IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 2, 2021, accepted August 14, 2021, date of publication August 16, 2021, date of current version August 24, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3105420

Real-Time Reconfiguration of Time-Aware Shaper for ULL Transmission in Dynamic Conditions

YU NAKAYAMA[1], (Member, IEEE), RYO YAEGASHI[2], (Student Member, IEEE), ANH HOANG NGOC NGUYEN[3], AND YUKO HARA-AZUMI[3], (Member, IEEE)

[1]Institute of Engineering, Tokyo University of Agriculture and Technology, Koganei, Tokyo 184-8588, Japan
[2]Graduate School of Engineering, Tokyo University of Agriculture and Technology, Koganei, Tokyo 184-8588, Japan
[3]Department of Information and Communications Engineering, Tokyo Institute of Technology, Fuchu, Tokyo 183-8538, Japan

Corresponding author: Yu Nakayama (yu.nakayama@ieee.org)

This work was supported in part by JST ACT-I, Japan, under Grant JPMJPR18UL, and in part by JSPS KAKENHI, Japan, under Grant JP20K21789.

ABSTRACT Many applications such as Industrial Internet of Things (IIoT), tactile Internet, and 5G/6G mobile require Ultra-Low Latency (ULL) in data transmission. The end-to-end latency is required to be on the order of sub-millisecond in these applications. The IEEE 802.1 Time Sensitive Networking (TSN) standards have been developed to provide ULL networking. The IEEE 802.1Qbv Time-Aware Shaper (TAS), which is a typical flow control mechanism of TSN, is a traffic shaper to provide deterministic end-to-end ULL transmission for express traffic. The Gate Control List (GCL) of all TAS enabled nodes must be configured in a coordinated manner to ensure ULL. However, existing configuration schemes cannot be employed in highly dynamic conditions where the distribution of time-sensitive streams frequently changes. It will be a significant problem to deal with such dynamic conditions in the future. Therefore, this paper proposes a real-time adaptive gate scheduling scheme for TAS. The scheduling problem is formulated as a boolean satisfiability problem (SAT), and the employment of an FPGA-based solver is proposed to achieve runtime fast computation. The proposed scheme enables real-time reconfiguration for high flexibility and high bandwidth utilization to deal with the dynamics of network conditions. The feasibility of the proposed scheme is confirmed with computer simulations.

INDEX TERMS Packet switching, quality of service, time division multiplexing, runtime, Internet of Things.

I. INTRODUCTION

Many current and future network applications require Ultra-Low Latency (ULL) [1]. Representative examples of such applications are Industrial Internet of Things (IIoT) [2]–[4] and tactile Internet [5], [6], where the end-to-end latency is required to be on the order of sub-millisecond. Another typical application is 5G mobile fronthaul, the latency requirements of which are strictly defined depending on the type of functionality split supported, e.g. 100 μs [7] and 250 μs [8]. Further reduced latency will be required In the era of 6G.

In recent years, the IEEE 802.1 Time Sensitive Networking (TSN) standards have been studied to provide ULL networking in link layer (layer-2). Also, the IETF Deterministic Networking (DetNet) standards have been discussed for providing ULL networking in network layer (layer-3). TSN standardization consists of various technologies including flow synchronization, flow management, flow control, and flow integrity [1]. Although TSN has mainly focused on industrial or in-vehicle networks, the application of TSN technologies has been expanding in accordance with the growing demand of ULL data transmission. For instance, the IEEE 802.1CM TSN for fronthaul was standardized to guarantee ULL data transmission of fronthaul streams in bridged networks [9].

The IEEE 802.1Qbv Time-Aware Shaper (TAS) [10] is a typical flow control mechanism of TSN. TAS is a traffic shaper which is usually employed along with the IEEE 802.1Qbu frame preemption [11] to provide deterministic end-to-end ULL transmission for express traffic. TAS follows the Time-Division Multiplexing (TDM) paradigm using the Gate Control List (GCL) to prevent lower priority traffic from interfering with the transmission of high priority traffic. All TAS enabled nodes must be synchronized in time and their gate time schedule in GCL must be configured in a coordinated manner. However, existing GCL configuration schemes cannot be employed in highly dynamic conditions

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

where the distribution of time-sensitive streams frequently changes. It will be a significant problem to deal with such dynamic conditions in the future.

Therefore, this paper proposes a real-time adaptive gate scheduling scheme for TAS. The scheduling problem is formulated as a boolean satisfiability problem (SAT), and the employment of an FPGA-based SAT solver is proposed to achieve runtime fast computation. The proposed scheme enables real-time reconfiguration for high flexibility and high bandwidth utilization to deal with the dynamics of network conditions. The rest of the paper is organized as follows. The related work and the mechanism of TAS is described in section II. Section III introduces the proposed concept. The computational speed of the implemented SAT solver is evaluated in IV. Then, the results of traffic simulations are described in section V. Finally, the conclusion is provided in section VI.

II. RELATED WORK

A. TIME-AWARE SHAPER

This section briefly explains the mechanism of TAS. An example architecture of a TAS enabled node is depicted in Fig. 1. To utilize TAS, it is required for all nodes from sender to receiver to be synchronized in time with the 802.1AS time reference. In this example, there are eight packet queues that correspond with traffic classes. Queue 0 is used for control packets, and express traffic that requires end-to-end ULL transmission is enqueued into Queue 1, which is a general configuration. TAS employs a gate driver mechanism which opens/closes according to a preset time schedule. The packet transmission from each queue is controlled by the GCL that determine which queues are open during each time period. An example GCL is shown in Fig. 2. The GCL contain multiple gate control entries (GCEs), where 1/0 is set for open/close for each queue, respectively. In time interval T0, the gates for Queues 2–7 are open. Then, in time interval T1, the gate for Queue 1 is open, and so on. The GCL is executed in periodically repeating cycle times.

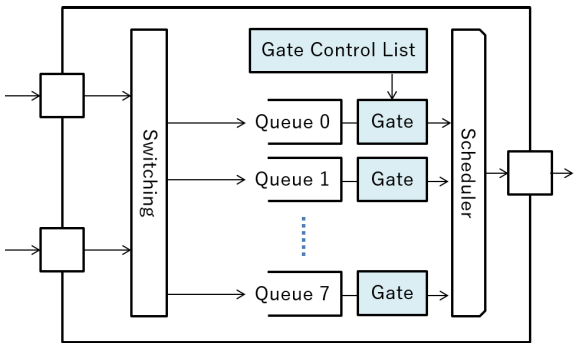


FIGURE 1. Architecture of TAS enabled node.

