IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received August 31, 2017, accepted September 27, 2017, date of publication October 2, 2017, date of current version October 25, 2017.

Digital Object Identifier 10.1109/ACCESS.2017.2758803

A Real-Time Delay-Sensitive Communication Approach Based on Distributed Processing

AKIO KAWABATA[1], BIJOY CHAND CHATTERJEE[2], (Member, IEEE), SEYDOU BA[3], (Student Member, IEEE), AND EIJI OKI[4], (Fellow, IEEE)

[1]NTT Network Service System Laboratories, Tokyo 180-8585, Japan
[2]Indraprastha Institute of Information Technology, New Delhi 110020, India
[3]The University of Electro-Communications, Tokyo 182-8585, Japan
[4]Graduate School of Informatics, Kyoto University, Kyoto 606-8501, Japan

Corresponding author: Akio Kawabata (kawabata.akio@lab.ntt.co.jp)

This work was done in part when A. Kawabata and E. Oki were with The University of Electro Communications, Tokyo, Japan. This work was supported in part by JSPS KAKENHI, Japan, under Grant 15K00116, and in part by the Inspire Faculty Scheme, Department of Science and Technology, New Delhi, India, under Grant DST/INSPIRE/04/2016/001316.

ABSTRACT This paper proposes a delay-sensitive communication approach based on distributed processing for real-time applications that provide interactive services for multiple users in order to minimize the delay considering both admissible delay and delay variation rate. The proposed approach considers two scenarios, namely, simultaneous participation and successive participation. In the simultaneous participation, all users and servers are given, and the application is processed in different distributed servers; a user accesses a suitable server as a solution of the server selection problem. In the successive participation, where all servers are given, different users will be participated sequentially in a greedy manner with variation of time, while executing the currently applications. We formulate an integer linear programming (ILP) problem in the simultaneous participation scenario for the distributed server selection when all users and servers are given considering the parameter of admissible delay and delay-variation rate. We prove that the distributed server selection problem is NP-complete. By using a high-performance optimization solver, we solve the introduced ILP problem within a practical time for 800 users. We provide a method for the successive participation scenario by utilizing the ILP formulated in the simultaneous participation. Numerical results indicate that the proposed delay-sensitive communication approach based on distributed processing outperforms the conventional centralized processing approach in terms of delay.

INDEX TERMS Virtual time, real-time application, distributed processing, simultaneous participation, successive participation.

I. INTRODUCTION

Network function virtualization (NFV) [1], [2] affords numerous functions on servers in a cloud environment [3] in order to serve various real-time applications, such as network games, telephone services, and music sessions through networks, using virtualization techniques. The real-time applications suffer large delay produced in wide-area networks due to lack of synchronization among themselves, which can be suppressed up to several tens of milliseconds. The occurrence of event order must be assured for these real-time interactive applications. When the distances among all users and servers are different, a difference among delays is observed. Figure 1 depicts the problem of event reordering among users; the replication of event order is mandatory for these applications when actual events occur.

In a wide-area network, typically two types of application processing approaches are considered, which are (i) centralized processing approach and (ii) terminal processing approach, for interactive applications, such as shooting-type multi-player games. The centralized processing approach processes all the states of users in the centralized server instead of each user's terminal. This type of approach memorizes all the states that have the previous experience of the network delay in order to replicate the event occurrence order at the actual time. In the conventional approach, which follows the idea of centralized processing, the high delay in the wide-area network deteriorates the quality-of-service (QoS).

The terminal processing approach maintains the network delay of its own events in order to replicate the original event occurrence order at the actual time; actual time is stated as a time when the event really occurs at the terminal. The terminal processing approach processes a terminal type application, such as fighting-type network games, and this type

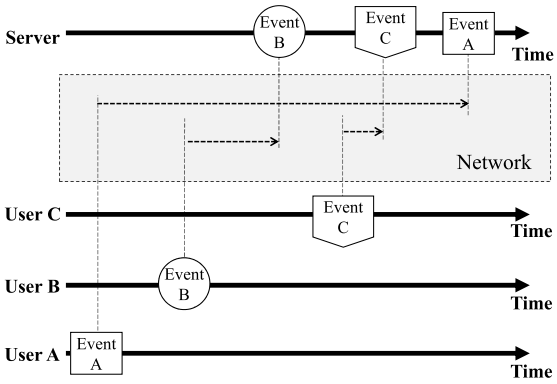

FIGURE 1. Reorder of events caused by network latency.

of approach is impressive to replicate the event occurrence order at the actual time. However, it introduces a growing processing load at the terminal with increase in the number of partners.

