In-network congestion control toward enhanced network resource utilization

Mitsuhiro Watanabe
Graduate School of Information Sciences
Tohoku University
Sendai, Japan
mitsuhiro.watanabe.r7@dc.tohoku.ac.jp

Go Hasegawa
Research Institute of Electrics Communication
Tohoku University
Sendai, Japan
hasegawa@riec.tohoku.ac.jp

***Abstract*—As the Internet becomes larger scaled and more diversified, the traditional end-to-end (E2E) congestion control faces various problems. In this paper, we propose a novel congestion control architecture, called *in-network congestion control (NCC)*. Specifically, by introducing one or more nodes (NCC nodes) on an E2E network path, we divide the network path into multiple sub-paths and maintain a congestion-control feedback loop on each sub-path. In each sub-path, a specialized congestion control algorithm can be applied according to its network characteristics. This architecture can provide various advantages compared with the traditional E2E congestion control, such as higher data transmission throughput, better per-flow fairness, and incremental deployment nature. In this paper, we describe NCC's advantages and challenges, and clarify its potential performance by numerical evaluation results.**

***Index Terms*—Congestion Control, Transmission Control Protocol (TCP), in-Network Processing, End-To-End Principle**

I. INTRODUCTION

Congestion control mechanisms used on the current Internet have been unchanged for over thirty years and they are based on an end-to-end (E2E) principle. The E2E congestion control mechanisms assume the network between a sender and a receiver as a black box, and no explicit information on the network congestion is obtained. Instead, as shown in Figure 1, they construct a feedback loop between the sender and receiver by sending data packets and receiving corresponding acknowledgment (ACK) packets. Specifically, based on the observations such as packet loss events, round trip times (RTTs), and sending/receiving intervals of packets, the sender finds some relationships between the observed results and network congestion and controls the data transmission rate.

The relationships between the observation of the network congestion and the ideal control become complex in a highly diverse network environment, and it is difficult to find an ideal transmission rate. Congestion control mechanisms that are currently used (CUBIC [1], BBR [2], etc.) are based on heuristic algorithms using one or a few index values. Due to research and development activities over thirty years, they have excellent properties that their performance does not deteriorate significantly in a variety of network environments. However, they sometimes fail to optimize their performance in a specific network environment especially when a network path between a sender and a receiver traverses multiple networks with various characteristics [3].

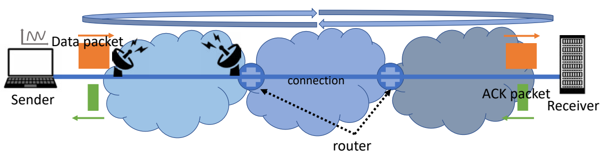


Fig. 1: End-to-end (E2E) congestion control

Based on the above discussion, the authors of this paper believe that it is hard to realize an ideal congestion control using only information obtained from the E2E observation. Therefore, in this paper, we propose a novel congestion control architecture, called *in-network congestion control (NCC)*. Specifically, by introducing one or more nodes (called NCC nodes) on the E2E network path, we divide the network path into multiple sub-paths and use a series of congestion-control feedback loops between the sender and receiver. In each sub-path, a specialized congestion control algorithm can be applied according to the network characteristics of the sub-path. We can expect high performance of congestion control algorithms because the sub-path would have a shorter propagation delay and simplified network characteristics. In other words, using the proposed congestion control architecture, we can maximize the advantages of congestion control algorithms developed over the years.

The main objective of this paper is to explain the fundamental architecture of NCC and describe strong points and issues to be solved for the realization of the proposed architecture. We also show some evaluation results to reveal the potential performance of the proposed architecture.

The rest of this paper is organized as follows. Section II describes the related work. In Section III, we describe the overview of NCC and discuss its advantages and challenges. In Section IV, we present performance evaluation results and clarify the potential performance of the architecture. Finally, in Section V, we present the summary of this paper and future work.