B. FRAME PREEMPTION

TAS is usually employed along with the frame preemption [11], [12], which is briefly introduced here.

	Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q7
T0	0	0	1	1	1	1	1	1
T1	0	1	0	0	0	0	0	0
T2	1	0	0	0	0	0	0	0
T3	0	0	1	1	1	1	1	1
T4	0	1	0	0	0	0	0	0
T5	0	0	1	1	1	1	1	1
. . .								
T_{MAX}	REPEAT							

FIGURE 2. Gate control list.

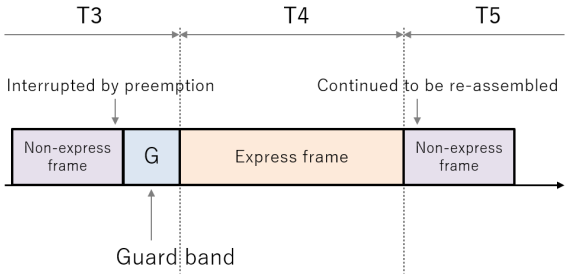


FIGURE 3. Frame preemption.

The mechanism of the frame preemption is depicted in Fig. 3. The frame preemption enables express frames to interrupt transmission of non-express, i.e., preemptable frames. The preempted frame is re-assembled in the next bridge. The frame preemption also contributes to minimize the guard band, which is configured the end of the time intervals for non-express streams as shown in Fig. 3. The guard band is reduced to be $127 = 64 + 63$ byte, which is the sum of the minimum frame size and the maximum remaining frame length that cannot be preempted. As a consequence, this mechanism minimizes the wasted non-express bandwidth. Note that although non-express traffic is expected to be best-effort in many cases, it is not mandatory.

C. TAS RELATED RESEARCH

There have been research efforts for utilizing and improving TAS. Here we briefly introduce them. A formal timing analysis of TAS was presented in [13] to derive worst-case latency bounds. The results showed that TAS can provide ULL transmission when all shapers are synchronized, whereas it suffers from long blocking times if this condition is not guaranteed. Also, the performance of Ethernet-based fronthaul networks with TAS was evaluated through computer simulations using an Opnet model [14]. A gate shrunk (GS-)TAS was proposed in [15], where a GS-frame is added at the end of high priority streams. When a TAS bridge receives a GS-frame, it immediately closes the corresponding gate and opens the gates for other class traffic to improve the link utilization.

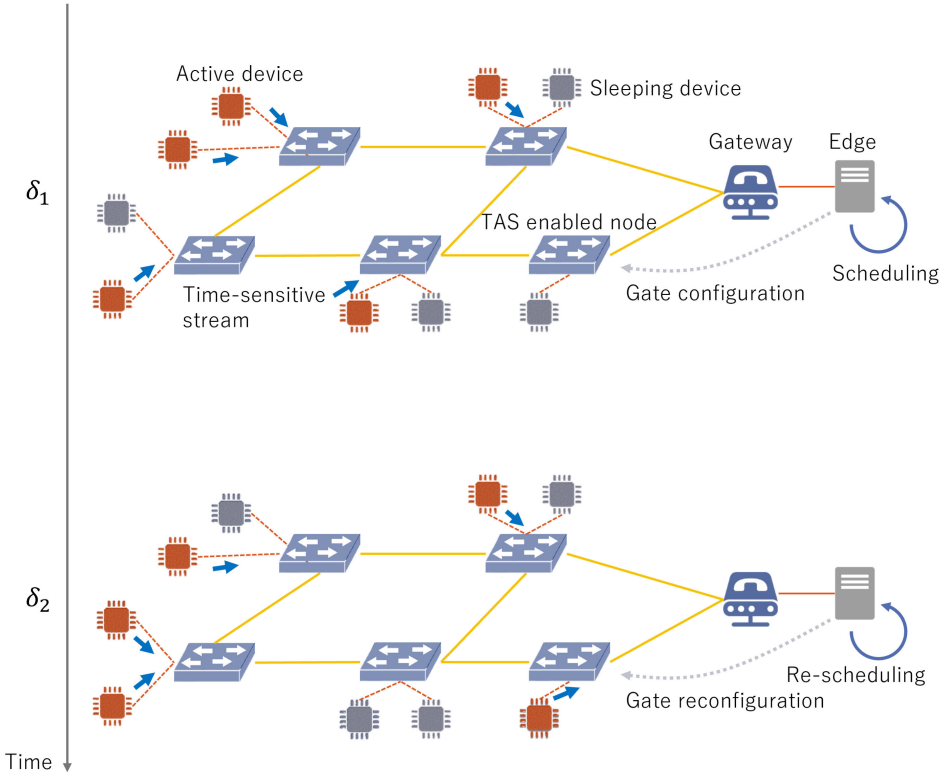


FIGURE 4. Concept of proposed scheme.

This scheme is effective as regards dynamic reconfiguration of time windows based on the real-time fluctuations in the amount of traffic. To decrease the configuration overhead of TSN, an ontology-based plug-and-play approach was presented for in-vehicle networks [16]. This work proposed a plug-and-play framework that utilizes existing protocols such as dynamic host configuration protocol and link layer discovery protocol (LLDP).

It was pointed out that TAS has high configuration complexity and the runtime reconfiguration is an important research topic [1]. As regards runtime reconfiguration, an adaptive TAS algorithm was proposed in [17] which includes an adaptive bandwidth sharing (ABS) and an adaptive slotted window (ASW) mechanisms. The ABS provides transmission opportunity to other queues if the scheduled queue is empty. The ASW shifts the express to non-express gating ratio according to network statistics. A heuristic algorithm was proposed in [18] to determine the GCL at runtime such that the delay requirements are satisfied and the queue usage is minimized to accommodate non-express flows. As regards configuration protocols, the use of IEEE 802.1Qcc stream reservation protocol was investigated in [19] to configure TAS enabled switches. However, existing schemes cannot be employed in highly dynamic conditions because they are based on network statistics. When the distribution of time-sensitive streams frequently changes, statistics-based approach causes packet loss and increase in latency. Therefore, in this paper we propose a real-time reconfiguration scheme for such conditions.

