IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received November 12, 2021, accepted December 5, 2021, date of publication December 20, 2021,
date of current version January 4, 2022.

Digital Object Identifier 10.1109/ACCESS.2021.3136876

Demonstration of Real-Time Motion Control Method for Access Edge Computing in PONs

YUSHI KOYASAKO, TAKAHIRO SUZUKI, TAKASHI YAMADA, TATSUYA SHIMADA, AND TOMOAKI YOSHIDA, (Member, IEEE)
NTT Access Network Service Systems Laboratories, NTT Corporation, Yokosuka-shi, Kanagawa 239-0847, Japan
Corresponding author: Yushi Koyasako (yushi.koyasako.xn@hco.ntt.co.jp)

ABSTRACT In the remote motion control of IoT devices, the delay adversely affects control performance. In our access edge concept, in which motion control is performed from a central office, the delay becomes large due to Dynamic Bandwidth Allocation (DBA) when the Passive Optical Network (PON) is adopted as the access system. No existing motion control method considers the delay changes of the DBA algorithm. We rectify this omission by proposing a novel real-time and high-precision motion control method that links the PON device and the controller. The proposed method can improve the control performance while maintaining the bandwidth utilization efficiency without changing the DBA delay specified to meet the requirements. In the proposed method, the delay in the PON system is calculated by using a delay model based on the setting parameter received from the communication device, and is used for delay compensation control. An experiment shows that, unlike the conventional method which becomes unstable and fails to realize control, the proposed method achieves high resolution control within 4 s, which demonstrates the feasibility of motion control for factory automation using edge computing in PON systems.

INDEX TERMS Internet of Things, edge computing, access network, networked control systems, delays, motion control, passive optical network, dynamic bandwidth allocation.

I. INTRODUCTION

The demand for Internet of Things (IoT) devices such as drones and industrial robots continues to expand. So far, motion control of these devices is mostly performed by computation resources located near or on the IoT devices. In recent years, in order to meet the increasing demand for applications that require huge computation resources such as Artificial Intelligence (AI) and big data, and to reduce the system maintenance cost for users, remote motion control of IoT devices by resources in the cloud is being adopted [1], [2]. However, since the distance between the cloud and the device can be significant, control delay between them is inevitable, a barrier to real-time and high-precision motion control. Inaccurate delay estimation degrades control performance, and as a result, the settling time, which is the time taken to reach within a certain percentage range of the reference value, becomes longer, or worse, the control system becomes unstable.

In realizing applications that require low latency, edge computing is attracting attention [3], [4]. Optical access systems are suitable for edge computing because of the infrastructure is closest to users [5]. There are two major types of access topologies: Point-to-Point topology and Point-to-Multipoint topology. Passive Optical Network (PON) systems are generally used for the Point-to-Multipoint topology. PON systems distribute a single wavelength among multiple users, and have many advantages for control of the IoT devices in factories, such as connectivity of massive IoT devices, high capacity transmission, low cost, electromagnetic resistance, and flexible bandwidth allocation. However, unlike Point-to-Point systems, latency fluctuations and bandwidth limitations can be significant. The latency requirement for factory automation is between 0.25 and 10 ms [6], [7]. PON systems need an access control protocol to prevent data collision, and currently the Dynamic Bandwidth Allocation (DBA) algorithm based on the TDMA protocol is used to dynamically distribute bandwidth according to service requests and Quality of Service (QoS) requirements. The DBA algorithm can create serious delay fluctuations, which degrades control performance.

The associate editor coordinating the review of this manuscript and approving it for publication was Chao-Yang Chen.

A variety of algorithms have been proposed to reduce the latency of DBA. In references [8]–[10], there is generally a trade-off between low latency and bandwidth utilization efficiency in DBA because it is necessary to collect bandwidth requests from each user and calculate bandwidth allocation. Also, the reference [11] method assumes the use of mobile devices, so it cannot be used for IoT devices in factories. On the other hand, from the viewpoint of IoT devices control, more detailed control may be possible if the end-to-end delay information including the PON section can be accurately obtained, instead of reducing the delay of DBA itself.

Several previous studies have used PONs in edge computing [12]–[14], but few studies have examined their support of real-time motion control. Reference [15] describes the maximum permissible number of branches in a direct connection between a factory control center and a factory machine using a PON with ultra-high branching numbers. That study, however, calculated the maximum number of PON branches for factories based on the optical power loss, and did not evaluate the control performance of industrial machines. Reference [16] demonstrated a low-latency application for motor control using a long-reach 10G-EPON for optical access edge computing. However, that study did not propose a control system that took account of the DBA delay changes. No paper to date has proposed a motion control system that considers the delay changes created by the DBA algorithm.

