with the purpose of protection in [6].

Although the above benefits, network coding incurs overheads such as encoding and decoding computation, delay, additional cost in existing network. As the data travels through the network, these overheads at intermediate nodes is increasing constantly. N. Cleju et al. insist the few well selected nodes are able to bring a large throughput gain in [7]. Shortly, increasing number of network coding nodes, it is not always increasing throughput gain. And if specific network node turns into network coding node, it is more effective in terms of throughput improvement.

Several previous works try to find minimal number or specific placement of network coding nodes without compromising the achievable network throughput. In practical manner, selection of network coding nodes in an overlay media streaming system is introduced to reduce the decoding delay in [7]. And their results show there is more effective nodes for network coding to reduce decoding delay within improvement of throughput performance. For diversity, redundant packets are intentionally added and this mechanism iteratively selects the nodes for network coding such that the decoding delay is decreased. In [8], the problem of minimizing the number of coding nodes is proved to be NP-hard. A heuristic mechanism for solving the problem of minimizing the number of network coding operations is suggested in [9]. They approximate the minimum number of NC nodes in polynomial time for constant number of receivers. Kim et al. propose a solution based on genetic algorithms which is aimed to reduce the resource by network coding and complexity of encoding in [10].

Considering the problem of minimizing the number of coding nodes proven for NP-hard, this paper propose Centrality-based Network Coding Node Selection (CNCNS) that is designed to heuristically find network coding node with centrality of network node without constraint on network environment. Under CNCNS, a network node adjusts network coding on-off in tandem with network centrality which can be defined through node degree and strength. The simulation results show that a few well selected effective nodes can improve the network throughput and these results close to throughput performance of conventional RLNC. Effective

¹ MaxFlow-MinCut theorem is maximum flow rate between the source and the destination is known to be equal to the minimum cut between them.

network coding nodes can approximately close to achievable maximum throughput. Thus, CNCNS make it possible to apply network coding partially in the network with improvement of network throughput by network coding. The following sections will present the detailed operation of CNCNS mechanism and throughput gain and end-to-end delay in random network.

II. CENTRALITY-BASED NETWORK CODING NODE SELECTION MECHANISM

Let us observe CNCNS that is a heuristic mechanism which provides a practical way of approximating the minimum number of NC nodes required for achieving network capacity. To select network coding node in a distributed manner without central information, we define an area which is the set of nodes with a focal node as the center and its neighbor nodes.

In network, there are concentrated nodes which are more connected and communicated with its neighbors than other nodes. These nodes have more incoming traffic and opportunity to transmit traffic into network. It means if these nodes turn into NC node, it can more generate innovative packets and transmit packets to receivers with improving the network throughput. A centrality is a tendency to be these concentrated nodes. To estimate the centrality of network node, we define a node degree and strength. The node degree represents the number of flow which is traffic connection passing through the node from source to receiver(s). And the node strength represents the sum of link bandwidth connected to the other node. Based on the flow direction, we classify those two parameters into input and output.

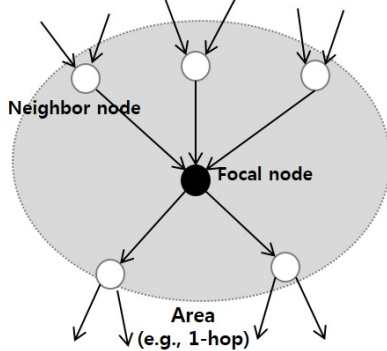


Figure 1. Example of an area

As shown in Figure 1, there is a focal node which can be any node in the network. In the area, the focal node is connected to its neighbor nodes within the same hop distance. Degree of focal node i (d_{in}) is given by:

$$d_{in}(i) = \sum_{j=1}^N x_{ji}, \quad d_{out}(i) = \sum_{j=1}^N x_{ij} \quad (1)$$

where $i, j \in \{1, 2, \dots, N\}$, N is the number of network nodes, i is the focal node and j represents neighbor nodes in the network. If there is traffic connection between node i and node j , x_{ij} is 1 or not 0. When node j sends messages to node i , $d_{in}(i)$ is the input degree of the node i . Otherwise node i sends messages to node j , $d_{out}(i)$ is the output degree of the node i .

Strength of the node i (s_{in}) is given by:

Algorithm 1 The operation mechanism of CNCNS

```

1: Initialize the number of NC node in each area:  $N_c = 1$ 
2: for  $i = 1$  : All network nodes do
3:   Compute  $C(i)$  using Eq. (1)-(5)
4:   for  $j = 1$  : neighbor nodes of  $i$  do
5:     Compute  $C(j)$  using Eq. (1)-(5)
6:     Send  $C(j)$  of node  $j$  to node  $i$ 
7:   end for
8:   Select NC node which has maximum value of  $C(i)$ 
9:   If NC node is not node  $i$ , then
10:    Send the messages to NC node notification to node  $j$ 
11:   else
12:    Turn node  $i$  into NC node
13:   end if
14: end for

```

$$s_{in}(i) = \sum_{j=1}^N w_{ji}, \quad s_{out}(i) = \sum_{j=1}^N w_{ij} \quad (2)$$

where i is the focal node, j represents neighbor nodes in the network. If there is traffic connection between node i and node j , w_{ij} is the value of link bandwidth between node i and node j or not 0. Thus, $s_{in}(i)$ is a sum of input strength of the node i and $s_{out}(i)$ is a sum of output strength of the node i .