III. PROPOSED SCHEME

A. CONCEPT

This paper proposes a real-time adaptive gate scheduling scheme for TAS. The concept of proposed scheme is depicted in Fig. 4. Time-sensitive streams are sent from active devices and forwarded via TAS enabled nodes. The devices can be robots, drones, and sensors in IIoT, and radio units in mobile networks. This paper assumes high dynamicity for these devices including changes of positions and activation states. They are activated/deactivated according to the situation for efficient resource consumption including energy.

With the proposed scheme, following the state transition, the TAS gates are instantly reserved for activated streams for ULL transmission. At the same time, the gate reservations for deactivated streams are immediately cancelled for efficient use of link bandwidth. The route optimization and gate scheduling are formulated as a SAT, and an FPGA-based SAT solver is employed for the fast computation to deal with the dynamics of network conditions. The proposed algorithm is executed by an edge server based on the reported states of devices.

The advantages of the proposed scheme are high flexibility and high bandwidth utilization. It can address the gate time setting problem which has been a significant issue for

employing TAS by real-time reconfiguration to satisfy the delay requirements. Also, it can efficiently forward other service class traffic by cancelling unnecessary gate reservations to improve bandwidth utilization.

B. PROBLEM DEFINITION

This section describes the problem definition for the proposed scheme.

1) ASSUMPTIONS

The time-synchronization and scheduled traffic are general and fundamental assumptions for TSN. The most of TSN standards only work among time-synchronized nodes using the 802.1AS protocol. The queue isolation of express and non-express traffic is also general and fundamental assumptions for TSN. Note that the proposed scheme assumes only class based queuing, not flow isolation such as per-flow queuing. Also, the existing TSN standards and protocols have focused on only wired networks so that this paper assumes wired TSN. In summary, this paper makes very weak and general assumptions for TSN systems. The only specific assumption of this paper is dynamics of the traffic distribution; the GCL must be updated in accordance with the change of flow distribution. Based on these general assumptions, there is no doubt that the TAS nodes receive and transmit deterministic traffic. The runtime reconfiguration of the proposed approach enables low-latency packet transmission even if the traffic distribution changes.

2) OBJECTIVE

As a premise for employing TAS, it is assumed that time-sensitive streams have periodicity; the future data transmission time can be forecasted and scheduled in TAS enabled nodes. The maximum burst size and the delay requirements of transferred data are predetermined by the application, which are also reasonable assumptions for employing TAS. The length of a time interval is denoted as T_{slot}, and defined as

$$T_{slot} = \frac{B}{L} + \sigma \tag{1}$$

where B denotes the maximum burst size and L denotes the link speed. σ is a small fixed variable defined to absorb the signal processing and propagation delay between nodes.

The goal of the reconfiguration is to determine the forwarding paths of time-sensitive streams satisfying the delay requirement even in the worst case, i.e., the maximum size of burst data are sent from all devices. Since TAS is employed based on time-synchronization between nodes, the hop-by-hop data transmission is executed with synchronized gates of TAS enabled nodes, which can be defined as time slots. Thus, the objective of the defined SAT problem is whether all the time-sensitive streams reach their destination nodes within available time slots. A stream experiences queuing delay during a time slot if multiple streams reach the same node at the same time. The number of time slots is determined by the delay requirements and the slot size T_{slot} defined in (1).

3) VARIABLES

Let $\mathcal{N} = \{1, 2, \ldots, i, j, \ldots N\}$ denote the set of nodes including TAS enabled nodes and gateways, where i and j are the node identifiers. The set of links between nodes in $\mathcal{N}$ are described as $\mathcal{E}$. A directed graph $G = (\mathcal{N}, \mathcal{E})$ is defined with these variables. The connectivity between ith node and jth node is represented using a binary variable $l_{i,j}$; if ith node and jth node is connected $l_{i,j} = 1$, otherwise $l_{i,j} = 0$.

Let t denote the identifier of the time slots. The set of time-sensitive streams is defined as $\mathcal{F} = \{1, 2, \ldots, f, g, \ldots, F\}$. The source node of fth stream is described as s_f. Also, the departure time of fth stream is denoted as δ_f. The set of destination gateways is denoted as $\mathcal{D} = \{1, 2, \ldots, d, \ldots, D\}$, where $\mathcal{D} \subset \mathcal{N}$. Since gateways are destination nodes for the time-sensitive streams, received data are not forwarded from a gateway to other node, which is defined as $l_{d,i} = 0 \quad \forall d, i$.

The number of available time slots is denoted as T, which represents the delay requirements. When a stream must be forwarded within T_{limit}, T is described as

$$T = \frac{T_{limit}}{T_{slot}}. \tag{2}$$

C. SAT FORMULATION

A literal $x_{f,t,i}$ represents that fth stream is at ith node at tth time slot. Note that the activation states of fth stream can be described with $x_{f,t,i}$. That is, the optimum configurations for different network conditions can be found with a single SAT problem.

The constraints of the proposed SAT problem are formulated as the following. A stream is at either one of the nodes at each time slot:

$$x_{f,t,1} \vee x_{f,t,2} \vee \cdots \vee x_{f,t,N} \quad \forall f, t \tag{3}$$

A stream cannot be at multiple nodes at the same time:

$$\neg x_{f,t,i} \vee \neg x_{f,t,j} \quad \forall f, t, i, j \tag{4}$$

A stream can be forwarded from ith node to jth node only if $l_{i,j} = 1$:

$$\neg x_{f,t,i} \vee \neg x_{f,t+1,j} \quad \forall f, t, i, j \; s.t. \; l_{i,j} = 0 \tag{5}$$

The source node of fth stream is s_f:

$$x_{f,t,s_f} \quad \forall f, t \; s.t. \; t \leq \delta_f \tag{6}$$

If fth stream is deactivated, the source node is set as $s_f = d$ not to configure the gates for this stream. A flow has reached either one of gateways within the available time slots:

$$x_{f,\delta_f+T,1} \vee x_{f,\delta_f+T,2} \cdots \vee x_{f,\delta_f+T,d} \vee \cdots \vee x_{f,\delta_f+T,D} \tag{7}$$

Only one stream can be forwarded by a link at the same time slot:

$$\neg x_{f,t,i} \vee \neg x_{f,t+1,j} \vee \neg x_{g,t,i} \vee \neg x_{g,t+1,j} \quad \forall f, g, t, i, j \tag{8}$$

These constraints are all conjuncted to construct a single function as a Conjunctive Normal Form (CNF).

