IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received May 21, 2021, accepted June 9, 2021, date of publication June 23, 2021, date of current version July 1, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3091707

Motion Control System With Time-Varying Delay Compensation for Access Edge Computing

YUSHI KOYASAKO, TAKAHIRO SUZUKI, SANG-YUEP KIM, (Member, IEEE), JUN-ICHI KANI, (Senior Member, IEEE), AND JUN TERADA

NTT Access Network Service Systems Laboratories, NTT Corporation, Yokosuka-shi 239-0847, Japan

Corresponding author: Yushi Koyasako (yushi.koyasako.xn@hco.ntt.co.jp)

ABSTRACT There is an increasing demand for motion control of Internet of Things devices such as drones and industrial machines from large computational resources on the network side. Since the delay between communication devices adversely affects the control performance, utilization of low-latency platforms such as edge computing is promising. However, no other study focuses on architecture and control methods in optical access network for motion control. We aim at realizing the access edge that performs real-time motion control on an edge server located in a central office. This paper proposes an access edge configuration and a control method that offers time-varying delay compensation based on delay information. We evaluate control performance of our method when delay is increased and packet loss occurs under a high network load environment. We demonstrate that when the downstream traffic changes from 9 to 10 Gbps, the proposed method offers motor control settling times of 2.7 s and 8.0 s, respectively, while the conventional method cannot control the motor in either case.

INDEX TERMS Internet of Things, edge computing, access network, networked control systems, delays, motion control.

I. INTRODUCTION

Internet of Things (IoT) devices, such as drones and industrial machines, are generally controlled from local computational resources. To perform high-load processing using artificial intelligence and big data, there has recently been an increasing demand for control from the cloud, which can provide large computational resources [1], [2]. This has the advantage in that the maintenance of resources can be left to the cloud service, reducing the burden on users. However, the problem with using the cloud is that the distance between the control target and controller is large, which causes a delay and is not suitable for motion control that requires low delay and high reliability. The effect of network delay degrades control performance. As a result, the settling time, which is the time taken to reach within a range of certain percentage of the reference value, becomes longer, or worse, the control system becomes unstable. For example, the latency requirement for factory automation is between 0.25 and 10 ms, while the latency requirement for tactile interaction is 1 ms [3], [4]. To address the delay, a control method has been proposed to link a controller in a factory with the cloud [5], [6], as shown in Figure 1. With this method, the controller of the industrial machine is local, so it can control the target with low latency, and, in conjunction with the cloud, it is possible to achieve high load processing, such as anomaly detection and fault prediction. The feature of this method is that the controller is located at the factory side, so resource maintenance must be done by the user.

With the spread of smartphones, edge computing technology is attracting attention to enhance mobile devices [7], [8]. Edge computing can provide services, such as connected cars, video streaming, augmented or virtual reality, and the tactile internet. It offers sufficient computing performance and shorter response times than cloud computing since the computational resources are placed at the network edge, which is close to the user. The European Telecommunications Standards Institute (ETSI) proposed the Multi-access Edge Computing (MEC) concept [9]. In a MEC system, mobile operators provide computational resources within the radio access network. Studies on integrating edge computing technology and optical networks have been reported [10]; however, these studies mainly focused on the system architecture, task offloading, and resource management, and did not tackle the direct use of edge computing technology for real-time motion control.

The associate editor coordinating the review of this manuscript and approving it for publication was Qunbi Zhuge.

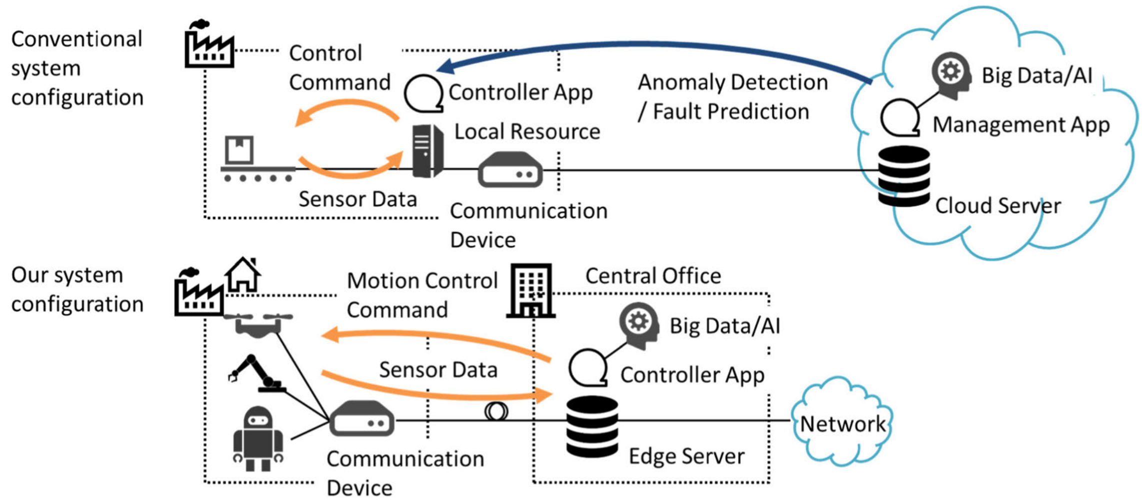

FIGURE 1. **Our system configuration compared with the conventional configuration.**

Various studies on delay compensation for motion control systems have been conducted for core networks, which suffer from high latency and jitter, which is fluctuation in the delay. The core network consists of many switches and other communication devices that the signals must pass through, resulting in overall delays of 100 ms or more [1], [11]–[16]. Several studies use the URLLC, which is a low latency requirement of 5G, but they propose the system models and no study has considered the motion-control method [17]–[19]. However, the control performance required for applications such as factory automation, which have delay requirements of 10 ms or less [3], are difficult to realize. No remote motion-control method has been proposed to realize such applications.