To overcome the problem of both types of conventional centralized processing and terminal processing approaches, Kawabata *et al.* [4] presented a distributed processing communication approach for real-time applications to suppress the delay; this work processes the application with distributed servers, where each user accesses a suitable server near it. The work in [4], all the users need to access servers simultaneously, which does not allow users to join and leave successively with variation of time. In the real-time applications, any user can participate and leave successively any time within the minimum admissible delay. In addition, the work in [4] does not consider admissible delay and delay variation rate. The delay variation rate has a significant impact on the end-to-end delay, when network traffic passes through a congested network. The delay variation rate [5] is defined as the variation in a sequence of delay values over time, and it is applicable to QoS-aware protocols that suppose or require relatively constant delay. The delay variation of an individual packet is logically defined as the difference between the actual delay experienced by that packet and a reference delay.

This paper proposes a delay-sensitive communication approach based on distributed processing for real-time network applications that provide interactive services for multiple users in order to minimize the delay considering both admissible delay and delay variation rate. The proposed approach considers two scenarios, namely simultaneous participation and successive participation. In the simultaneous participation, all users and servers are given, and the application is processed in different distributed servers; a user accesses a suitable server as a solution of the server selection problem. In the successive participation, where all servers are given, different users will be participated sequentially in a greedy manner with variation of time while executing the currently applications. A virtual time is introduced in the proposed approach, which is referred to the time that the event occurrence time at the server is modified for reordering in order to maintain the same event occurrence order at the terminals. At the virtual time, all events are replicated in an event occurrence order in the actual time. The event occurrence order is replicated at the virtual time by reordering in the actual time. The processed data at each distributed server is multi-casted to other servers to synchronize the processed data on each distributed server.

Part of this work was presented in [6] and [7]. The main contributions and significance of this work compared to [6] and [7] are explained in the following. In this paper, we discuss the ongoing researches on parallel and distributed communication systems. An integer linear programming (ILP) problem in the simultaneous participation scenario is formulated for the distributed server selection when all users and servers are given considering the parameter of admissible delay and delay-variation rate. We prove that the distributed server selection problem is NP-complete. Thanks to the progress of high-performance optimization solvers, we solve the introduced ILP problem within a practical time for 800 users. We provide the successive participation scenario for the proposed approach. The performance of the proposed approach is evaluated with different types of topologies. In addition, we evaluate the performance of the proposed approach in a typical backbone network, namely Japan Photonic Network (JPN48). Finally, we compare the performance of the proposed approach considering both simultaneous and successive participation scenarios.

The rest of the paper is organized as follows. Section II describes the related works on parallel and distributed communication systems. The framework and overview of the centralized processing and distributed processing approaches are presented in section III. Section IV presents the proposed delay-sensitive communication approach based on distributed processing for real-time network applications for two different scenarios. Section V evaluates the performance of the proposed approach. Finally, this paper is concluded in section VI.

II. RELATED WORKS

This section describes the related works on parallel and distributed communication systems; we focus our discussion on distributed processing considering the event order, followed by the low latency processing approach for network applications.

Several research works [8]–[12] have been carried out to demonstrate the pros and cons of parallel and distributed systems. Researchers explore that problem of reproducing of actual event occurrence while keeping the parallelism of processes is one of the significant issues in parallel and distributed systems [13], [14]. These research issues are typically handled by adapting different synchronization algorithms, which are mainly divided into two categories, namely conservative synchronization and optimistic synchronization [15].