We previously proposed the access edge concept for real-time motion control of the IoT devices by installing computation resources in the central office or micro data center near the user's site and implementing edge computing technology [17], [18]. Since the central office is located closest in the network to the user, the effect of delay is basically small, and high-precision motion control of IoT devices can be achieved. By treating the central office like a data center, it is possible to perform control in cooperation with complex applications that require the integrated control of many resources. The user can leave the system maintenance cost to the carrier, and the carrier can effectively utilize the surplus resources of the central office. However, our previous research focused on Point-to-Point systems. Our access edge concept as applied in PONs is shown in Figure 1. The IoT devices and the edge server can be directly connected via a PON, while the factory and the central office are connected in a point to point manner; the cascade connections offered by the PON are used to support a very large number of IoT devices. To realize the access edge with PON, it is necessary to deal with DBA delay changes.

We propose a novel real-time and high-precision control method that uses a PON delay model to compensate for PON-originated delay in order to achieve PON-based motion control that considers DBA delay changes. Since the delay generated by DBA is dominant in PONs, detailed control can be expected when the delay of the DBA is properly modelled. No change is made to the delay caused by the DBA algorithm determined according to the requirements, and the control performance can be improved while maintaining bandwidth utilization efficiency. The proposed method estimates a delay value that depends on the PON setting parameter by extracting it from the communication device, and realizes active delay compensation by using this delay value. An experiment on an XGS-PON verifies that the proposed method is able to achieve control even when the conventional method cannot.

This paper is organized as follows. In Section II, we describe the DBA algorithm of the PON system. In Section III, we briefly review the conventional motion control method. In Section IV we introduce our motion-control method for access edge computing. In Section V, we describe the experiment conducted to evaluate its control performance and discuss the results. In Section VI we conclude the paper.

II. DBA ALGORITHM

A PON system provides high-speed data communication to multiple users economically by connecting one Optical Line Terminal (OLT) with multiple Optical Network Units (ONUs). Time division multiplexing PON (TDM-PON) is commonly used, and in TDM-PON, the OLT assigns the transmission timing of ONUs to avoid collision of uplink signals sent from multiple ONUs.

Figure 2 shows the DBA mechanism of the PON system [19]. The ONU sends a Request message to the OLT after data arrives, and the DBA function of OLT calculates access rights based on Request messages received from each ONU and sends a Grant message which contains upstream bandwidth allocated to the ONU. The ONU sends data to the OLT based on the received Grant. The delay until the data

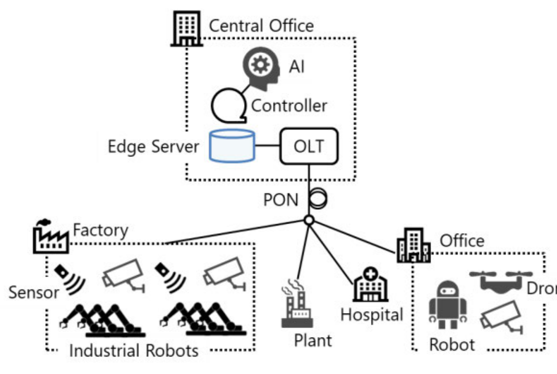


FIGURE 1. Access edge concept for PONs.

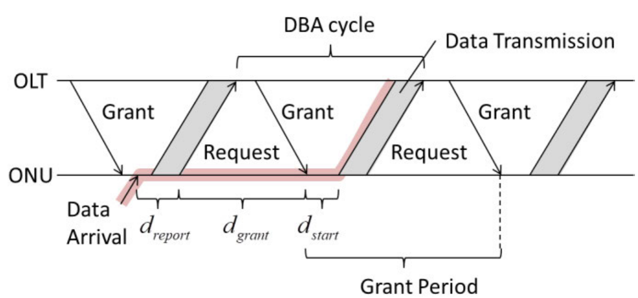


FIGURE 2. DBA mechanism.

received by the ONU arrives at the OLT varies in large part due to the DBA cycle. In this case, the Grant period is, from the viewpoint of an ONU, the same as the DBA cycle.

This method requires a Request message be sent and a Grant message be received before sending the uplink data, so while the bandwidth utilization efficiency is high, the delay at the start of transmission tends to be large.

DBA delay d_{DBA} is given by the following equation:

$$d_{DBA} = d_{report} + d_{grant} + d_{start} \quad (1)$$

d_{report} is the time from when the data arrives at the ONU to when a Request message is sent, d_{grant} is the time from when a Grant message is received, and d_{start} is the time from when the data is sent to the OLT. Note that the right to send a Request message is granted to all connected ONUs in each bandwidth allocation cycle.