We previously proposed the concept of the access edge, which integrates edge computing into the optical access system for real-time motion control [20]. In the optical access system, delay can be accurately measured by acquiring the dwell time from communication devices, which is expected to yield more detailed control. Figure 1 shows that our system configuration compared with the conventional configuration. With our method, an edge server is located in a central office, which is near the user and equipped with communication devices, and computes the real-time control commands for the user's IoT devices. The delay in the access system is small (10 ms or less [21]), but it is still present and needs to be addressed. Our approach is to apply motion control to low latency access networks, but dealing with real-time jitter remains a challenge. The communication delay between the control device and controller is acquired from communication devices, and the delay is sent to the controller for accurate control by taking into account the delay. Thus, the central office can offer data center services that involve real-time control, allowing these services to be easily connected to advanced data processing applications such as big data and artificial intelligence. We previously confirmed the effectiveness of our method by using a motor and a drone as applications in a non-loaded network environment. The current issue is that the architecture is not clear. In order to realize the access edge, the coordination of motion control and communication functions is essential. Some conventional methods have been proposed to design the local area network parameter for motion control [22], [23], but no study has considered optical access systems. SEBA, a virtualization platform for access systems, can use a common API to manage a variety of access systems, but the coordination of motion control and network management remains a challenge.

This paper extends our conference paper by elaborating the access edge configuration based on virtualization technology for greater flexibility. In the proposed configuration, the SEBA architecture is extended to collect delay information from communication devices and pass it to the controller. Also, we conduct an experiment when delay is increased and packet loss occurs under high network loads, and demonstrate that our method can control the motor accurately even if the conventional method cannot control it. To this extent, the main contributions of this paper are as follows:

1) Proposes a novel configuration for access edge systems that extends the virtualization platform; a new data model is defined to link delay to motion control applications
2) Details an algorithm that receives delay information feedback from communication devices and compensates delay in real time
3) Conducts additional experiments to examine system performance under high network loads and performance analysis using delay acquisition cycle and control cycle.

This paper is organized as follows. In section II, we introduce the proposed access edge configuration. In Section III, we discuss our motion-control method for the access edge. In Section IV, we describe the experiment conducted to evaluate the proposal's control performance and discuss the results. In Section V, we conclude the paper.

II. ACCESS EDGE CONFIGURATION

Delay in the access network is classified into the following components; propagation delay, transmission delay, processing delay, and queuing delay [24]. Propagation delay is the time taken for signal propagation on the physical medium. It is about 5 μs/km, and in the case of a typical optical access system in which the maximum transmission distance is 20 km, the delay is about 0.1 ms. This delay is fixed and difficult to reduce. Transmission delay is the time it takes to encode all the bits of a frame onto the physical line and depends on the frame size and bandwidth of the network. In the case of 1-Gbps optical access, Ethernet frames have a maximum size of 1518 bytes, so the transmission delay is up to 0.012 ms. Processing delay is the time it takes for an application to move a packet from an input interface to an output interface. This delay is on the order of μs for simple applications but increases with complexity. It depends on the number of nodes, protocol, scheduling, etc., and changes with the application algorithm. Queuing delay is the time spent in a buffer at a router or switch and is time-varying depending on the traffic load and congestion rate of the network. There is also retransmission due to buffer overflow, which greatly alters the delay.

Processing and queuing delays vary moment by moment depending on the network condition and are the dominant delay components. In controlling IoT devices, jitter affects control performance as does the size of the delay. It is expected that a large number of IoT devices will be connected to the network, so our research target is to reduce the effect of these delays.

The access edge configuration is assumed that the communication devices of the user and central office are connected in a point-to-point manner. An edge server is located in the central office and connected to the switch (SW) that integrates communication devices such as the fiber media converter (MC).

It will be necessary to change services on the edge server to support actual operation of the access edge. To satisfy the expected service changes, it is effective to adopt a virtualization approach [25]. With the progress in virtualization technology, various technologies, such as Akraino, a software stack for edge computing technology, have been proposed. Virtual machines (VMs) can be launched on the edge server and used to implement the applications that perform motion control.

Recent studies have targeted the flexibility attained by adding and replacing the functions of access network elements as part of the virtualization of the optical access network. The Open Networking Foundation (ONF) has introduced the software-defined network (SDN)-Enabled Broadband Access (SEBA) platform for the Central Office Re-architected as a Datacenter (CORD) concept [26]. This cutting-edge concept allows services to be implemented on general-purpose hardware by the instantiation of SDNs. A more flexible configuration can be achieved by connecting an edge server to general-purpose hardware or applying edge computing technology on general-purpose hardware using the surplus resources of the server. Such a configuration can be achieved by using an open source software (OSS) orchestrator such as Kubernetes, which can manage VMs on several servers. A few operators have started to deploy SEBA, but SEBA is mainly targeted at applications with relaxed latency requirements, such as content delivery networks that save last-mile bandwidth by traffic offloading at the network edge [27], and not at motion control applications with their severe latency requirements.