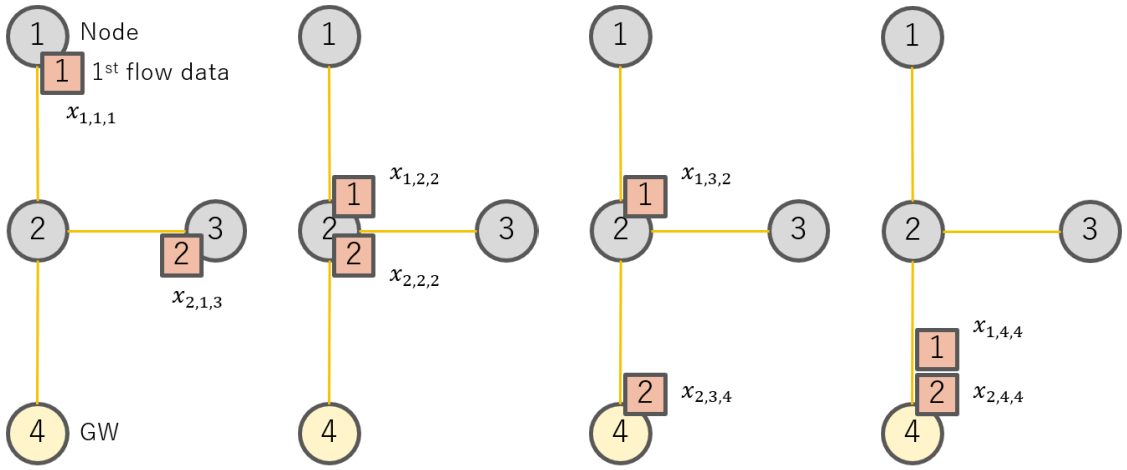


FIGURE 5. Literal expressions.

Fig. 5 shows an example literal expressions. There are three TAS enabled nodes and one gateway. Two time-sensitive streams are forwarded to the gateway. At $t = 1$, the data of 1st flow is at 1st node and the data of 2nd flow is at 3rd node, and thus the literals $x_{1,1,1}$ and $x_{2,1,3}$ are true. At $t = 2$, they are forwarded to 2nd node; $x_{1,2,2}$ and $x_{2,2,2}$ are true. At $t = 3$, the data of 2nd flow is forwarded to 4th node and the data of 1st node is queued; $x_{1,3,2}$ and $x_{2,3,4}$ are true. Finally at $t = 4$, the data of 1st node is forwarded to 4th node; $x_{1,4,4}$ and $x_{2,4,4}$ are true.

D. IMPLEMENTATION

1) SAT SOLVER

The SAT defined in the previous section was formulated in a form of k-SAT for simplicity, where k is an arbitrary integer that represents the maximum number of literals in each clause. The defined problem was converted to 3-SAT for solving it, because most SAT solvers employ 3-SAT based algorithms [20]. 3-SAT is a SAT with at most three literals in each clause. The detailed explanation for the reduction from k-SAT to 3-SAT is provided in [21]. In addition, the variables to be defined was limited based on the problem condition to effectively reduce the design space to be explored. It is sufficient to define only $x_{f,t,i}$ where t is $0 \ldots T$ and i represents the nodes within the T hops from s_f because of the reachability from the source node. We implemented a SAT algorithm [22] on an FPGA Zynq board [23]. The runtime to find a solution was evaluated with the implemented SAT solver. Each problem was represented in a 3-SAT CNF and fed to the implemented SAT solver as input, and the scheduling results were produced if the SAT was satisfiable.

2) GCL CONFIGURATION

The protocol used for GCL configuration is determined by the device specifications. This is because the configuration protocols are out of scope and undefined by the standards. There are many ways for GCL configuration; NETCONF/YANG and classic protocols such as Telnet and SSH can be employed. Some devices store config-files in certain directories to load the configuration; GCL is reconfigured by updating the config-files. Note that the proposed scheme can be employed irrespective of these specifications. The edge server reconfigures GCL of each node based on the computed schedule via the corresponding protocols.

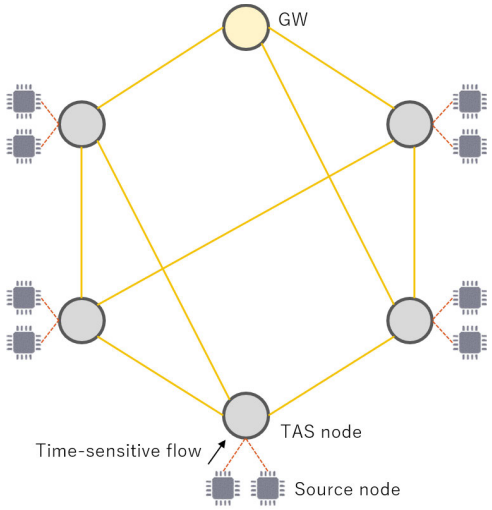


FIGURE 6. Example simulated network.

The frequency of state transitions depends on the application. Since the transmission of express frames is predictable, which is a general assumption for employing TAS, it is reasonable to start the transitional period during the off-period. If it is difficult considering the speed of configuration, it is better to set the GCL for express traffic as open during the transitional period. The speed of configuration depends on the device performance. This parameter affects the system scalability to ensure runtime reconfiguration; a low-performance device can accommodate small number of streams.

IV. SAT SIMULATION

The feasibility of the proposed scheme was evaluated with computer simulations. First, this section introduces the computational speed of the implemented SAT solver.

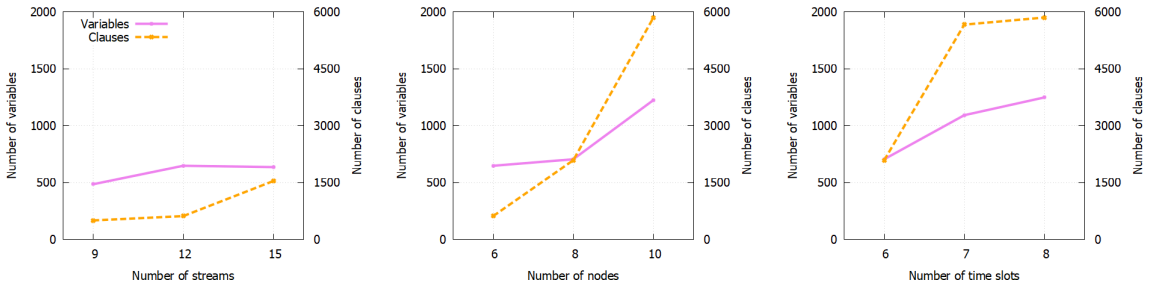
A. SIMULATION CONDITION

Here we explain the simulation conditions. The performance of the proposed scheme was measured in randomly generated networks with different number of nodes, flows, and the available time slots T. Each TAS enabled node was connected with randomly selected other three nodes. The source nodes of time-sensitive streams were uniformly connected to TAS enabled nodes. It was assumed that they periodically send burst data to a gateway node. An example topology is shown in Fig. 6, where there are five TAS enabled nodes, one gateway, and 10 source nodes. When the network topology was determined, the forwarding paths of time-sensitive streams and gate scheduling was computed with the proposed scheme. The computation was executed for 20 iterations with different random initial variables. Also, the scalability was evaluated with different number of nodes, flows, and T.