II. Related work

There are some existing methods to obtain explicit information on network congestion from network nodes. For example, in Explicit Congestion Notification (ECN) [4] and eXplicit Control Protocol (XCP) [5], a router writes congestion information to packets passing through and provides network congestion feedback to a sender. However, these methods employ the conventional E2E congestion control mechanisms based on the obtained information. Also, these mechanisms only convey the congestion information at the bandwidth-bottlenecked router, which is not enough for efficient congestion control in a diversified Internet environment.

With the advance of virtualization technologies for network and computing resources, it has become common to perform various and flexible control functions on network nodes by software. Also, in both core networks and edge networks, Software Defined Networking (SDN) technologies are now in operation [6]. The trend of the softwarization of network operations is an important factor for the realization of the proposed architecture in this paper. However, to the best of our knowledge, there is no existing work on in-network softwarization of congestion control functions.

On the other hand, methods to divide and relay communication protocols between senders and receivers are already in use, including Web proxies and firewalls. Most of them are used for security and policy enforcement, not for congestion control or utilizing network resources effectively. Performance Enhancing Proxy (PEP) [7] is a proxy mechanism for network nodes to improve the performance of E2E communication protocols such as TCP. A typical example is a WAN acceleration device [8]. PEP at the transport layer to improve the efficiency of congestion control has also been studied [9]. The proposed congestion control architecture in this paper can be considered as an evolution of the PEP concept.

III. In-network congestion control (NCC)

A. Overview

Figures 1 and 2 shows the traditional E2E congestion control and the in-network congestion control proposed in this paper. Figure 1 depicts the traditional E2E congestion control, where a single connection between a sender and a receiver is established. The connection passes through three networks which have different characteristics. On the other hand, in Figure 2, by introducing two NCC nodes on the network path, the network path is divided into multiple sub-paths, and three sub-connections are established for relaying data from the sender to the receiver. Each sub-connection conducts congestion control individually. In other words, we use a series of congestion-control feedback loops between the sender and receiver.

B. Advantages

1) Shortened congestion-control feedback loop: One of the reasons for the poor performance of congestion control mechanisms on the current Internet is a large propagation delay between a sender and a receiver. This is because the

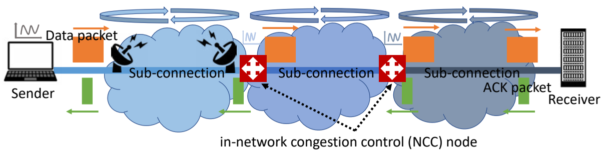


Fig. 2: In-network congestion control

congestion control algorithms are based on E2E, which can be mathematically explained [10], [11]. The following equations for NewReno and CUBIC are mathematically derived in [10], [11], respectively. Here, ρ_r and ρ_c are TCP throughput for NewReno and CUBIC respectively, RTT is a RTT of a TCP connection, T_0 is the initial value of retransmission timeout, p is the packet loss rate, and b is a parameter related to delayed ACK.

$$\rho_\mathrm{r} = \frac{1}{RTT\sqrt{\frac{2bp}{3}} + T_0 \min\left(1, 3\sqrt{\frac{3bp}{8}}\right) p\left(1+32p^2\right)} \tag{1}$$

$$\rho_\mathrm{c} = \frac{\max\left(1.3004\left(\frac{RTT}{p}\right)^{\frac{3}{4}}, \frac{1.31}{\sqrt{p}}\right)}{RTT} \tag{2}$$

As shown in Equations (1) and (2), the TCP throughput depends mainly on the RTT and packet loss ratio of the TCP connection. If the RTT or packet loss ratio increases, the throughput becomes small. However, with the proposed architecture, the RTT and packet loss ratio of sub-connections divided by NCC nodes would be smaller. When the processing speed of NCC nodes is large enough, the E2E data transfer efficiency can be improved by just introducing NCC nodes.