In the conservative synchronization algorithm [16], the order of event occurrence is insured sequentially by

attaching the time information to the event; the terminal processing approach is based on this approach. On the other hand, the optimistic synchronization approach does not maintain the correct event order in advance. If the processing function receives a past event, it guarantees the order of events by rolling back the status and corrects its result. Time Warp [17] is known as an implementation method of the roll-back process; the centralized processing approach is based on this approach. Since the processing history is continuously stored in the rollback approach, the memory consumption becomes a serious issue. To overcome this issue, Cingolani *et al.* [18] explored the practical design and implementation of undo code blocks, which are blocks of instructions implementing the reverse memory side effects generated by the forward execution of the events.

If the delay between two servers connected via a network is different, it is necessary to reproduce the event order [15]. When several servers participate distributed processing and they are connected via a network, it is essential to guarantee the same order of event occurrence. The causal ordering and total ordering are typically used as methods to reproduce the event order. The causal ordering [19] sends an event with a time stamp in order to guarantee the re-ordering of events using the value of the time stamp. On the other hand, the total ordering [20] works in a manner in which all events at all servers use the same order.

In the industrial field, various technologies related to distributed processing have been developed [21]–[25]; their main focus on network games and musical sessions [21]. In the fighting games [22], [23], a full-synchronous communication system holds the same state between players. In the full-synchronous communication system, a player waits until an event arrives from the opponent. The delay is measured by using the previous delay between players. The full-synchronous communication system is referred as conservative synchronization. In a role-playing game [24], the asynchronous communication system does not retain the same state between players; there is a player who manages the entire states. The process is rolled back when past events arrived at players. The asynchronous communication system is referred as optimistic synchronization. In the music field [25], the timing of the events is matched by putting a delay between other players in the own monitor sound.

The edge computing [26]–[29] has been considered in order to improve the delay characteristic by processing applications on the servers located near the user. Several server selection methods have been considered to reduce the delay. Taking this direction, Lee *et al.* [30] presented an adaptive server selection for large-scale interactive online games. Chen *et al.* [31], [32] presented two algorithms in order to reduce the delay variation: one for the client-server architecture and the other for the peer-to-peer architecture. Recent research has observed that, when edge computing is incorporated, the performances is improved 20-70 percent compared to the standard web engine [33].

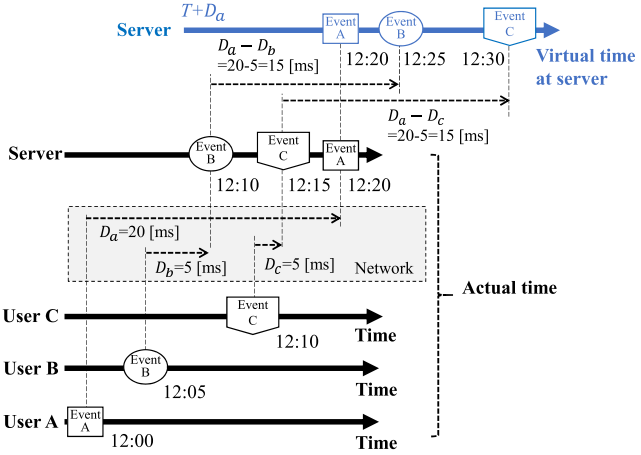

FIGURE 2. Correction of re-order of events in servers.

III. CENTRALIZED PROCESSING AND DISTRIBUTED PROCESSING APPROACHES

A. OVERVIEW

This section is intended as a prerequisite of the proposed delay-sensitive communication approach based on distributed processing, which espouses the concept of edge computing [28], [29]. The idea of edge computing is to process an application, such as a web service, with the help of computing resources that are near to users in order to suppress the network delay. The proposed approach processes the application at different distributed servers; an application is processed on the central server in case of the centralized processing approach. In the proposed approach, applications are handled on different servers in the similar manner to the edge computing or distributed processing.

The user chooses one of the servers among multiple distributed servers in order to process an application in the distributed manner. Then, the processed data is multi-casted to other distributed servers and synchronize the processed data for each server. A virtual time is introduced, which adds the correction time to the current time T for reproducing the occurrence order of an event when it actually occurs. The time of each event at the server is modified based on the corresponding virtual time to replicate the terminal event occurrence order. The application is processed on distributed servers at the virtual time.