Figure 2 shows the proposed specific configuration. New components (within red frames) are added to the existing SEBA system (within black frames) to connect the controller on the computing resource to SEBA. Also, the firmware (within blue frames) is newly installed in each communication device. To link the delay information and the motion control application, we extended the SEBA architecture to acquire delay information and calculate total delay; a new data model is defined. This configuration is based on the OSS of the access system, such as Network Edge Mediator (NEM), Open Network Operating System (ONOS), and Virtual OLT Hardware Abstraction (VOLTHA), and an SDN switch [28]–[30]. ONOS provides the control plane for SDN. VOLTHA is an open source project that offers hardware abstraction for broadband access equipment. The existing VOLTHA data model includes an ID to identify the device, a TYPE to indicate the device type, an ADDRESS, and a STATE, etc. The SDN switch (vSW) supports OpenFlow and is used for communication between the switch and the controller. In addition, although we discuss only one Media Converter in this paper, multiple Media Converters and Optical Line Terminals (OLTs) are expected to be connected to the

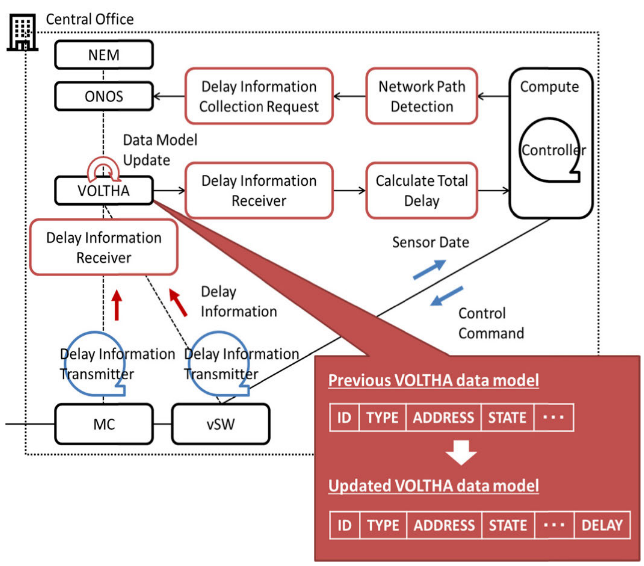

FIGURE 2. Proposed access edge configuration using virtualized optical access network technologies.

edge server. The controller on the computing resource is the user's application that controls the IoT device. We extend the data model of VOLTHA so that it can transfer the delay information of each device.

The processing flow is described below. First, the network path between the controller and the IoT device is detected. Based on this path information, ONOS is instructed to send a request to collect the delay information from each communication device on the path. Each device sends the delay information to the VOLTHA through the delay information transmission function. The delay information stored in VOLTHA is collected and summed to calculate the total delay on the path, which is then passed to the controller. SEBA makes it easy to get accurate identification of the network devices and to obtain delay information.

III. OUR MOTION-CONTROL METHOD

The concept of access edge ensures low latency and high reliability by locating resources in the central office that is physically close to the users and treating them as if they were data centers for real-time motion control. The problem in achieving access edge support is the delay in the access network might degrade control performance, even though both are smaller than what the cloud can achieve.

To reduce the effect of delay in the access network on control performance, our motion-control method adopts time-varying delay compensation based on delay information. Figure 3 a. shows the processing flow of this method. In order to realize the proposal, new parts are added near the network function in each communication device. These additional components are within red frames in the figure. First, as shown in Figure 3 a., (1) the sensor data representing the state of the IoT device are obtained and sent to the controller on the central office server. (2) At the same time, the communication device acquires the timestamps of the start and end times of each network function. (3) The dwell time is taken to be the difference between these times and the delay of this communication device. (4) Next, the delay is sent to the controller at the central office by frame processing. (5) In the controller, the delays of all communication devices are then summed. (6) Finally the controller calculates the delay-compensation control command using sensor data and this total delay. The controller sends the control command to the IoT device, achieving the motion control desired.

We adopt the Smith predictor as the delay-compensation method in the controller, as shown in Figure 3 b. where G_C represents the controller, G_P the IoT device, t_1 and t_2 the downstream and upstream delays, respectively, and $r(t)$, $u(t)$, and $y(t)$ the reference command, control command (calculated by the controller), and sensor command representing the state of the IoT device, respectively. When there is no predictor, the effect of delay is added to $u(t)$ and $y(t)$, and control performance degrades. Note that $R(s)$ is the Laplace transform of $r(t)$, as are the others. The definitions of $x_0(t)$, $x_1(t)$, $x_2(t)$ and $x_3(t)$ are shown in Figure 3 b. Thus, the