2) Dedicated congestion control on each sub-connection: Figure 3 shows the data transmission with the proposed architecture over the network consisting of a satellite communication network, a cellular network, a high-speed backbone network, and a datacenter network. Three NCC nodes are introduced at the borders of the networks. When we use a single E2E connection between the sender and the receiver, it passes through multiple networks which have various characteristics, which makes the optimization of the data transmission to be difficult. On the other hand, with NCC nodes, the network path is divided into sub-paths that have simpler network characteristics. Therefore, we can deploy congestion control algorithms specialized for the network characteristics of each sub-path, for example, for satellite networks [12] and data centers [13].

3) Contribution to per-flow fairness: Most current congestion control algorithms, including those using machine learning, are focused primarily on maximizing the performance of their own flows, while fairness among competing flows is not directly considered. In general, with E2E congestion control, per-flow fairness in a diversified Internet cannot be achieved especially when the flows pass through different

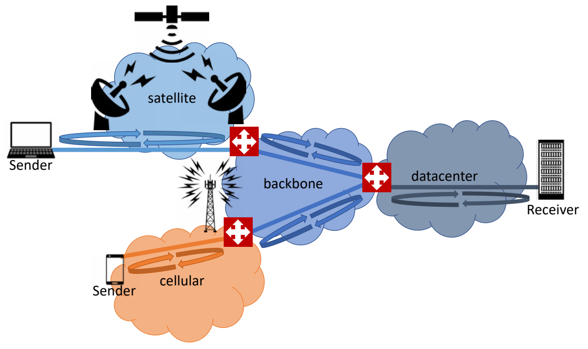


Fig. 3: Applying dedicated congestion control algorithms to sub-paths

network paths. On the other hand, in NCC architecture, sub-connections on a certain sub-path divided by two NCC nodes experience an identical network environment. Thus, we can expect enhanced fairness among sub-connections, bringing the improved fairness on E2E data transmission. Note that this fairness improvement can be obtained only by introducing NCC nodes to the network and it does not require a specialized congestion control algorithm for the fairness improvement.

4) Incremental deployment: Even when there is only one NCC node on an E2E network path, a sender can benefit from the proposed architecture. Furthermore, the more NCC nodes are introduced in the network, the greater the benefit becomes. Also, due to the per-flow nature of the proposed architecture, we can easily apply the proposed architecture to a part of flows that pass through NCC nodes. Also, we can easily parallelize the NCC operations with multiple NCC nodes. Therefore, we can control the number of flows that NCC nodes deal with, according to the performance and load of NCC nodes, by which we can avoid an urgent overload and underutilization of NCC nodes. From a different viewpoint, applying the proposed architecture to a part of flows may be one of the network operator's service differentiation to its customers.

C. Challenges

1) Processing overhead on NCC nodes: As NCC nodes process packets at the higher layer than the traditional IP networks, processing overhead becomes larger than the traditional layer-3 routers. However, we believe that this can be solved with the recent progress in computing performance and its lower cost. The parallelizing characteristics of the proposed architecture described above can also solve the problem by just increasing the number of NCC nodes to be deployed.

For relaying the data packets with multiple congestion-control feedback loops, NCC nodes must store a larger number of packets for a longer time than the traditional layer-3 routers. Especially when a NCC node is located at the border of two networks that have different speeds, many packets are accumulated at the NCC node, which increases the memory utilization. One of the solutions for this problem is to exploit the backward feedback using the advertised window size included in TCP ACK packets. Although similar approaches have been proposed for E2E congestion control [14], we need to evaluate the performance when they are applied to NCC nodes.

2) Fault tolerance and security issues: Unlike the conventional congestion control architecture, NCC nodes must maintain connections' information at internal network nodes. Therefore, complex functions are required to make NCC nodes redundant and deal with NCC node failures. We believe that the existing fault tolerant mechanisms for upper-layer protocol processing such as firewalls [15] can be applied to NCC nodes. Also, introducing upper-layer protocol processing functions inside the network increases the risk of vulnerability attacks. The effect of such attacks and protection mechanisms inside the network should be investigated.