We assume that the delays between all server-server pairs and between all user-server pairs are known. We define D_p as the delay between user p and the server that is selected from multiple distributed servers. We define $D_U^{\max}$ as the maximum value of D_p over all users. All events are considered to arrive at any server after $D_U^{\max}$ in the virtual time. It is able to replicate the event occurrence order at the actual time at all distributed servers, if the virtual time is delayed by at most $D_U^{\max}$ measured from the current time. The event of each user is actually processed after $D_U^{\max} - D_p$ at the arrival time, as shown in Fig. 2. All events of users are replicated on each distributed server at the virtual time of $D_U^{\max}$.

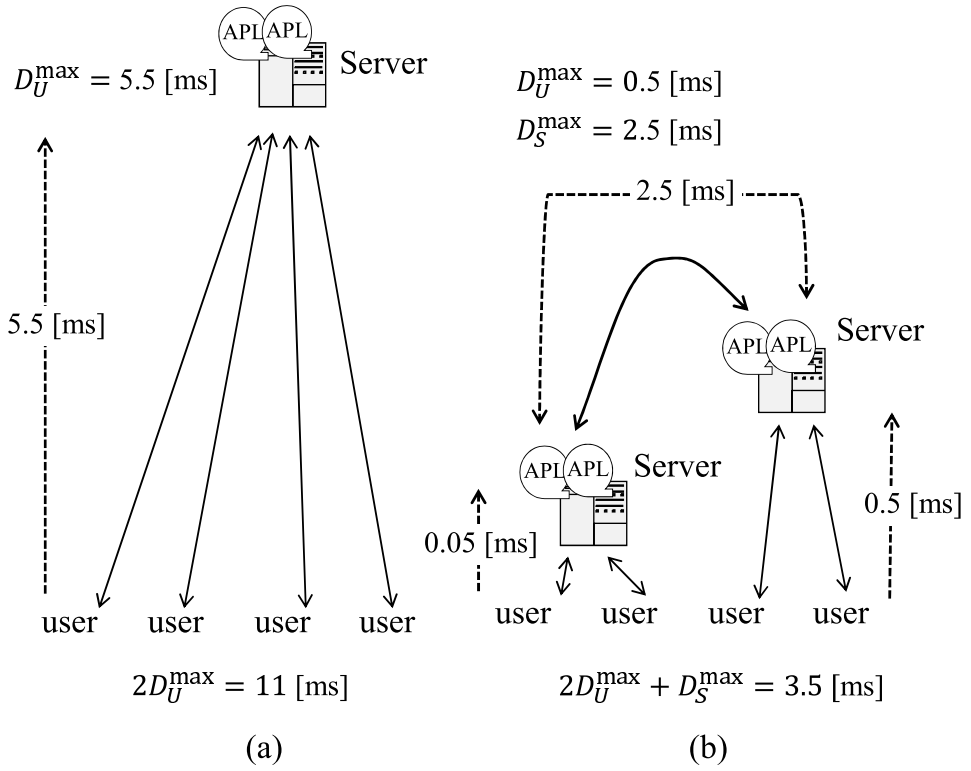

FIGURE 3. Illustrations of (a) centralized processing and (b) distributed processing approaches.

The processed data for each data procession function should be synchronized between distributed servers. We define $D_S^{\max}$ as the maximum value of delays over all server-server pairs. A server that receives the data from another server processes it after at most $D_S^{\max}$. Thus, all event of users on all servers are replicated, in the event occurrence order at the actual time, at the virtual time after at most $D_S^{\max} + D_U^{\max}$. The virtual time at the server is identical to at most $T + D_S^{\max} + D_U^{\max}$, in which all events on all distributed servers are replicated in the event occurrence order at the actual time. The virtual time at the user can be at most $T + D_S^{\max} + 2D_U^{\max}$; $D_U^{\max}$ is added to $T + D_S^{\max} + D_U^{\max}$. As a result, all users are replicated in the event occurrence order at the actual time.