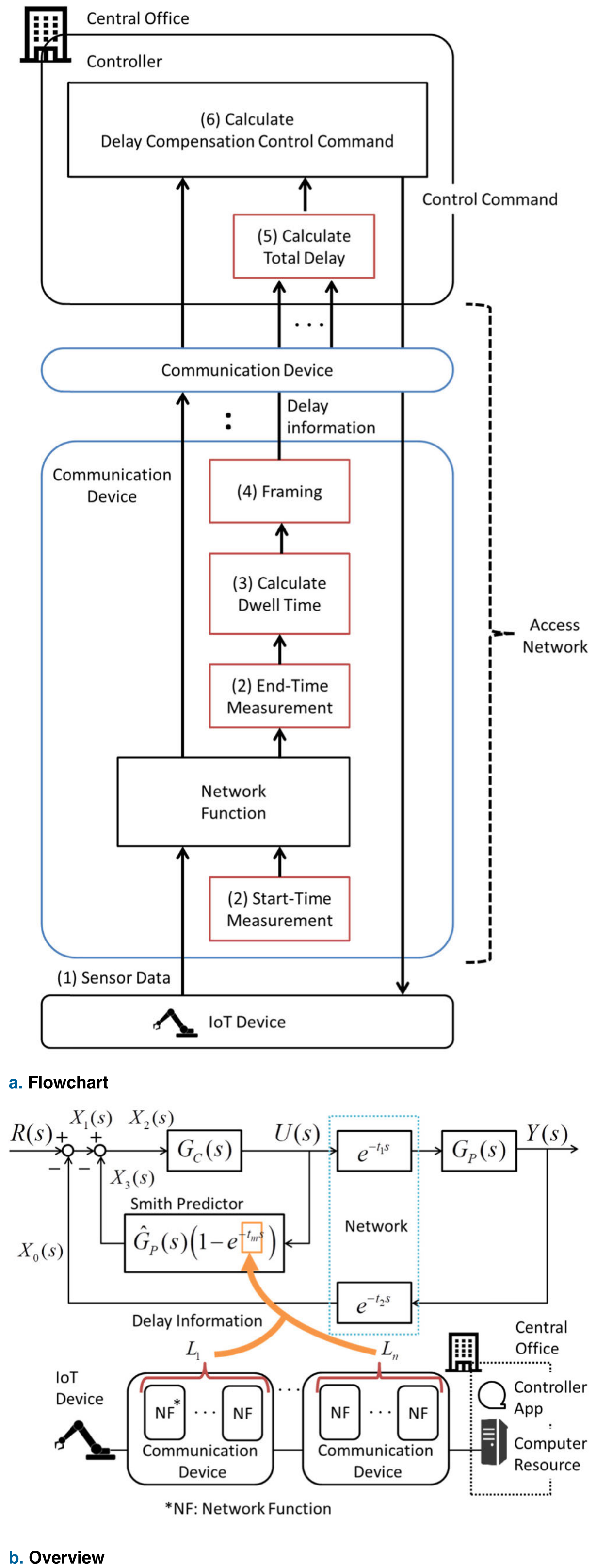

FIGURE 3. Our motion-control method.

transfer function is given as

$$\frac{Y(s)}{R(s)} = \frac{G_C(s)G_P(s)e^{-t_1 s}}{1 + G_C(s)G_P(s)e^{-(t_1+t_2)s}} \quad (1)$$

However, the effect of delay can be reduced by using the Smith predictor. Notation t_m is a design parameter and is the predicted delay. When this value and the actual delay $t_1 + t_2$ are equal, it is possible to form a control system in which the effect of the delay is suppressed. This t_m is calculated by summing the delay information $L_1, .., L_n$ from all network devices. $x_3(t)$ is output by the Smith predictor given t_m. The functional flow of the proposed method is shown in Algorithm 1. Control command $u(t)$ is iteratively calculated by the controller with $x_3(t)$ until $r(t)$-$x_0(t)$ is below the threshold. Our motion-control method replaces the total delay with t_m in real time.

Algorithm 1 Algorithm for the Proposed Method

Input: $r(t), x_0(t), L_1, .., L_n, threshold$
Output: $u(t)$

```
1: while |r(t) − x0(t)| < threshold do
2:   Receive the delay information L1, .., Ln from all the communication devices
3:   tm = Σ_{i=1}^{n} Li
4:   x1(t) = r(t) − x0(t)
5:   x2(t) = x1(t) − x3(t)
6:   u(t) is calculated by the controller with x2(t)
7:   x3(t) is calculated by Smith predictor with tm
8: end while
9: return u(t)
```

Note that our proposed method can actively handle time-varying delay because the network condition is repeatedly sent to the controller by the communication devices. Also, a communication device that has long delay can be identified. By properly managing the communication devices, it is possible to configure a low delay network for real-time motion control.

IV. EXPERIMENT

A. EXPERIMENTAL SETUP

We measured control performance in a high network load environment by using a traffic generator. We also verified the control performance of the proposed method when the delay acquisition cycle and the control cycle were changed. A DC motor experimental system was constructed, as shown in Figure 4. We used two servers to simulate the communication device of the user side and the edge server of the central office side. Each server had an Intel Xeon E5-2699 (2.20 GHz, 22 cores) CPU with 128-GB memory. The operating system was Ubuntu 16.04. The MC was simulated using a 10-Gbps network interface card and Small Form-Factor Pluggable Plus module.