III. CONVENTIONAL MOTION CONTROL METHOD

There have been various studies on compensating the delay of communication systems [1], [20]–[25]. The Smith predictor is a typical delay compensation control method. In Figure 3, G_C represents the controller, G_P represents the control target, and, t_1, t_2 represent, respectively, the downstream and upstream delays. $r(t)$, $u(t)$, and $y(t)$ represent the reference command, the control command calculated by the controller, and the sensor command representing the state of the control target like a IoT device, respectively. For network-based control, the delay impacts $u(t)$ and $y(t)$, which degrades control performance. Note that $R(s)$ is the Laplace transform of $r(t)$,

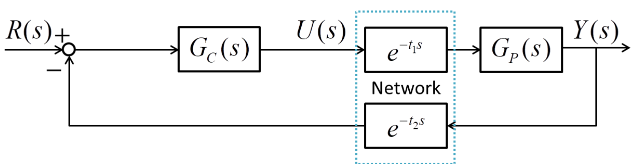


FIGURE 3. Network-based control system.

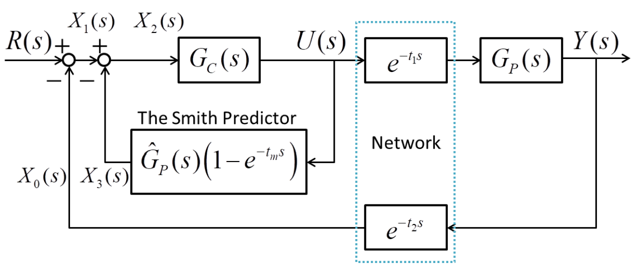


FIGURE 4. System with smith predictor.

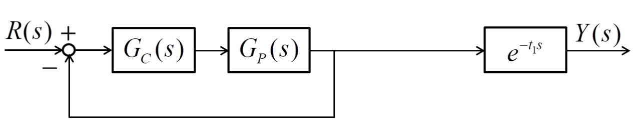


FIGURE 5. Network-based control system.

as are the others. The transfer function is given as:

$$\frac{Y(s)}{R(s)} = \frac{G_C(s)G_P(s)e^{-t_1 s}}{1 + G_C(s)G_P(s)e^{-(t_1+t_2)s}} \quad (2)$$

Figure 4 shows the control system with Smith predictor $P(s)$, which is expressed as follows:

$$P(s) = \hat{G}_P(s)\left(1 - e^{-t_m s}\right) \quad (3)$$

$\hat{G}_P(s)$ is the nominal model, and t_m is the predicted delay. When the predicted delay value t_m equals the actual delay value $t_1 + t_2$, the control system can exactly offset the effect of the delay, as shown in Figure 5.

IV. PROPOSED MOTION CONTROL METHOD

To deal with DBA delay changes, we propose a method that achieves more detailed control. In the proposed method, the delay information is extracted from the communication device experiencing the delay and passed to the controller. The controller maintains a delay model whose input is the setting parameter of the communication device. The model predicts the delay generated between the controller and the control target, and uses it for delay compensation. Though the setting parameter is influenced by the number of IoT devices and the amount of traffic, this study proposes a method to achieve high-precision control even when the PON setting parameter is changed. This study uses XGS-PON [26].

In XGS-PON, the setting parameter of the Max Grant Period P_{MAX}, which is the maximum amount of time between consecutive Grant messages from OLT to ONU, alters the delay generated in the PON system. The maximum upstream delay is determined as follows.

$$t_{2MAX} = P_{MAX} + 2 \cdot \frac{T \cdot P_{MAX}}{L} \quad (4)$$

The second term is the transmission delay. TrafficRate T is upstream traffic rate. LineRate L is the maximum amount of network traffic. The average delay is expected to be half this value.

$$t_m = t_1 + \frac{t_{2MAX}}{2} \quad (5)$$

At this time, the downstream delay t_1 is sufficiently smaller than the upstream delay t_2, and is fixed regardless of the DBA cycle, so it can be ignored, or determined by measurements.

In the PON system, the delay is not only the DBA delay but also the queuing delay caused by traffic demand. When the traffic demand exceeds the traffic rate, the excess traffic is stored in transmission buffers, causing some queuing delay before transmission. Since the amount of traffic is random, this behavior also occurs randomly, making it difficult to predict.

Since the control traffic such as sensor data and control commands are usually text data, the throughput required to control one IoT device is usually a few hundred kbps or less. In fact, the current industrial Ethernet protocol used in most factories is capable of 100 Mbps throughput [27]. In this paper, we consider the case where the control traffic of all devices does not exceed 10 Gbps and queuing delay does not occur.