3) Flow's requirement for specialized congestion control: Some existing congestion control algorithms are designed not only to maximize throughput but also to achieve low latency [16] or to transmit with a lower priority than competing traffic [17]. Some applications need to choose such congestion control algorithms. Therefore, a mechanism is required that allows senders to select congestion control algorithms deployed between NCC nodes.

4) NCC protocols: The authors in this paper consider that the proposed architecture should be realized as a general mechanism used in the whole Internet. Since most applications on the current Internet use TCP as a transport-layer protocol, the proposed architecture must be transparently integrated with TCP-based applications, at least as a short-term solution. One of the possible candidates is a tunneling protocol such as Internet VPN protocols. In this case, the duplication of congestion controls on a network path (TCP's one and NCC's one) [18] is one of the performance problems to be solved. The authors consider that exploiting UDP-based protocols such as QUIC [19] instead of TCP is one of the possible long-term solutions.

In general, there are two ways to divide E2E network paths: transparently and explicitly. In transparent methods, senders cannot use NCC nodes when the network path between senders and receivers varies over time. The combination of source routing and NCC is one of the possible solutions. On the other hand, in explicit ways, senders must be informed where NCC nodes are located before starting data transfer. For example, we can solve this by adding information on the location of NCC nodes as a part of the routing information in Border Gateway Protocol or by using a mechanism such as Domain Name System.

IV. Performance Evaluation

The performance of NCC architecture was evaluated numerically by using the analytical models in Equations (1) and (2). We show its evaluation results to confirm the advantages described in Subsection III-B.

A. Evaluation settings

Figure 4 shows the network topologies used for performance evaluation. Figure 4 (a) shows the dumbbell topology used to evaluate the fundamental performance of NCC. The parking lot topology in Figure 4 (b) is used to assess the effect of the number of NCC nodes in the network. As shown in Figure 4 (c), the nation-wide network topology that models the Japanese optical network [20] is used to evaluate the performance in a realistic network.

The node labeled "Node" in Figures 4 (a) and 4 (b) and the node labeled with a number in Figure 4 (c) correspond to the network nodes. In the dumbbell topology, one hundred senders connect with Node 1 and one hundred receivers connect with Node 2. In the parking lot topology and the nation-wide topology, one hundred senders and one hundred receivers connect to each network node.

The packet loss rate and propagation delay of the access links from senders/receivers to network nodes are set randomly between 10^{-5} and 10^{-1} and between 10^{-1} and 10^{2} [ms], respectively. In the dumbbell and parking lot topologies, the packet loss rate and the propagation delay of the links between network nodes are set to 0.001 and 20 [ms], respectively. For the links between network nodes in the nation-wide topology, the packet loss rate is set to 0.001 and the propagation delay is defined as the geographical distance between each of two network nodes divided by 10^{5} [km/s].

The packet size is set to 1500 [B]. The link bandwidth of the network is not considered, meaning that we do not consider the bandwidth-bottlenecked situation, and packet losses occur randomly regardless of the amount of traffic on each link.

In all three topologies, a data transmission flow is established for all combinations of senders and receivers, except for senders and receivers connected to the same node. We assume that it is a long-lived TCP connection. There are ten thousand flows in the dumbbell topology, three hundred thousand flows in the parking lot topology, and twenty-two million five hundred sixty thousand flows in the nation-wide topology. Network paths between senders and receivers are determined based on Dijkstra's shortest-path algorithm. When a network path of a flow passes through NCC nodes, the connection is divided and data transfer is relayed by multiple sub-connections.

We assume to use NewReno or CUBIC for congestion control algorithms on (sub-)connections. The throughput of the (sub-)connections is calculated by Equations (1) and (2) for NewReno and CUBIC, where RTT is assumed to be twice the one-way propagation delay, T_0 is set to $5 \cdot RTT$, and b is set to one in this evaluation.