The communication device was a software SW used with Open vSwitch (OVS). This software SW makes it easy to install any desired function into the communication device. OVS was modified to measure the dwell time and send it to the controller. OVS was installed on each server. We used User Datagram Protocol as the transmission protocol because of its real-time capability. A controller was also installed on the central office server.

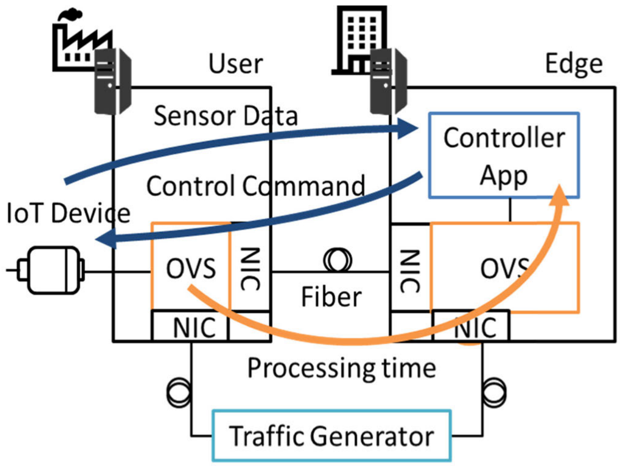

FIGURE 4. Experimental system.

The traffic generator we used was VIAVI MTS-5800. The traffic generator and server were connected with optical fiber. The traffic was set from 1 to 10 Gbps in 1-Gbps steps with two traffic patterns: upstream only and downstream only. To enable 10-Gbps throughput, the Data Plane Development Kit (DPDK) was installed on each server and connected with OVS.

We adopted a classical control approach because it is still widely used in factories. The following three methods were compared.

- Method 1: proportional integral (PI) control without Smith predictor
- Method 2: PI control with Smith predictor setting the delay to the fixed value of 10 ms
- Proposed: PI control with Smith predictor using the measured time repeatedly sent from the communication devices

Note that the motor model used in the Smith predictor was approximated as a second-order system, as shown by the following equation. The numerical value was determined based on the motor's specifications sheet.

$$\hat{G}_P(s) = \frac{195000}{s^2 + 114s + 150} \tag{2}$$

The controller used PI control and the gain was determined by parameter tuning.

$$G_C(s) = K_P E(s) + K_I E(s)/s \tag{3}$$

$$K_P = 0.002 \quad K_I = 0.001 \tag{4}$$

The control cycle was 10 ms, and position control was achieved with a target value of 4000 quad counts, which was one rotation of the motor. The settling time was the time taken to reach within ± 5% of the reference angle.

B. EXPERIMENTAL RESULTS

Table 1 shows the results of delay and frame loss that occurred throughout the entire system. Frame loss was measured

TABLE 1. Delays and frame loss with upstream and downstream traffic.

		Upstream [Gbps]		Downstream [Gbps]		
		1-9	10	1-8	9	10
Delay	Average [ms]	8.4	8.4	8.4	11.4	11.5
	Standard Deviation	0.90	0.90	0.90	1.35	1.45
Packet Loss [frames]		0	19000	0	0	8300

for 30 s. The delays with upstream traffic were almost constant; 8.4 ms. With downstream traffic, from 1 to 8 Gbps, the average delay was almost constant; 8.4 ms, but when traffic was 9 Gbps, the average delay was 11.4 ms, and when traffic was 10 Gbps, the average delay was 11.5 ms, and packet loss occurred. The delay increased with the downstream traffic, so control performance under a high network load environment was measured with downstream traffic.

The control performance with downstream traffic values of 1 to 10 Gbps are shown in Figure 5. Figure 6 shows the time history of method 2 and proposed method at the traffic rate of 9 Gbps. When the traffic was changed from 1 to 8 Gbps, the settling time fluctuated slightly, which may be due to the instantaneous jitter that could not be compensated. When the traffic was changed from 9 to 10 Gbps, methods 1 and 2 failed to handle the resulting delay change, so they could not control the motor. However, the proposed method could handle the delay changes, and it could control the motor. The settling time was 2.7 s at the traffic rate of 9 Gbps, and 8.0 s at 10 Gbps. With 10-Gbps traffic, packet loss also occurred, which may have degraded control performance [31] and increased the settling time. The proposed method could successfully deal with the delay variation, but not the packet loss. If packet loss occurs, delay information and sensor values may not be collected correctly, and control values may not be transmitted correctly, so it is necessary to take a different

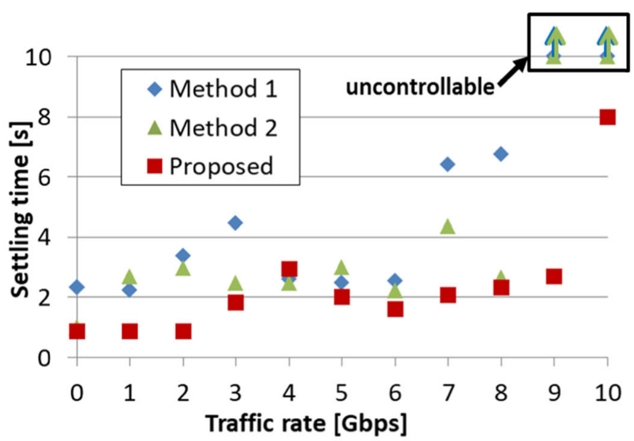

FIGURE 5. Control performance with downstream traffic.