However, in an actual network, traffic other than the control traffic, such as video data for monitoring must be expected to flow along with control traffic. If the total traffic exceeds 10 Gbps, queuing delay occurs and control traffic will be affected. To solve this problem, existing industrial protocols assign a different priority to real-time traffic, such as data required for motion-control, than to non-real-time traffic, to avoid queuing delay [28]. In this study, we use this method to avoid the queuing delay instead of predicting it by setting the traffic used for real-time motion control to a higher priority and differentiating it from other traffic such as video data. Since the data size of control data is usually not large, this solution is expected to be effective.

We describe the processing procedure in Figure 6 and Algorithm 1. (1) First, the PON setting parameters (MaxGrantPeriod, TrafficRate, LineRate) are obtained from OLT and passed to the PON latency model. (2) The model uses these values to calculate the upstream delay value t_{2MAX}, and then the total delay value t_m is determined by summing t_2 with the downstream delay value t_1. This value is then passed to the controller that calculates the control value. (3) Next, the controller receives the sensor data $x_0(t)$ sent from the IoT device. (4) The controller calculates the delay compensation control command $u(t)$ using sensor data $x_0(t)$ and delay value t_m. Here we assume the Smith predictor is used as the delay compensation method. (5) The controller sets the priority of this data. (6) Finally, the controller sends the control command $u(t)$ to the IoT device. The delay is received only once at the beginning, while (3)-(6) are repeated to achieve control.

V. EXPERIMENT

A. EXPERIMENTAL SETUP

In order to confirm the advantages of the proposed method, a basic motor-based experimental system was constructed, see Figures 7 and 8. We used two servers to simulate the user side and the central office side. The CPU of each server was an Intel Xeon E3-1270 (3.80 GHz, 4 core) CPU with 32 GB of memory. The operating system was Ubuntu 16.04. OLT

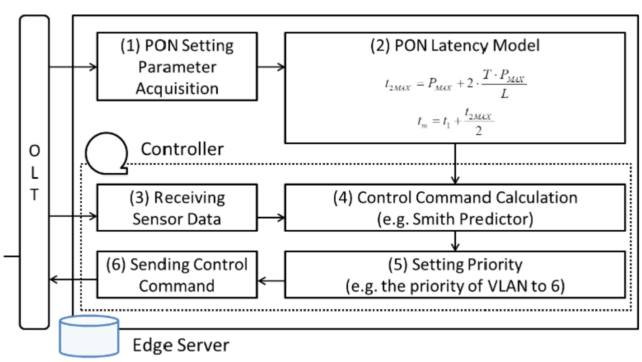


FIGURE 6. Architecture of the proposed method.

Algorithm 1 Algorithm for the Proposed Method

Input: $r(t)$, $x_0(t)$, P_{MAX}, T, L, *threshold*
Output: $u(t)$
$t_{2MAX} = P_{MAX} + 2 \cdot \frac{T.P_{MAX}}{L}$
$t_m = t_1 + \frac{t_{2MAX}}{2}$
while $|r(t) - x_0(t)| < threshold$ **do**
  $x_1(t) = r(t) - x_0(t)$
  $x_2(t) = x_1(t) - x_3(t)$
  $u(t)$ is calculated by the controller with $x_2(t)$
  $x_3(t)$ is calculated by Smith predictor with t_m
end while
return $u(t)$

was a Tibit Micro Plug OLT, and ONU was a T&W ONU. Traffic generator was a VIAVI MTS-5800. Traffic generator and server were connected by optical fiber. In this research, the switch function was realized by Open vSwitch (OVS), a software switch. A program controller was installed on the central office server. The motor was a Maxon EC-max 40. The motor and the user server were connected by a USB cable and signal cable via EPOS2 24/5. The calculated control command was converted into a current value to realize current control.

The following three methods were compared.

- Method 1: PI control with Smith predictor setting the predicted delay t_m to the fixed value of 10 ms
- Method 2: PI control with Smith predictor setting the Max Grant Period in the PON delay model to the fixed value of 20 ms
- Proposed method: PI control with Smith predictor using the PON delay model output

Method 1 was set the end-to-end delay including the PON section to a fixed value of 10 ms, and Method 2 was set the value of the Max Grant Period to the maximum value.

We conducted the following two experiments.

- Experiment 1: Unloaded network environment
- Experiment 2: 40 Mbps load environment using generated traffic

The motor characteristics are given by the following equation. The numerical value was decided based on the motor's specification sheet.

$$\hat{G}_P(s) = \frac{195000}{s^2 + 114s + 150} \tag{6}$$

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

$$G_C(s) = K_P E(s) + K_I E(s)/s \tag{7}$$
$$K_P = 0.001 \quad K_I = 0.0016 \tag{8}$$

The control cycle was 2 ms, and position control was performed with a target value of 4000 quad counts. The upstream traffic was 40 Mbps. The settling time was the time taken to achieve ± 5% of the reference angle.