The throughput between a sender and a receiver is calculated by the following equation, where N is the number of sub-connections used for the data transmission and ρ_i $(1 \leq i \leq N)$ is the throughput of each sub-connection.

$$\rho_{\mathrm{E2E}} = \min_{1 \leq i \leq N} \rho_i \tag{3}$$

For the performance evaluation metric to reveal the potential performance of NCC, we use the average throughput of all flows in the network. Jain's Fairness Index [21] is also used as the fairness among flows. For each configuration, the evaluation is repeated one hundred times changing the propagation delays and packet loss ratio of the access links.

B. Evaluation in dumbbell topology

Figure 5 plots the relationship between the average throughput and the fairness among flows of the evaluation results with the dumbbell topology in Figure 4 (a). In the graph, we plot the combination of the average throughput and the fairness among flows for all evaluation trials. We change the number of NCC nodes introduced in the network from 0 to 2. For one NCC node, we locate an NCC node to Node 1. We also change the congestion control algorithm used for (sub-)connections. The label "NewReno + CUBIC", NewReno is used on sub-connections on the access links and CUBIC is used on sub-connections between NCC nodes. For other cases, we use the same algorithm for all (sub-)connections in the network.

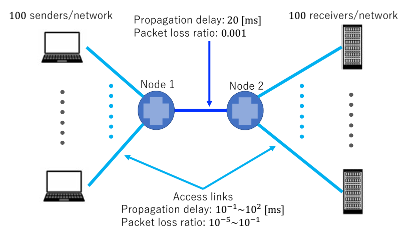
Comparing the result of "0 NCC node, NewReno" and that of "0 NCC node, CUBIC", we find that changing the congestion control algorithm from NewReno to CUBIC slightly improves the average throughput and fairness among flows. On the other hand, the result of "1 NCC node, NewReno" shows that deploying one NCC node is more effective on both throughput and fairness than changing the congestion control algorithm without introducing NCC nodes. The result of "2 NCC nodes, NewReno" shows that deploying two NCC nodes can improve the performance significantly compared with one NCC node case. Furthermore, according to the result of "2 NCC nodes, NewReno + CUBIC", changing the congestion control algorithm between NCC nodes further improves the performance. Overall, comparing "2 NCC nodes, NewReno + CUBIC" and "0 NCC node, NewReno", the average throughput is increased by 2.8 times and the fairness among flows is improved by 1.6 times.

C. Evaluation in parking lot topology

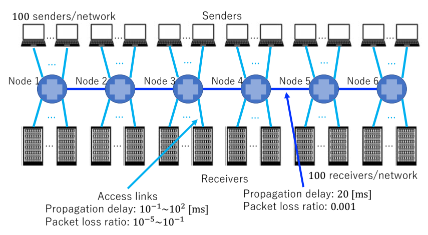
In the parking lot topology of Figure 4 (b), we assess the effect of the number of NCC nodes. Figure 6 shows the relationship between the average throughput and the fairness among flows when CUBIC is used for all (sub-)connections. Multiple plots for the same number of NCC nodes correspond to the results for all possible locations of NCC nodes. Table I shows the locations of NCC nodes to maximize the average throughput or fairness among flows for each number of NCC nodes.

From Figure 6, we can see that the average throughput and the fairness among flows improve as the number of NCC nodes increases. According to Table I, when we increase the number of NCC nodes to improve the average throughput rather than the fairness, NCC nodes should be deployed in order of the number of E2E paths passing through the network nodes, corresponding to the order of the degree of network centrality. On the other hand, to improve the fairness among flows in priority, it is effective to deploy NCC nodes so that as many E2E paths as have at least one NCC node.