B. SIMULATION RESULTS

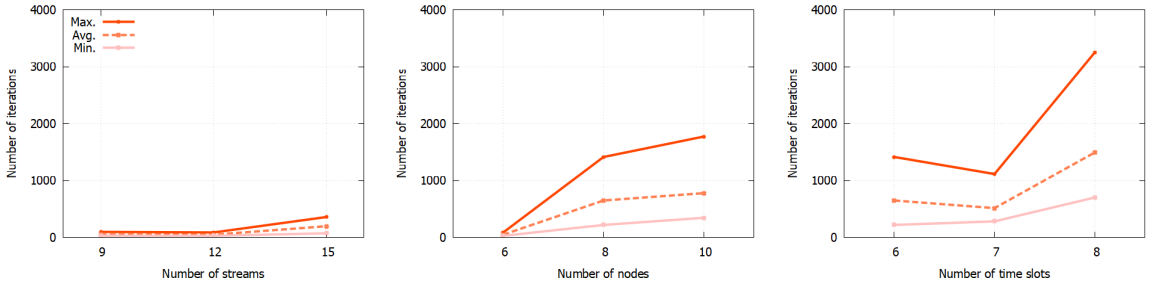
First, we introduce the simulation results for a case of six streams, five nodes, and six time slots. The numbers of variables and clauses were 212 and 478. The minimum, maximum, and average numbers of iterations were 31, 219, and 106, respectively. The clock frequency of the implemented FPGA SAT solver was 146.6 MHz. Since the SAT solver takes two cycles per iteration, the average computational time were 0.72 μs. These results implied that the forwarding paths and TAS gates can be efficiently computed with the proposed scheme in real-time.

Second, the scalability of the proposed scheme was evaluated with different number of streams, nodes, and available time slots. The numbers of variables and clauses in the proposed SAT formulation for each case are shown in Fig.7. The numbers iterations for solving the problem are shown in Fig.8. The scale of the problem increases in accordance with the number of streams, nodes, and time slots, respectively. Among them, the number of nodes and time slots have a large effect on the problem size. The proposed scheme is robust for increase in the number of streams, which is important for accommodating a large number of devices. Based on the assumption that at least 100 MHz frequency was achieved, the execution time was calculated with software program from the number of iterations. Fig. 9 shows the estimated execution time for each condition. The computation finishes in several microseconds on the average. It was confirmed with these results that the runtime reconfiguration of TAS can be achieved with the proposed scheme.



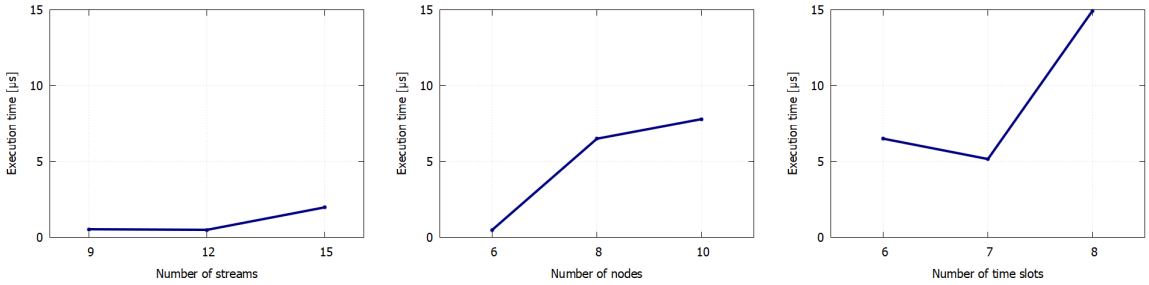
(a) No. streams (6 nodes, 6 time slots)
(b) No. nodes (12 streams, 6 time slots)
(c) No. time slots (8 nodes, 12 streams)

FIGURE 7. Variables and clauses in different conditions.



(a) No. streams (6 nodes, 6 time slots)
(b) No. nodes (12 streams, 6 time slots)
(c) No. time slots (8 nodes, 12 streams)

FIGURE 8. Number of iterations in different conditions.



(a) No. streams (6 nodes, 6 time slots)
(b) No. nodes (12 streams, 6 time slots)
(c) No. time slots (8 nodes, 12 streams)

FIGURE 9. Execution time in different conditions.

V. TRAFFIC SIMULATION

This section reports the performance of the proposed scheme with the results of traffic simulation. The simulation was carried out with a self-developed simulator written in Python3.

A. SIMULATION CONDITION

The simulated topology is depicted in Fig. 10. We simulated a ring topology with G.8032 Ethernet ring protection protocol where blocking ports are configured. There are seven TAS enabled nodes, one gateway, and 14 source nodes of time-sensitive streams. The link bandwidth was 1 Gbps for all of the links. The simulation consisted of five time phases to simulate highly dynamic environments. The length of each phase was 2100 μs. The source nodes were randomly activated/deactivated at the beginning of each phase. The maximum number of activated source nodes were five in each phase. The activated nodes periodically sent 1250 bytes data to the gateway node in every 420 μs assuming control messages. The length of a time slot was set as 10.5 μs assuming $\sigma = 0.5$ μs in (1). The number of time interval within the GCL cycle length was set as 40. The GCL cycle time was set as 420 μs, i.e., the GCL is periodically repeated in every 420 μs. All active source nodes started to transmit express packets at the beginning of each 420 μs cycle assuming time-synchronization for simplicity. To simulate a congested network, the TAS enabled nodes received random non-express streams. The total number of non-express packets during each GCL cycle was 200; 200 packets were randomly arrived at one of the nodes. The packet size was randomly determined from 15000, 10000, 5000, 2500, 1250, or 625 bytes with a uniform distribution. The arrival time was also randomly determined during each cycle with a uniform distribution. It was assumed that each node has sufficient buffer size to store the received packets.