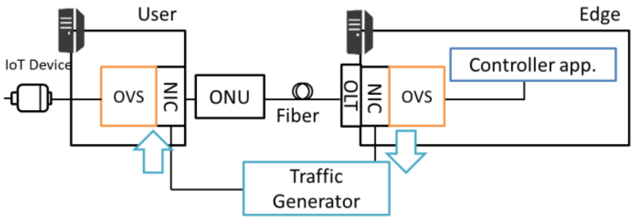


FIGURE 7. Experimental system.

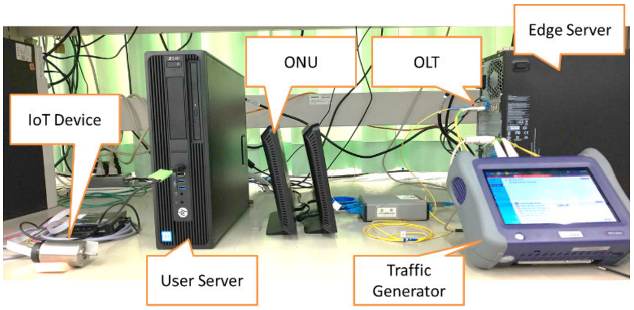


FIGURE 8. Demonstration system.

B. EXPERIMENTAL RESULTS

1) EXPERIMENT 1

In order to evaluate the effect of the delay changes due to the change in the Max Grant Period on the control performance, we measured the settling time in an unloaded network environment. The proposed method is compared to Method 1 and Method 2 in Figures 9 and 10, respectively. Each result is from 4 trials and the standard error is shown. As shown in Figure 9, there was no significant difference between Method 1 and the proposed method for Max Grant Periods under 12 ms. However, when it exceeded 12 ms, the settling time became longer with Method 1. This was considered to be caused by its inability to deal with the difference between the expected delay value and the actual delay value. On the other hand, the proposed method could maintain the same control performance regardless of the Max Grant Period because the delay model was able to deal with changes in delay value.

As shown in Figure 10, in Method 2, since the delay model was set to the worst case value, it could be seen that control succeeded at all Max Grant Periods. However, when the Max Grant Period was 1 ms, there was a difference in settling time of up to about 6 ms with the proposed method because it did not correspond to the actual delay value. As the difference between the set delay value and the actual delay value decreased, the difference in the settling time with the proposed method decreased, but more detailed control could be achieved by designing the delay model to match the actual delay. These experiments confirmed that the proposed method could flexibly deal with DBA cycle changes and realized detailed control.

Then, in order to confirm the effectiveness of the proposed method regardless of the control parameters, the control results of each method are investigated when the gain is changed. As shown in Figure 11, the settling times of

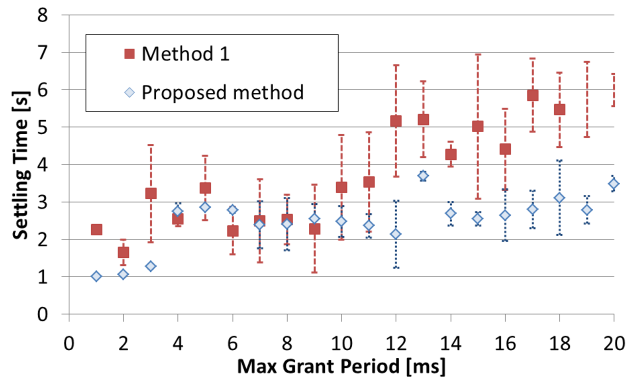


FIGURE 9. Comparison of Method 1 and Proposed method.

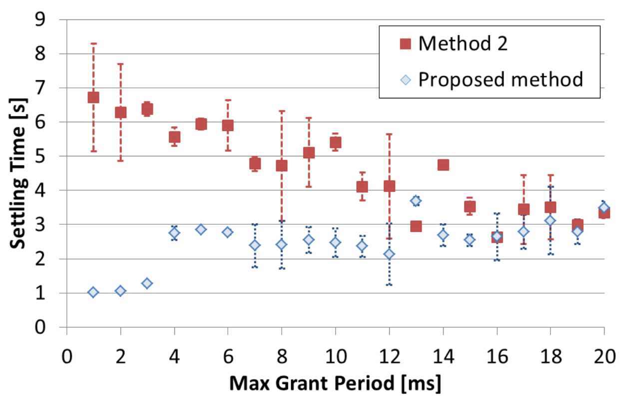


FIGURE 10. Comparison of Method 2 and proposed method.

Method 1 and the proposed method are compared when K_P gain is fixed and K_I gain is varied from 0.002 to 0.024 in intervals of 0.002. In general, when K_I gain is small, the effect of correcting the error from the reference command becomes weak, while when K_I gain is too large, the method is sensitive to the error and exhibits oscillatory behavior. In the conventional method, the settling time was 23 s when K_I gain is 0.002, and it diverges and does not converge when K_I gain is 0.024. On the other hand, the proposed method takes the delay into account, and it is confirmed that the proposed method can control the system faster than the conventional method for any selected K_I gain.