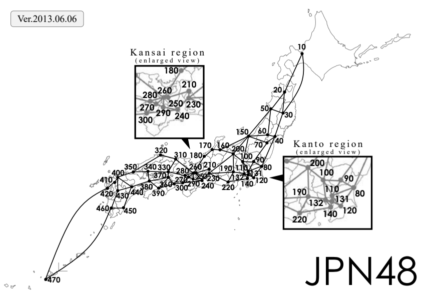
Figure 7 shows the relationship between average throughput and fairness among flows when NewReno is used on all


100 senders/network
Propagation delay: 20 [ms]
Packet loss ratio: 0.001
100 receivers/network
Node 1
Node 2
Access links
Propagation delay: 10^{-1}~10^{2} [ms]
Packet loss ratio: 10^{-5}~10^{-1}

(a) Dumbbell topology



(b) Parking lot topology



(c) Nation-wide topology

Fig. 4: Network topologies for performance evaluation

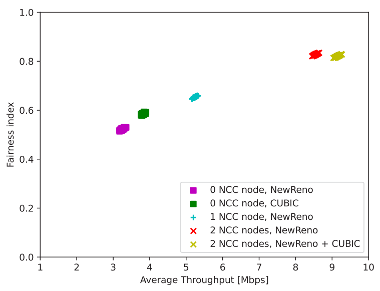


Fig. 5: Evaluation results in dumbbell topology

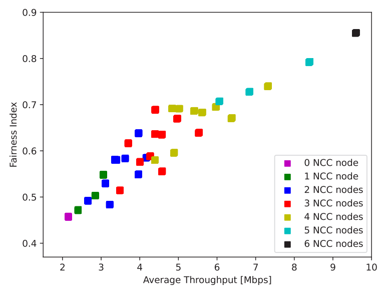


Fig. 6: Evaluation results in parking lot topology: Effect of the number of NCC nodes

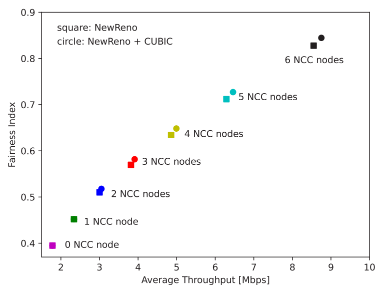


Fig. 7: Evaluation results in parking lot topology: Comparison of changing congestion control algorithms and increasing the number of NCC nodes

TABLE I: Optimal locations of NCC nodes in parking lot topology

number of NCC nodes	to maximize average throughput	to maximize fairness among flows
1	(Node 3), (Node 4)	(Node 3), (Node 4)
2	(Node 3, Node 4)	(Node 2, Node 4), (Node 3, Node 5)
3	(Node 2, Node 3, Node 4), (Node 3, Node 4, Node 5)	(Node 1, Node 3, Node 5), (Node 2, Node 4, Node 6)
4	(Node 2, Node 3, Node 4, Node 5)	(Node 2, Node 3, Node 4, Node 5)
5	(Node 1, Node 2, Node 3, Node 4, Node 5), (Node 2, Node 3, Node 4, Node 5, Node 6)	(Node 1, Node 2, Node 3, Node 4, Node 5), (Node 2, Node 3, Node 4, Node 5, Node 6)

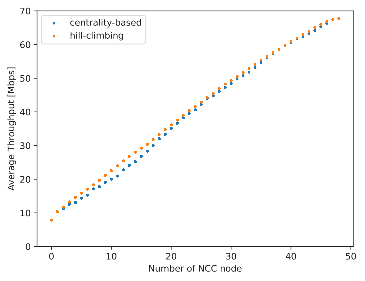


Fig. 8: Evaluation results in nation-wide topology: Effect of incremental deployment and selecting methods of NCC nodes

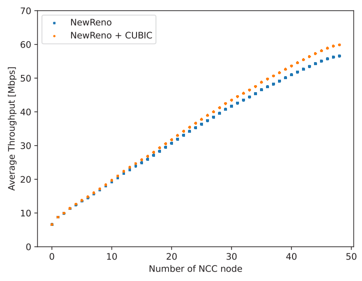


Fig. 9: Evaluation results in nation-wide topology: Comparison between changing congestion control algorithms and increasing the number of NCC nodes

(sub-)connections and that when NewReno is used on (sub-)connections on the access link and CUBIC is used on sub-connections between NCC nodes. This graph plots the averaged results by the number of NCC nodes. Figure 7 indicates that to improve both the average throughput and fairness among flows, it is more effective to increase the number of NCC nodes than to change the congestion control algorithm between NCC nodes to a better one (from NewReno to CUBIC).