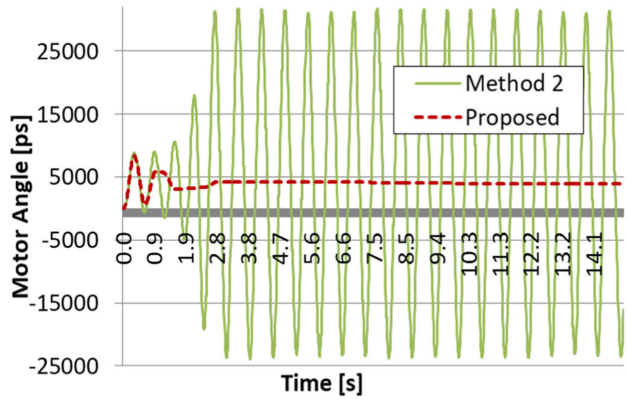

FIGURE 6. Time history at the traffic rate of 9 Gbps.

approach in environments where packet loss is expected. One approach to compensating packet loss is to use a disturbance observer (DOB) in addition to the Smith predictor. DOB can reduce the effect of packet loss by estimating the effect and adding it to the feedback value.

Figure 7 plots control performance versus delay acquisition cycle values at the traffic rate of 9 Gbps. When the delay acquisition cycle was changed with a constant control cycle of 10 ms, we measured the effect on the control performance. When the control cycle and the delay acquisition cycle were equal at 10 ms, the settling time was 2.7 s. However, as the delay acquisition cycle increased, the settling time increased. The control cycle was 10 ms, so the sensor data from the motor was received every 10 ms. If the delay information is not updated when calculating the control command, the old value was used, and it was not possible to handle delay fluctuations in fine detail. The difference between the delay information and the actual delay caused a system model error, and the control performance deteriorated.

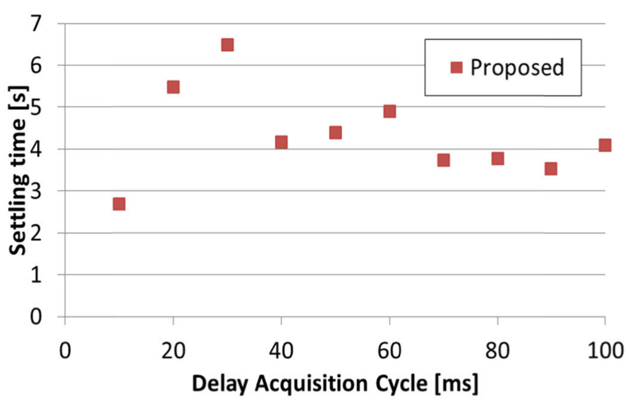

FIGURE 7. Control performance versus delay acquisition cycle values.

Figure 8 shows the results of control performance with control cycle changes in a non-loaded network environment. We measured the effect on the control performance of changing the control cycle while holding the delay acquisition

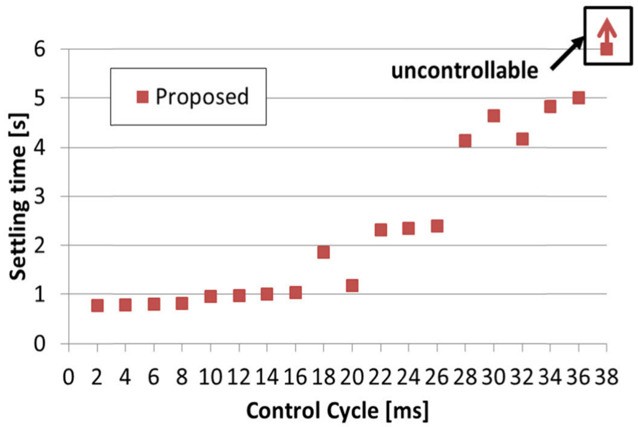

FIGURE 8. Control performance with control cycle changes.

interval was constant, and checked that the control cycle of 10ms was appropriate for the other experiments. There was no significant difference in control performance between the control cycles of 2 ms and 16 ms, but the settling time gradually increased from 18 ms and became uncontrollable at 38 ms. It can be seen that even if the delay information is passed correctly, control cannot be performed when the control cycle is too large. The tighter the control cycle is, the easier it is to deal with the model errors that are triggered by differences between the delay information and the actual delay, making higher control accuracy critical be. In this experiment, it was confirmed that a control cycle of 2 to 16 ms was appropriate.