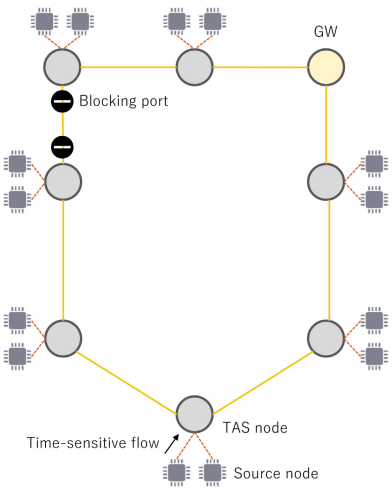


FIGURE 10. Network topology for traffic simulation.

The proposed scheme was compared with the conventional priority queuing and TAS with the conventional static GCL configuration, which is called static TAS in the following. The priority queuing provides strict priority queuing (SPQ) function at each link. To simulate the conventional static TAS which does not provide dynamic reconfiguration of GCL, GCL was only optimally set at the beginning of phase 1. The frame preemption was employed and the guard band was set to correspond with 127 bytes. With the proposed scheme, the GCL in each TAS enabled node was reconfigured at the beginning of each time phase. We measured the worst case delay of express traffic and bandwidth utilization of links between the gateway and the TAS enabled nodes. Note that if queuing delay exceeds 420 μs, which is the GCL cycle length, such unsent packets are discarded to avoid increased delay by packet accumulation in the queue.

B. SIMULATION RESULTS

1) DELAY

Fig. 11 shows the worst case delay of express traffic in each time phase. The delay distribution is also depicted in Fig. 12. The proposed scheme always minimized the delay of express streams by dynamic reconfiguration of GCL. The static TAS achieved low-latency during phases 1 and 3. However, the delay increased during phases 2, 4, and 5. This is because of the time-gap of GCL due to the changes in the flow distribution. The worst case delay with the priority queuing was about 400 μs. This value corresponds with the waiting time

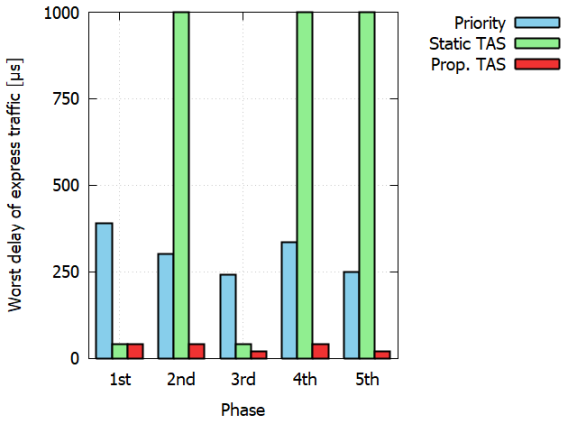


FIGURE 11. Worst case delay of express traffic.

for transmitting the longest non-express frame at each hop, which can be reduced with the frame preemption. This result indicates the advantage of and high-speed reconfiguration of GCL to deal with dynamic environments.

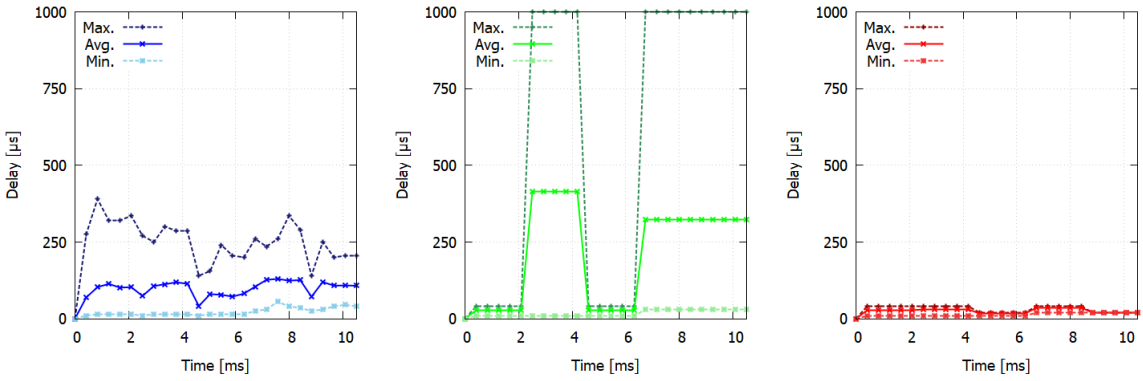
2) BANDWIDTH UTILIZATION

Fig. 13 shows the bandwidth utilization with each scheme, which was calculated from the amount of received data during each 420 μs. Note that it shows the total link utilization including express and non-express streams. The priority queuing reached 100% utilization, because it does not stop sending frames. The proposed scheme achieved comparable performance with the priority queuing. The several percent of unused bandwidth resulted from the guard band length and σ in (1) assuming the processing delay. Although the static TAS achieved the same performance as long as the configured GCL was optimum, the bandwidth utilization was also deteriorated when the traffic distribution changed.

C. DISCUSSION

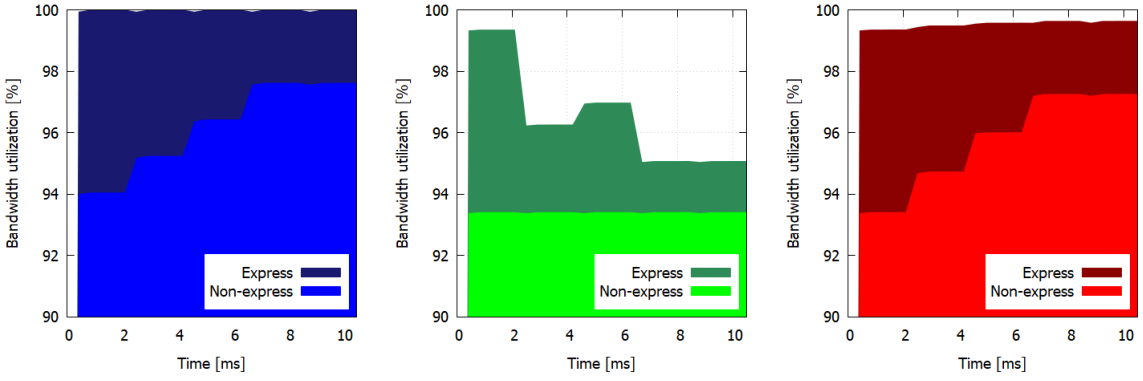
1) SIMULATED SCENARIO

The simulated scenario was determined to simulate a general condition where the total data rate of express traffic is relatively low. Network traffic usually consists of relatively low rate priority flows and enormous volume of non-express flows. However, the effect of the proposed scheme will be exaggerated if the data rate of express traffic increases. This is



(a) Priority queuing (b) Static TAS (c) Proposed scheme

FIGURE 12. Delay distribution of express traffic.