In these experiments, the average processing time from the time the controller application received the sensor data to the time it sent the control command was 69 μs. This value is 1/28$^{\text{th}}$ of the control cycle of 2 ms in these experiments, so the proposed algorithm is expected to guarantee real-time performance even on a cheaper verification platform with 28x slower performance.

2) EXPERIMENT 2

In order to verify the effectiveness of the proposed method in a high-loaded network environment, we measured the control performance and evaluated the difference between the output of the PON delay model used in the proposed method and the

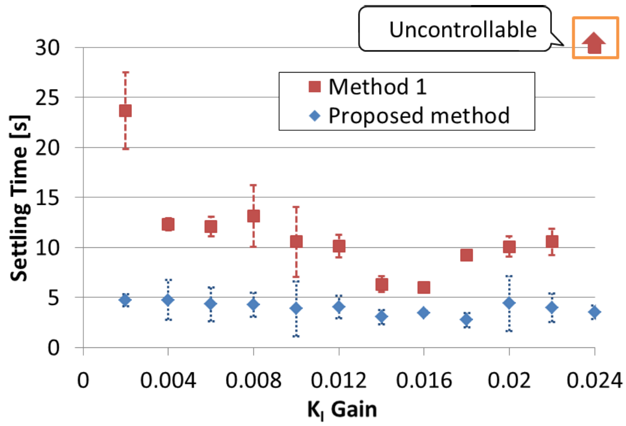


FIGURE 11. Comparison of Method 1 and proposed method when Ki gain is changed.

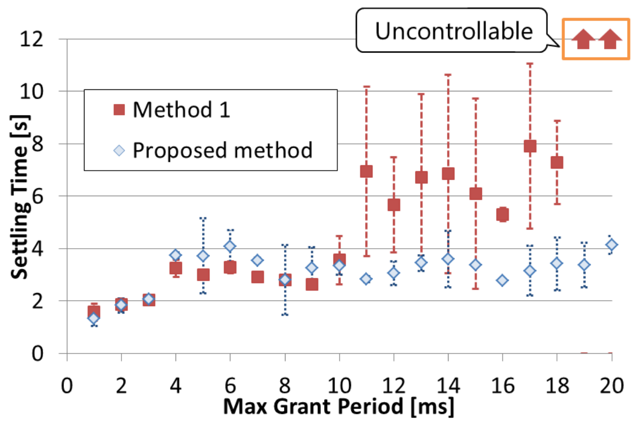


FIGURE 12. Comparison of Method 1 and proposed method under 40 Mbps load environment.

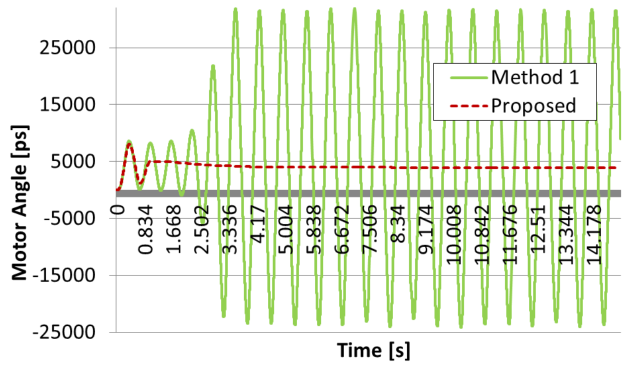


FIGURE 13. Time histories of Method 1 and proposed method with the Max Grant Period of 20 ms in 40 Mbps load environment.

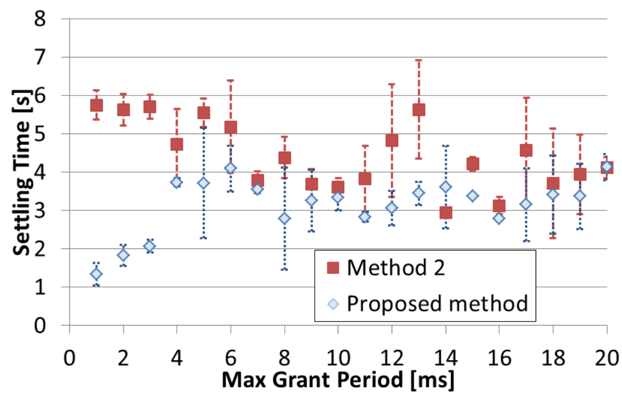


FIGURE 14. Comparison of Method 2 and proposed method under 40 Mbps load environment.