D. *Evaluation in nation-wide topology*

We finally evaluate the performance on a realistic large network in Figure 4 (c). We consider the following two methods of selecting network nodes as NCC nodes.

- Centrality-based: Deploys NCC nodes in order of the number of E2E paths passing through the network nodes.
- Hill-climbing: When adding an NCC node, the network node with the maximum average throughput is selected.

Figure 8 shows the relationship between the number of NCC nodes and the average throughput when CUBIC is used as the congestion control algorithm for all (sub-)connections. Figure 8 indicates that as the number of NCC nodes increases, the average throughput improves almost constantly for both methods. This clearly shows the great characteristics of the incremental deployment, meaning that increasing the number of NCC nodes has almost the same effect regardless of the ratio of deployed NCC nodes. This is because as the number of NCC nodes increases, the number of connections that benefit from the addition of NCC nodes decreases, while the throughput of such connections largely increases. On the other hand, comparing the two methods, we can see from Figure 8 that when the number of NCC nodes is small, the average throughput of hill-climbing is up to around 10 % higher than that of centrality-based. Although centrality-based was the optimal way in Section IV-C, it can be seen that there are better selection methods of NCC nodes in the larger-scale and complex networks as in Figure 4 (c).

Figure 9 shows the relationship between the number of NCC nodes and the average throughput when we use NewReno for all (sub-)connections and that when we use CUBIC for sub-connections between NCC nodes and NewReno for (sub-)connections on the access links. While we show only the results with hill-climbing, the results with centrality-based have similar tendencies. Figure 9 indicates that when the ratio of NCC nodes is less than around 50 %, increasing the number of NCC nodes contributes more to improving the average throughput than changing the congestion control algorithm between NCC nodes. On the other hand, when the ratio of NCC nodes is larger than 50 %, changing the congestion control algorithm among NCC nodes is more effective than increasing the number of NCC nodes. This is because the effect of reducing propagation delays by adding NCC nodes can be different depending on the ratio of NCC nodes. In detail, when the ratio of NCC nodes is small, the effect of decreasing propagation delays caused by increasing the number of NCC nodes is large. Thus, increasing the number of NCC nodes is more effective. However, as the ratio of NCC nodes increases, the effect of reducing propagation delays becomes smaller and the effect of changing the congestion control algorithm becomes relatively larger.

V. Summary and Future Work

In this paper, we proposed NCC and explained its advantages and challenges. We then evaluated its potential performance in various network topologies. The evaluation results showed that whether increasing the number of NCC nodes or changing the congestion control algorithm to a better one is more effective depended on the ratio of NCC nodes introduced to the network. Furthermore, we found that how to deploy NCC nodes could make a difference of up to about 10 % in data transfer performance in a realistic network topology.

As a future work, we would like to conduct a detailed performance evaluation of the proposed architecture for technical issues described in Subsection III-C.

Acknowledgement

A part of this research results were obtained from the commissioned research (#00101) by National Institute of Information and Communications Technology (NICT) , JAPAN.

References

[1] S. Ha, I. Rhee, and L. Xu, "CUBIC: a new TCP-friendly high-speed TCP variant," *ACM SIGOPS Operating Systems Review*, vol. 42, pp. 64–74, July 2008.

[2] N. Cardwell, Y. Cheng, C. S. Gunn, S. H. Yeganeh, and V. Jacobson, "BBR: Congestion-based congestion control," *Communications of the ACM*, vol. 60, no. 2, pp. 58–66, February 2017.

[3] X. Nie, Y. Zhao, Z. Li, G. Chen, K. Sui, J. Zhang, Z. Ye, and D. Pei, "Dynamic TCP initial windows and congestion control schemes through reinforcement learning," *IEEE Journal on Selected Areas in Communications*, vol. 37, pp. 1231–1247, June 2019.

[4] K. Ramakrishnan, S. Floyd, and D. Black, "The addition of explicit congestion notification (ECN) to IP," *Request for comments 3168*, September 2001.