Based on node degree and strength, we compute input centrality and output centrality. Input centrality (c_{in}) and output centrality (c_{out}) of node i is given by:

$$c_{in}(i) = d_{in}(i) \times \left(\frac{s_{in}(i)}{d_{in}(i)} \right)^\alpha = d_{in}(i)^{(1-\alpha)} \times s_{in}(i)^\alpha \quad (3)$$

$$c_{out}(i) = d_{out}(i) \times \left(\frac{s_{out}(i)}{d_{out}(i)} \right)^\alpha = d_{out}(i)^{(1-\alpha)} \times s_{out}(i)^\alpha \quad (4)$$

where α is a positive weighted parameter that can control between node degree and strength. If this parameter is between 0 and 0.5, then a degree of node is powerful value, whereas if it is between 0.5 and 1, a strength is powerful value. In this paper, we set this value to 0.5 arbitrarily.

Finally, centrality of node (C) is given by:

$$C(i) = c_{out}(i) \times \left(\frac{c_{in}(i)}{c_{out}(i)} \right)^\beta = c_{out}(i)^{(1-\beta)} \times c_{in}(i)^\beta \quad (5)$$

where β is a positive weighted parameter to control between packet transmission rate and packet innovativeness. For example, if this parameter is between 0 and 0.5, output centrality is more powerful than input centrality. This means if this node turns into NC node, the volume of output traffic from node i to neighbors is larger and this increase the packet transmission rate. Whereas if it is between 0.5 and 1, input centrality is more powerful than output centrality. This means if this node turns into NC node, the packet innovativeness of node i is higher and this increase the decoding probability at the receiver. Thus this parameter is an indicator to control weight between packet transmission rate and packet innovativeness.

To select NC nodes, CNCNS is composed of following procedures as shown in Alg.1. **(i) Compute centrality:** to begin with, CNCNS initializes the number of NC node in each area (N_c) (=1) which can be decided by network administrator. Based on computed degree and weight using equation (1)-(2), focal node computes the centrality itself using equation (3)-(4). And connected neighbor nodes to focal node compute its centrality. **(ii) Compare the centrality of focal node and its neighbor nodes:** next, the focal node is one of all network node at a time and neighbor nodes are connected to focal node in a one hop area. To compare the centralities of focal node and connected neighbor nodes, focal node collects the centrality of neighbor nodes. And then, as much as N_c , focal node decides the NC node(s) which has the maximum value in the area. If selected NC node is the one of the neighbor nodes, focal node makes it known. Otherwise, focal node operates NC node.

Above overall procedures of CNCNS iteratively operate to accustom the variable network status, thus NC nodes are dynamically changed by centrality.

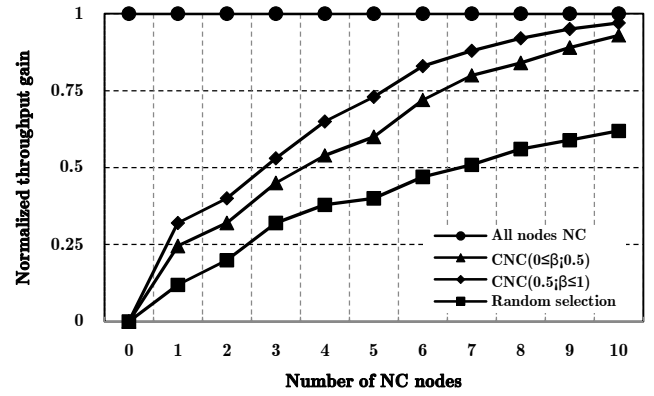
CNCNS objects to select the NC node individually in the defined as an area. Thus CNCNS can be adopted in network without constraint on network environment i.e., the number of network nodes, links, receivers, etc. In the area, NC node is chose as much as the prescribed NC node number(s). Areas are overlapped each other, thus occasionally more than one node work NC operation in the area.

III. PERFORMANCE EVALUATION

This section evaluates performance of the CNCNS against general network coding node selection mechanisms such as the all network nodes operate network coding based on RLNC and the network coding nodes are randomly selected by using our computer simulator. The results were obtained with the following assumptions: single source connect with multiple receivers; and the packet arrival process follows the Poisson process with rate λ . We use the random network which consists of 50 nodes with network diameter of 6. To put weight between packet transmission rate and packet innovativeness, the indicator parameter β is properly chosen within above defined bounds. For the precise computation, redundant packets are deleted by receiver, thus the network throughput is analyzed by decoded packets.