V. CONCLUSION

We previously proposed the concept of the access edge that performs real-time motion control on an edge server located in a central office. This motion-control method can reduce the effect of delay in the access network on control performance by adopting time-varying delay compensation on the basis of the collection of delay information. In this study, we extend our previous paper by elaborating the access edge configuration based on virtualization technology for greater flexibility. We conduct an experiment to examine high network loads. The conventional motion-control method cannot control the motor at downstream traffic values of 9 or 10 Gbps, whereas our motion-control method can handle the delay and its motor control settling times are within 2.7 s and 8.0 s, respectively. Extension of the proposed method to Passive Optical Network (PON) systems is a likely future work. PON systems can accommodate a large number of devices with large capacity, which is especially beneficial for use in factories. PON systems have some network parameters such as dynamic bandwidth allocation (DBA) cycle that significantly affect the delay, and it is necessary to design a control system that takes these parameters into account. Also, reduction in the delay itself should be considered.

REFERENCES

[1] L. Zhang, H. Gao, and O. Kaynak, "Network-induced constraints in networked control systems—A survey," *IEEE Trans. Ind. Informat.*, vol. 9, no. 1, pp. 403–416, Feb. 2013.

[2] X. Luan, P. Shi, and F. Liu, "Stabilization of networked control systems with random delays," *IEEE Trans. Ind. Electron.*, vol. 58, no. 9, pp. 4323–4330, Sep. 2011.

[3] I. Parvez, A. Rahmati, I. Guvenc, A. I. Sarwat, and H. Dai, "A survey on low latency towards 5G: RAN, core network and caching solutions," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 3098–3130, May 2018.

[4] S. Weibin, L. Yun, D. Yi, D. Yingguo, P. Mingbo, and X. Gang, "Three-real-time architecture of industrial automation based on edge computing," in *Proc. IEEE Int. Conf. Smart Internet Things (SmartIoT)*, Aug. 2019, pp. 372–377.

[5] *AWS IoT Greengrass—Amazon Web Services*. Accessed: Nov. 1, 2020. [Online]. Available: https://aws.amazon.com/jp/greengrass/

[6] *IoT Edge | Microsoft Azure*. Accessed: Nov. 1, 2020. [Online]. Available: https://azure.microsoft.com/en-us/services/iot-edge/

[7] M. T. Beck, M. Werner, S. Feld, and T. Schimper, "Mobile edge computing: A taxonomy," in *Proc. 6th Int. Conf. Adv. Future Internet*, 2014, pp. 48–55.

[8] H. Tanaka, M. Yoshida, K. Mori, and N. Takahashi, "Multi-access edge computing: A survey," *J. Inf. Process.*, vol. 26, pp. 87–97, Feb. 2018.

[9] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2322–2358, Aug. 2017.

[10] N. Yoshikane, "Applications of SDN-enabled optical transport network and cloud/edge computing technology," in *Proc. Opt. Fiber Commun. Conf. Exhib. (OFC)*, Mar. 2019, pp. 1–3.

[11] X.-M. Zhang, Q.-L. Han, and X. Yu, "Survey on recent advances in networked control systems," *IEEE Trans. Ind. Informat.*, vol. 12, no. 5, pp. 1740–1752, Oct. 2016.

[12] C.-L. Lai and P.-L. Hsu, "Design the remote control system with the time-delay estimator and the adaptive smith predictor," *IEEE Trans. Ind. Informat.*, vol. 6, no. 1, pp. 73–80, Feb. 2010.

[13] D. Yashiro and T. Yakoh, "Feedback controller with low-pass-filter-based delay regulation for networked control systems," *IEEE Trans. Ind. Electron.*, vol. 61, no. 7, pp. 3744–3752, Jul. 2014.

[14] K. Natori and K. Ohnishi, "A design method of communication disturbance observer for time-delay compensation, taking the dynamic property of network disturbance into account," *IEEE Trans. Ind. Electron.*, vol. 55, no. 5, pp. 2152–2168, May 2008.

[15] H.-C. Yi, C.-J. An, and J.-Y. Choi, "Compensation of time-varying delay in networked control system over Wi-Fi network," *Int. J. Comput. Commun. Control*, vol. 12, no. 3, pp. 415–428, Jun. 2017.

[16] M. Ahmadi, N. Khanezaei, S. Manavi, F. F. Moghaddam, and T. Khodadadi, "A comparative study of time management and energy consumption in mobile cloud computing," in *Proc. IEEE 5th Control Syst. Graduate Res. Colloq.*, Aug. 2014, pp. 199–203.

[17] E. Markoval, D. Moltchanov, R. Pirmagomedov, D. Ivanova, Y. Koucheryavy, and K. Samouylov, "Priority-based coexistence of eMBB and URLLC traffic in industrial 5G NR deployments," in *Proc. 12th Int. Congr. Ultra Modern Telecommun. Control Syst. Workshops (ICUMT)*, Oct. 2020, pp. 1–6.

[18] C. Stefanovic, "Industry 4.0 from 5G perspective: Use-cases, requirements, challenges and approaches," in *Proc. 11th CMI Int. Conf., Prospects Challenges Towards Developing Digit. Economy EU*, Nov. 2018, pp. 44–48.

[19] H. Yang, K. Zhang, K. Zheng, and Y. Qian, "Leveraging linear quadratic regulator cost and energy consumption for ultrareliable and low-latency IoT control systems," *IEEE Internet Things J.*, vol. 7, no. 9, pp. 8356–8371, Sep. 2020.

[20] Y. Koyasako, T. Suzuki, S. Kim, J. Kani, and J. Terada, "Real-time motion control method using measured delay information on access edge computing," in *Proc. Annu. Consum. Commun. Netw. Conf. (CCNC)*, Jan. 2020, pp. 1–4.