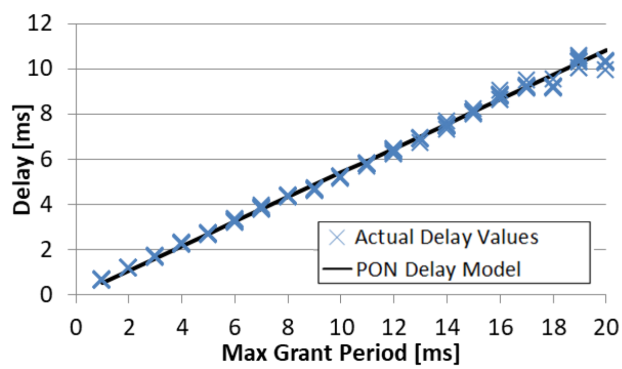


FIGURE 15. Actual delay values and PON delay model outputs.

actual delay. The proposed method is compared to Method 1 and Method 2 in Figures 12 and 14, respectively. Figure 13 shows the time histories of method 1 and the proposed method with the Max Grant Period of 20 ms. As shown in Figure 12, there was no significant difference between Method 1 and the proposed method for Max Grant Periods under 11 ms. However, when it exceeded 11 ms, the settling time became longer with Method 1, and control was no longer possible when the Max Grant Period was 19 or 20 ms. This loss of control was due to the increased latency caused by the upstream traffic. In actual factories, video data from cameras, etc., may be concurrent with control data, so it is necessary to respond flexibly to these delays. By using the PON delay model, the proposed method was able to achieve control even when the conventional method could not.

As shown in Figure 14, as the Max Grant Period increased, the difference in the settling time between Method 2 and the proposed method decreased, but when the Max Grant Period was 1 ms, there was a difference in settling time of up to about 4 ms. Unlike Method 1, Method 2 offered some control, but the proposed method had a shorter settling time.

Figure 15 shows the PON delay model outputs and the actual delay values obtained from 10 trials. Unpredictable delays can degrade control performance, which is reflected as the difference in the settling time. In this experiment, delays caused by the PON section were obtained by Traffic Generator, and compared with our PON delay model. This result confirmed that the model and actual delay values were almost equal and that our proposed PON delay model successfully estimated the delay correctly.

The proposed method achieved settling times within 4 s even when the Max Grant Period was 20 ms. These experiments confirmed that the proposed method could improve the control performance while maintaining the bandwidth utilization efficiency without changing the DBA delay specified to meet the requirements.

VI. CONCLUSION

This paper proposed a novel real-time motion control method based on a PON delay model that offers active compensation of the various delay values expected from a PON system. The model predicts the delay determined by the DBA cycle, and uses it for motion control. The proposed method allows access edge computing to realize highly precise control of a large number of IoT devices. We constructed a basic motor-based experimental system and verified the effectiveness of the proposed method. The experiment showed that the proposed method maintained accurate control performance regardless of PON setting parameters, and so yielded successful motor control within 4 s, whereas the conventional method made the motor unstable and uncontrollable as it not deal with delay changes. Thus, we showed the feasibility of real-time motion control for factory automation using edge computing in PON systems.

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

[4] 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.

[5] 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.

[6] 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, 4th Quart. 2018.

[7] 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.

[8] L. Ruan, M. P. I. Dias, and E. Wong, "Machine learning-based bandwidth prediction for low-latency H2M applications," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 3743–3752, Apr. 2019.

[9] R. Raad, E. Inaty, and M. Maier, "Improving latency and jitter performance in CDMA-based next-generation Ethernet passive optical networks for 5G applications," in *Proc. IEEE 38th Int. Perform. Comput. Commun. Conf. (IPCCC)*, Oct. 2019, pp. 1–8.

[10] K. Kim, K.-H. Doo, H. H. Lee, S. Kim, H. Park, J.-Y. Oh, and H. S. Chung, "High speed and low latency passive optical network for 5G wireless systems," *J. Lightw. Technol.*, vol. 37, no. 12, pp. 2873–2882, Jun. 15, 2019.

[11] T. Tashiro, S. Kuwano, J. Terada, T. Kawamura, N. Tanaka, S. Shigematsu, and N. Yoshimoto, "A novel DBA scheme for TDM-PON based mobile fronthaul," in *Proc. OFC*, Mar. 2014, pp. 1–3.

[12] T. Suzuki, S.-Y. Kim, Y. Koyasako, J.-I. Kani, and J. Terada, "Demonstration of adaptive image transmission that meets various application requirements in 10G-EPON," *IEEE Access*, vol. 8, pp. 186433–186440, 2020.

[13] A. Helmy and A. Nayak, "Toward parallel edge computing in long-reach PONs," *J. Opt. Commun. Netw.*, vol. 10, no. 9, pp. 736–748, Sep. 2018.