Figure 2 shows the normalized throughput gain of CNCNS compared with other mechanisms as increasing NC nodes in the network. CNCNS has an approximate results to the network throughput in which all network nodes operate network coding. When the indicator parameter β is between 0.5 and 1, CNCNS selects the NC nodes which has more innovative incoming packets, thus effective packets to decode original data are more transmitted. It means packet innovativeness is meaningful to improve the network throughput.

The normalized average end-to-end delay including decoding delay versus the number of NC nodes is presented in Figure 3. Network coding causes the overhead of decoding delay because a receiver waits the sufficient independent



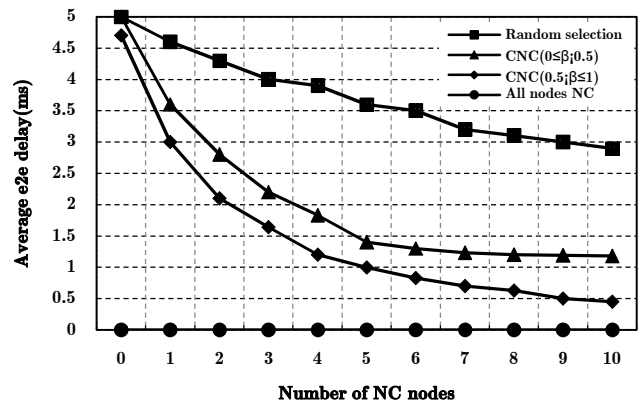
packets to decode original data. When the indicator parameter β is between 0.5 and 1, as the more innovative packets are

Figure 2. Normalized throughput gain versus the number of NC nodes

Figure 3. Normalized average end-to-end delay versus number of NC nodes

transmitted, the more independent packet is received. Thus the decoding delay is reduced, therefore the performance of end-to-end delay is improved.

To balance packet transmission rate and packet innovativeness, we can adjust the value of indicator parameter β with consideration of network characteristics. For example, in the case where the network supports real-time applications to satisfy end-to-end delay requirement, the above parameter β is set to be between 0.5 and 1. Otherwise, when the network undergoes severe loss to guarantee the reliability, the above parameter β is set to be between 0 and 0.5.



IV. CONCLUSION

To resolve the problem of minimizing the number of coding nodes, we propose a new CNCNS. Without constraints on network environment, a network node can heuristically adjust network coding on-off operation based on node centrality. CNCNS improves the network throughput with the minimizing the number of coding nodes. Since CNCNS operates under distributed manner, it is simple while not requiring centralized information of network. Through the control indicator, CNCNS is useful for enhancing a performance of the network throughput and the network

reliability. We show the well selected network coding node by CNCNS can improve the network throughput and end-to-end delay. And CNCNS achieves the almost same throughput as a system where all network nodes operate network coding.

ACKNOWLEDGMENT

This research was funded by the MSIP (Ministry of Science, ICT & Future Planning), Korea in the ICT R&D Program 2013.

REFERENCES

- [1] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Trans. Inf. Theory*, vol. 46, no. 4, pp. 1204–1216, Jul. 2000.
- [2] T. Ho, M. Medard, J. Shi, M. Effros, and D. R. Karger, "On randomized network coding," in *Proc. 41st Allerton Annual Conf. Communication, Control, and Computing*, Monticello, IL, Oct. 2003.
- [3] T. Ho, M. Medard, R. Koetter, D. R. Karger, M. Effros, S. Jun, and B. Leong, "A random linear network coding approach to multicast," *IEEE Trans. Inf. Theory*, vol. 52, no. 10, pp. 4413–4430, Oct. 2006.
- [4] P. A. Chou, Y. Wu, and K. Jain, "Practical network coding," in *Proc. 41st Allerton Conf. Communication, Control, and Computing*, Monticello, IL, Oct. 2003.
- [5] P. A. Chou and Y. Wu, "Network Coding for the Internet and Wireless Networks," *Signal Processing Magazine, IEEE*, vol. 24, no. 5, pp. 77–85, Sept. 2007.
- [6] A. E. Kamal, A. Ramamoorthy, L. Long and L. Shizheng, "Overlay Protection Against Link Failures Using Network Coding," *Networking, IEEE/ACM Transactions on*, vol. 19, no. 4, pp. 1071–1084, Aug. 2011.
- [7] N. Cleju, N. Thomos, and P. Frossard, "Network coding node placement for delay minimization in streaming overlays," in *Proc. Int. Conf. Communications (ICC'10)*, Cape Town, South Africa, May 2010.
- [8] M. Langberg, A. Sprintson, and J. Bruck, "The encoding complexity of network coding," *Information Theory, 2005. ISIT 2005. Proceedings. International Symposium on*, vol., no., pp. 1987–1991, 4–9 Sept. 2005.
- [9] J. Thibault and M. Hajiaghayi, "Minimizing network coding nodes for multicast," *IEEE Trans. Inf. Theory*, vol., no., pp., 2009.
- [10] M. Kim, M. Medard, V. Aggarwal, U.-M. O'Reilly, W. Kim, C. W. Ahn and M. Effros, "Evolutionary Approaches To Minimizing Network Coding Resources," *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, vol., no., pp. 1991–1999, 6–12 May 2007.