[21] N. Cvijetic, "Optical network evolution for 5G mobile applications and SDN-based control," in *Proc. 16th Int. Telecommun. Netw. Strategy Planning Symp. (Networks)*, Sep. 2014, pp. 1–5.

[22] K. Huang, W. Liu, Y. Li, A. Savkin, and B. Vucetic, "Wireless feedback control with variable packet length for industrial IoT," *IEEE Wireless Commun. Lett.*, vol. 9, no. 9, pp. 1586–1590, Sep. 2020.

[23] Y. Maddahi, S. Liao, W.-K. Fung, E. Hossain, and N. Sepehri, "Selection of network parameters in wireless control of bilateral teleoperated manipulators," *IEEE Trans. Ind. Informat.*, vol. 11, no. 6, pp. 1445–1456, Dec. 2015.

[24] I. Grigorik and Y. Wada, *High Performance Browser Networking*. Sebastopol, CA, USA: O'Reilly Media, 2014.

[25] A. C. Baktir, A. Ozgovde, and C. Ersoy, "How can edge computing benefit from software-defined networking: A survey, use cases, and future directions," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2359–2391, Jun. 2017.

[26] L. Peterson, A. Al-Shabibi, T. Anshutz, S. Baker, A. Bavier, S. Das, J. Hart, G. Palukar, and W. Snow, "Central office re-architected as a data center," *IEEE Commun. Mag.*, vol. 54, no. 10, pp. 96–101, Oct. 2016.

[27] *VOLTHA/SEBA Status in Telefónica's ONLIFE*, SEBA/VOLTHA Community Virtual Meeting, Menlo Park, CA, USA, 2020.

[28] *XOS and NEM—OpenCORD Wiki*. Accessed: Feb. 18, 2021. [Online]. Available: https://wiki.opencord.org/display/CORD/XOS+and+NEM

[29] *ONOS—Wiki*. Accessed: Feb. 18, 2021. [Online]. Available: https://wiki.onosproject.org/display/ONOS/ONOS

[30] *VOLTHA Docs*. Accessed: Feb. 18, 2021. [Online]. Available: https://docs.voltha.org/master/index.html

[31] R. Imai and R. Kubo, "Experimental validation of communication disturbance observer for networked control systems with information losses," *IEICE Commun. Exp.*, vol. 5, no. 4, pp. 102–107, 2016.

YUSHI KOYASAKO received the B.E. and M.E. degrees from Keio University, Kanagawa, Japan, in 2015 and 2017, respectively. He joined NTT Access Network Service Systems Laboratories, Yokosuka, Japan, in 2017, where he has been engaged in research on access system virtualization.

TAKAHIRO SUZUKI received the B.E., M.E., and Ph.D. degrees in engineering from Waseda University, Tokyo, Japan, in 2012, 2014, and 2017, respectively. He joined NTT Access Network Service Systems Laboratories, NTT Corporation, Kanagawa, Japan, in 2014. His research interests include signal processing and real-time implementation for optical access and image/video systems. He is a member of the Institute of Electronics, Information and Communication Engineers (IEICE) and serving as an IEICE Technical Committee Member on Smart Info-Media Systems (SIS).

SANG-YUEP KIM (Member, IEEE) received the Ph.D. degree in electronics engineering from Kwangwoon University, Seoul, South Korea, in 2004. From 2004 to 2007, he was with The University of Tokyo, Japan, under a Postdoctoral Foreign Researcher Fellowship. He joined NTT Access Network Service Systems Laboratories, NTT Corporation, Chiba, Japan, in 2008. He is currently researching DSP technologies for future optical access systems.

JUN-ICHI KANI (Senior Member, IEEE) received the B.E., M.E., and Ph.D. degrees in applied physics from Waseda University, Tokyo, Japan, in 1994, 1996, and 2005, respectively. He joined NTT Optical Network Systems Laboratories, in 1996, where he engaged in researching optical multiplexing and transmission technologies. Since 2003, he has been with NTT Access Network Service Systems Laboratories, where he is engaged in the research and development (R&D) of optical communication systems for metro and access applications and currently heads the Access Systems Technology Group. He has been participating in ITU-T and the Full Service Access Network initiative (FSAN), since 2003.

JUN TERADA received the B.E. degree in science and engineering and the M.E. degree in computer science from Keio University, Kanagawa, Japan, in 1993 and 1995, respectively. He joined NTT LSI Laboratories, in 1995, where he engaged in the research and development of low-voltage analog circuits, especially A/D and D/A converters. Since 1999, he has been engaged in developing small and low-power wireless systems for sensor networks. Since 2006, he has been engaged in high-speed front-end circuits for optical transceivers. He is currently a Senior Research Engineer and a Supervisor with NTT Access Network Service Systems Laboratories, where he is responsible for research and development management of optical access networks including fixed-wireless convergence and virtualization technology. He is a Senior Member of IEICE of Japan. He is serving as a Technical Committee Member on Asian Solid-State Circuits Conference (A-SSCC) as well as the Vice Chair of the IEICE Technical Committee on Communication Systems (CS).