(a) Priority queuing (b) Static TAS (c) Proposed scheme

FIGURE 13. Bandwidth utilization.

simply because the maximum value of unused bandwidth increases in accordance with the data rate. The QoS performance of non-express streams was not evaluated in the simulation. This is because TSN protocols are employed only for ensuring the QoS of time-critical applications. The preemption and TAS inevitably affect the performance of non-express traffic in principle; non-express packets must wait for the transmission of express packets. It can be noted that the proposed scheme does not deteriorate the non-express delay compared with existing TAS thanks to the dynamic reconfiguration. In other words, the proposed technique can minimize the negative impact on non-express streams by dynamically optimizing the GCL.

While the burst size was the same for all flows in the simulation, the proposed scheme can be employed for flows with different burst sizes. To reserve time slots for bursts with different sizes, the time interval length is configured as the greatest common divisor of to be used as a time unit. Several successive time slots are configured for a large burst, and the burst is transmitted using these reserved successive time slots. If the greatest common divisor becomes too small to be configured by the switches, an approximate value (slightly large value) can be used, while the underutilization of bandwidth occurs. Note that this slot length issue is common for TAS. The optimization of time slot length is further study.

The major causes of delay are the waiting time for non-express packets and the time-gap of GCL due to the changes in the flow distribution. The express frames are not accumulated in the FIFO queue for express flows as long as the GCL is appropriately configured, because the gate opens at the time of frame arrival to enable the frame to be transmitted without queuing. The number of express flows does not affect the worst case delay. The obtained result illustrated this phenomenon. Therefore, the simulated condition is appropriate for evaluating the performance of the proposed scheme.

2) DRAWBACKS

The drawbacks of the proposed scheme are the messaging overhead and the requirement for an edge server with an FPGA-based SAT solver to achieve enough computing performance. First, it is required for TAS enabled nodes to exchange control messages for resource reservation. The signaling overhead was not considered in the traffic simulation so that additional signaling overhead will occur when the proposed scheme is employed in real equipment. This overhead is expected to be slight as such control messages of network protocols are generally forwarded as the highest priority packets, i.e., higher priority than express applications. Second, the result of SAT simulation was obtained assuming the defined SAT problem is solved by an FPGA-based solver. Since the hardware computation speed is significantly higher than the speed of a software solver, the performance is expected to deteriorate if we employ a software solver. The required performance for the FPGA is also determined by the scale of the target network. A high-performance and expensive FPGA may be required if we want to configure a large network with a large number of nodes and links. Therefore, further research on the scalability issue is required.

3) WIRELESS ENVIRONMENT

Although the existing TSN protocols have focused on wired networks. it is an important viewpoint to connect wireless devices to TSN. Assume that a Wi-Fi access point is linked to a TAS enabled node. A wireless device is connected to the TAS node via the Wi-Fi access point. The TSN standards only guarantee low-latency data transmission within the TSN domain. When the wireless device sends data, the multiple access schemes such as CSMA/CA makes it difficult to predict the packet arrival time at the TAS node. Hence, the time-synchronization and deterministic packet transmission among TAS nodes and wireless devices constitute future work.

Finally, we did not provide full theoretical analysis for the worst-case delay in this paper since we considered a solution for highly dynamic condition. However, we think that it can be possible in further study to analyze the worst case delay for each transitional period with the proposed scheme based on a deterministic timing analysis method such as [24].

VI. CONCLUSION

ULL is one of the major requirements for today's and future network applications including IIoT and 5G/6G mobile. To guarantee sub-millisecond end-to-end latency, the IEEE 802.1 TSN standards and related researches have been intensely studied. Among them, the IEEE 802.1Qbv TAS is a typical flow control mechanism, which can provide deterministic end-to-end ULL transmission for express streams. However, the coordinated configuration of GCL in highly dynamic conditions has been an unsolved problem for employing TAS because of its high complexity.

To address this problem, this paper proposed a real-time adaptive gate scheduling scheme for TAS to deal with the dynamics of network conditions. With the proposed scheme, gates are instantly reserved/cancelled for activated/deactivated streams following the state transition for efficient resource consumption such as energy. We formulated the route optimization and gate scheduling as a SAT, which can be solved in real-time using an FPGA-based solver. The advantages of the proposed scheme are high flexibility and high bandwidth utilization. We confirmed the feasibility of the proposed scheme via computer simulations. The implementation and experiments of the proposed scheme constitute future work.

REFERENCES

[1] A. Nasrallah, A. S. Thyagaturu, Z. Alharbi, C. Wang, X. Shao, M. Reisslein, and H. ElBakoury, "Ultra-low latency (ULL) networks: The IEEE TSN and IETF DetNet standards and related 5G ULL research," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 1, pp. 88–145, 1st Quart., 2019.
[2] P. Schulz, M. Matthe, H. Klessig, M. Simsek, G. Fettweis, J. Ansari, and S. A. Ashraf, "Latency critical IoT applications in 5G: Perspective on the design of radio interface and network architecture," *IEEE Commun. Mag.*, vol. 55, no. 2, pp. 70–78, Feb. 2017.

[3] S. Samii and H. Zinner, "Level 5 by layer 2: Time-sensitive networking for autonomous vehicles," *IEEE Commun. Standards Mag.*, vol. 2, no. 2, pp. 62–68, Jun. 2018.

[4] C. San Vicente Gutiérrez, L. U. San Juan, I. Z. Ugarte, and V. M. Vilches, "Time-sensitive networking for robotics," 2018, *arXiv:1804.07643*. [Online]. Available: http://arxiv.org/abs/1804.07643

[5] G. P. Fettweis, "The tactile internet: Applications and challenges," *IEEE Veh. Technol. Mag.*, vol. 9, no. 1, pp. 64–70, Mar. 2014.