[5] D. Katabi, M. J. Handley, and C. Rohrs, "Congestion control for high bandwidth-delay product networks," in *Proceedings of ACM SIGCOM 2002*, August 2002, pp. 89–102.

[6] K. G. Yalda, D. J. Hamad, and N. Ţăpuş, "A survey on software-defined wide area network (SD-WAN) architectures," in *Proceedings of HORA 2022*, June 2022, pp. 1–5.

[7] J. Border, M. Kojo, J. Griner, G. Montenegro, and Z. Shelby, "Performance enhancing proxies intended to mitigate link-related degradations," *Request for comments 3135*, June 2001.

[8] S. Ihm, K. Park, and V. S. Pai, "Wide-area network acceleration for the developing world," in *Proceedings of USENIX ATC 2010*, June 2010.

[9] I. Maki, G. Hasegawa, M. Murata, and T. Murase, "Performance analysis and improvement of TCP proxy mechanism in TCP overlay networks," in *Proceedings of IEEE ICC 2005*, vol. 1, May 2005, pp. 184–190.

[10] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, "Modeling TCP throughput: A simple model and its empirical validation," *SIGCOMM Computer Communication Review*, vol. 28, pp. 303–314, October 1998.

[11] S. Poojary and V. Sharma, "An asymptotic approximation for TCP CUBIC," *Queueing Systems*, vol. 91, pp. 171–203, February 2019.

[12] I. Akyildiz, G. Morabito, and S. Palazzo, "TCP-Peach: A new congestion control scheme for satellite ip networks," *IEEE/ACM Transactions on networking*, vol. 9, pp. 307–321, June 2001.

[13] M. Alizadeh, A. Greenberg, D. A. Maltz, J. Padhye, P. Patel, B. Prabhakar, S. Sengupta, and M. Sridharan, "Data center TCP (DCTCP)," in *Proceedings of ACM SIGCOMM 2010*, August 2010, pp. 63–74.

[14] O. Pop, I. Moldován, C. Simon, J. Bíró, A. Koike, and H. Ishii, "Advertised window-based TCP flow control in routers," in *Proceedings of IFIP SmartNet 2000*, September 2000, pp. 197–218.

[15] V. Tkachov, M. Hunko, and V. Volotka, "Scenarios for implementation of nested virtualization technology in task of improving cloud firewall fault tolerance," in *Proceedings of IEEE PIC S&T 2019*, October 2019, pp. 759–763.

[16] R. Mittal, V. T. Lam, N. Dukkipati, E. Blem, H. Wassel, M. Ghobadi, A. Vahdat, Y. Wang, D. Wetherall, and D. Zats, "TIMELY: RTT-based congestion control for the datacenter," *ACM SIGCOMM Computer Communication Review*, vol. 45, no. 4, pp. 537–550, October 2015.

[17] A. Venkataramani, R. Kokku, and M. Dahlin, "TCP Nice: A mechanism for background transfers," *ACM SIGOPS Operating Systems Review*, vol. 36, pp. 329–343, December 2003.

[18] C. H. Chua and S. C. Ng, "SSL VPN over TCP and UDP Tunnels: Performance evaluation with different server-side congestion control," in *Proceedings of CCIoT 2022*, September 2022, pp. 26–31.

[19] J. Iyengar and M. Thomson, "QUIC: A UDP-based multiplexed and secure transport," *Request for comments 9000*, May 2021.

[20] S. Arakawa, T. Sakano, Y. Tsukishima, H. Hasegawa, T. Tsuritani, Y. Hirota, and H. Tode, "Topological characteristic of Japan photonic network model," *IEICE Technical Report; IEICE Tech. Rep.*, vol. 113, no. 91, pp. 7–12, June 2013.

[21] R. Jain, D. Chiu, and W. Hawe, "A Quantitative measure of fairness and discrimination for resource allocation in Shared Computer Systems," *DEC Research Report TR-301*, September 1984.