[14] A. M. Alqahtani, S. H. Mohamed, T. E. H. El-Gorashi, and J. M. H. Elmirghani, "PON-based connectivity for fog computing," in *Proc. 22nd Int. Conf. Transparent Opt. Netw. (ICTON)*, Jul. 2020, pp. 1–6.

[15] Z. Guo, K. Zhang, H. Xin, M. Bi, H. He, and W. Hu, "An optical access network framework for smart factory in the industry 4.0 era supporting massive machine connections," in *Proc. 16th Int. Conf. Opt. Commun. Netw. (ICOCN)*, Aug. 2017, pp. 1–3.

[16] R. Kubo, T. Michigami, K. Yamada, and M. Yoshino, "Demonstration of wide-area networked motion control over long-reach 10G-EPON based on optical access edge computing," in *Proc. 24th Optoelectron. Commun. Conf. (OECC) Int. Conf. Photon. Switching Comput. (PSC)*, Jul. 2019, pp. 1–3.

[17] 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. IEEE 17th Annu. Consum. Commun. Netw. Conf.*, Jan. 2020, pp. 1–4.

[18] Y. Koyasako, T. Suzuki, S.-Y. Kim, J.-I. Kani, and J. Terada, "Motion control system with time-varying delay compensation for access edge computing," *IEEE Access*, vol. 9, pp. 90669–90676, 2021.

[19] *TR-402 Functional Model for PON Abstraction Interface*, Broadband Forum, Fremont, CA, USA, 2018.

[20] 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. 2015.

[21] 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.

[22] 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.

[23] 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.

[24] 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.

[25] 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.

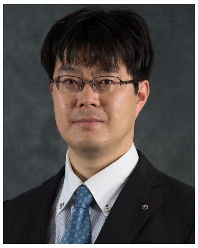
[26] *G.9807.1: 10-Gigabit-Capable Symmetric Passive Optical Network (XGS-PON)*, document G.9807.1, ITU-T Recommendation, 2016.

[27] A. B. D. Kinabo, J. B. Mwangama, and A. A. Lysko, "An overview of time-sensitive communications for the factory floor," in *Proc. IST-Africa Conf. (IST-Africa)*, May 2021, pp. 1–9.

[28] *PROFINET System Description—Technology and Application*, PROFIBUS Nutzerorganisation e.V., Haid-und-Neu-Strasse, Karlsruhe, Germany, 2018.

YUSHI KOYASAKO received the B.E. and M.E. degrees from Keio University, Kanagawa, Japan, in 2015 and 2017, respectively. In 2017, he joined NTT Access Network Service Systems Laboratories, Yokosuka, Japan, 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. In 2014, he joined NTT Access Network Service Systems Laboratories, NTT Corporation, Kanagawa, Japan. From 2019 to 2020, he was involved in OSS development for SDN enabled broadband access (SEBA) at the Open Networking Foundation (ONF), Menlo Park, CA, USA. 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).



TAKASHI YAMADA received the B.E., M.E., and Ph.D. degrees in electrical engineering from Hokkaido University, Sapporo, Japan, in 1997, 1999, and 2003, respectively. In 2003, he joined NTT Access Network Systems Laboratories, Chiba, Japan, where he was involved with research on energy-efficient optical access network systems. From 2013 to 2016, he was with the Research and Development Planning Department, NTT Information Network Laboratory Group, where he was involved in public relations and information strategy. His current research interest includes virtualization of optical access networks.

TATSUYA SHIMADA received the B.E. degree in applied physics from Tohoku University, Sendai, Japan, in 1997, and the Ph.D. degree from Hokkaido University, Sapporo, Japan, in 2009. In 1997, he joined the NTT Multimedia Systems Development Center, Chiba, Japan, where he was engaged in the development of ATM optical access systems. In 1999, he moved to NTT Access Network Service Systems Laboratories, Chiba. Since 2000, he has been working on WDM optical access networks and systems. Since 2014, he has been working on an optical network for 5-G mobile system. He is a member of the Institute of Electronics, Information and Communication Engineers of Japan.

TOMOAKI YOSHIDA (Member, IEEE) received the B.E., M.E., and Ph.D. degrees in communication engineering from Osaka University, Osaka, Japan, in 1996, 1998, and 2007, respectively. In 1998, he joined the NTT Multimedia Systems Development Center, Chiba, Japan, where he engaged in the development of ATM optical access systems. In 1999, he moved to NTT Access Network Service Systems Laboratories, Chiba. Since 2000, he has been engaged in research on next-generation optical access networks and systems. From 2013 to 2015, he was involved in the research of WDM/TDM-PON and worked on the standardization of optical access systems, such as NG-PON2. He is a member of the Institute of Electrical, Information, and Communication Engineers (IEICE) of Japan.