[6] M. Maier, M. Chowdhury, B. P. Rimal, and D. P. Van, "The Tactile internet: Vision, recent progress, and open challenges," *IEEE Commun. Mag.*, vol. 54, no. 5, pp. 138–145, May 2016.

[7] *Common Public Radio Interface: eCPRI Interface Specification*, eCPRI Specification V2.0, May 2019.

[8] *Study on New Radio Access Technology: Radio Access Architecture and Interfaces*, document TR 38.801, 3GPP, 2017.

[9] *Time-Sensitive Networking for Fronthaul*, Standard 802.1CM Draft 2.2, Mar. 2018.

[10] *IEEE Standard for Local and Metropolitan Area Networks-Bridges and Bridged Networks-Amendment 25: Enhancements for Scheduled Traffic*, Standard 802.1Qbv-2015, Mar. 2016.

[11] *IEEE Standard for Local and Metropolitan Area Networks-Bridges and Bridged Networks-Amendment 26: Frame Preemption*, Standard 802.1Qbu-2016, Aug. 2016.

[12] *Interspersing Express Traffic*, Standard 802.3br Draft 2.3, Oct. 2015.

[13] D. Thiele, R. Ernst, and J. Diemer, "Formal worst-case timing analysis of Ethernet TSN's time-aware and peristaltic shapers," in *Proc. IEEE Veh. Netw. Conf. (VNC)*, Dec. 2015, pp. 251–258.

[14] M. K. Al-Hares, P. Assimakopoulos, D. Muench, and N. J. Gomes, "Modeling time aware shaping in an Ethernet fronthaul," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2017, pp. 1–6.

[15] D. Hisano, Y. Nakayama, T. Kubo, T. Shimizu, H. Nakamura, J. Terada, and A. Otaka, "Gate-shrunk time aware shaper: Low-latency converged network for 5G fronthaul and M2M services," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2017, pp. 1–6.

[16] M. H. Farzaneh and A. Knoll, "An ontology-based plug-and-play approach for in-vehicle time-sensitive networking (TSN)," in *Proc. IEEE 7th Annu. Inf. Technol., Electron. Mobile Commun. Conf. (IEMCON)*, Oct. 2016, pp. 1–8.

[17] A. Nasrallah, A. S. Thyagaturu, Z. Alharbi, C. Wang, X. Shao, M. Reisslein, and H. Elbakoury, "Performance comparison of IEEE 802.1 TSN time aware shaper (TAS) and asynchronous traffic shaper (ATS)," *IEEE Access*, vol. 7, pp. 44165–44181, 2019.

[18] M. L. Raagaard, P. Pop, M. Gutierrez, and W. Steiner, "Runtime reconfiguration of time-sensitive networking (TSN) schedules for fog computing," in *Proc. IEEE Fog World Congr. (FWC)*, Oct. 2017, pp. 1–6.

[19] A. Nasrallah, V. Balasubramanian, A. Thyagaturu, M. Reisslein, and H. ElBakoury, "Reconfiguration algorithms for high precision communications in time sensitive networks: Time-aware shaper configuration with IEEE 802.1 qcc (extended version)," 2019, *arXiv:1906.11596*. [Online]. Available: http://arxiv.org/abs/1906.11596

[20] A. A. Sohanghpurwala, M. W. Hassan, and P. Athanas, "Hardware accelerated SAT solvers—A survey," *J. Parallel Distrib. Comput.*, vol. 106, pp. 170–184, Aug. 2017.

[21] J. Erickson, *Algorithms*. Independently Published, 2019.

[22] M. Aono, S.-J. Kim, and S. Kasai, "Amoeba-inspired spatiotemporal dynamics for solving the satisfiability problem (special issue on new challenges in complex systems science)," *Adv. Sci., Technol. Environmentology*, vol. 11, pp. 37–40, Mar. 2015.

[23] A. H. N. Nguyen, M. Aono, and Y. Hara-Azumi, "Amoeba-inspired hardware SAT solver with effective feedback control," in *Proc. Int. Conf. Field-Program. Technol. (ICFPT)*, Dec. 2019, pp. 243–246.

[24] S. S. Craciunas, R. S. Oliver, M. Chmelík, and W. Steiner, "Scheduling real-time communication in IEEE 802.1Qbv time sensitive networks," in *Proc. 24th Int. Conf. Real-Time Netw. Syst. (RTNS)*, 2016, pp. 183–192.

YU NAKAYAMA (Member, IEEE) received the B.Agr. degree in agriculture, the M.Env. degree in environmental studies, and the Ph.D. degree in information and communication engineering from the University of Tokyo, Tokyo, Japan, in 2006, 2008, and 2018, respectively.

From 2008 to 2018, he was with the NTT Access Network Service Systems Laboratories. He is currently an Associate Professor with the Institute of Engineering, Tokyo University of Agriculture and Technology. He is also the President of neko 9 Laboratories, which is a nonprofit organization in Tokyo. His research interests include mobile computing, network architecture, the IoT, and ultra low latency. He is a member of IEICE and IPSJ.

RYO YAEGASHI (Student Member, IEEE) received the B.E. degree from Tokyo University of Agriculture and Technology, Japan, in 2021, where he is currently pursuing the master's degree in information technology.

ANH HOANG NGOC NGUYEN received the B.E. degree in electronics and communications engineering from Hanoi University of Science and Technology, Vietnam, in 2016, and the M.E. degree in information and communications engineering from Tokyo Institute of Technology, Japan, in 2018, where she is currently pursuing the Ph.D. degree in information and communications engineering. Her research interests include high-level designs for embedded systems and bio-inspired computing. She is a member of ACM. She has been a recipient of the Japanese Government Scholarship (MEXT), since 2016.

YUKO HARA-AZUMI (Member, IEEE) received the Ph.D. degree in information science from Nagoya University, Japan, in 2010. She was a JSPS Postdoctoral Research Fellow with Ritsumeikan University, from 2010 to 2012, during which she was also a Visiting Scholar with the University of California, Irvine, USA, and Karlsruhe Institute of Technology, Germany. In 2012, she joined Nara Institute of Science and Technology as an Assistant Professor. Since 2014, she has been with Tokyo Institute of Technology, where she is currently an Associate Professor. Her research interests include system-level design automation, especially on high-level and logic synthesis, microprocessor architectures, and hardware/software co-design for embedded/the IoT systems. She currently serves as an Organizing and a Program Committee Member for several premier conferences, including DAC, ICCAD, DATE, CASES, ASP-DAC, and FPL.