The comparison of the centralized processing approach and the distributed processing approach is shown in Fig. 3. In case of the centralized processing approach, the virtual time at the server to process all users is $T + 5.5$ [ms] by adding 5.5 [ms] to the current time T; the maximum value of the delay between the users and a server is 5.5 [ms]. In the proposed delay-sensitive communication approach based on distributed processing, the virtual time at the server to process all users is $T + 0.5$ [ms] by adding 0.5 [ms] to T; the maximum value of the delay between users and server is 0.5 [ms]. The maximum time for synchronization between servers is 2.5 [ms]. When two approaches are compared in terms of virtual time from users, the conventional centralized processing approach requires $(T + 5.5) + 5.5 = T + 11$ [ms], whereas the proposed delay-sensitive communication approach based on distributed processing requires $T + 2.5 + 0.5 + 0.5 = T + 3.5$ [ms]. We consider these values, such as 5.5 [ms], 0.5 [ms], and 2.5 [ms], for demonstration purposes. We assume that the communication delay is mainly caused by the transmission distance, which is proportional to the transmission distance. For the centralized and distributed approaches, the server locations are given, and the best server assignment is determined by the solving optimization problem explained in section IV.

There are some discussions on pros and cons of communication infrastructures of centralized and distributed systems [8]–[12]. When the distributed system is considered, the delay is less compared to a centralized server. In that case, it can be believed that the advantage of reducing delay is obtained by paying extra resources and cost. This issue was carefully investigated by Shahraeini *et al.* [8]; they first find a minimum spanning tree (MST) as a communication backbone for both centralized and distributed strategies, and then compare major critical parameters of these two backbone networks including latency, reliability, power, and cost. The study in [8] confirmed that the reliability of a communication network is improved by decentralizing infrastructure as the average of communication network hops for centralized control strategy is three times larger than that of distributed one; the reliability of a link reduces if its length increases. The cost of a communication network for distributed control strategy is almost equal to the cost of communication network

for centralized control strategy. Communication network cost in both strategies is mainly determined based on the sum of two major costs, which are the cost of active devices and the cost of passive components. It is observed that the number of MST nodes and length of MST in the centralized and distributed cases are almost equal. The latency is improved in the distributed case. Barnett *et al.* [9] showed that a distributed approach to video-on-demand incurs no greater cost in terms of storage than a centralized approach.

When several servers are used in the distributed approach rather than one server in the centralized approach, the power consumption is increased. This is because several servers are activated at a time and consume extra power. Several power saving strategies [10] are typically used in the distributed approach in order to reduce power consumption. Bhaumik *et al.* [11] introduced a framework for data-centric based radio access networks in order to show that the centralized architecture has the potential in savings of at least 22% of computational resources by exploiting the variations in the processing load across base stations; these savings are achievable with statistical guarantees on successfully processing the base station's signals.

The above discussion summarizes that the cost of a communication network using distributed strategy is comparable with that of the centralized control strategy [8], [9]. In addition, the distributed approach has several other advantages, such as reduced core network bandwidth requirements, improved response time, decreased jitter, and increased reliability [9]. However, using several servers can increase power consumption in the network.

B. SERVER SELECTION PROBLEM IN DISTRIBUTED PROCESSING COMMUNICATION APPROACH

In the distributed processing communication approach, if the user terminal is connected to multiple servers, $D_U^{\max}$ is different according to the selected server. Further, $D_U^{\max}$ is different by the choice of the server from distributed servers. There is a need to select an appropriate server in order to minimize the user correction time, which is $2D_U^{\max} + D_S^{\max}$, under the condition that each server can accommodate at most an allowable number of users.

Note that the users select servers based on several parameters, such as distance, delay, and attenuation of channel. For an example, it may be possible that the user might be closer to server 1 and it is capable to process it, but the server 1 can not be selected due to channel attenuation. In that case, another server is considered based on parameters of distance, delay, and attenuation of channel [12]. For the sake of simplicity, channel attenuation in our ILP formulation in section IV is ignored.

Figure 4 shows two examples of server selection on the same server topology; one of them is the best selection (see Fig. 4(a)) and the other one is a feasible selection but not the best (see Fig. 4(b)). In this example, we assume that each server can accommodate at most three users. In the best selection, user 5 selects server 2, but it does not select server 3,

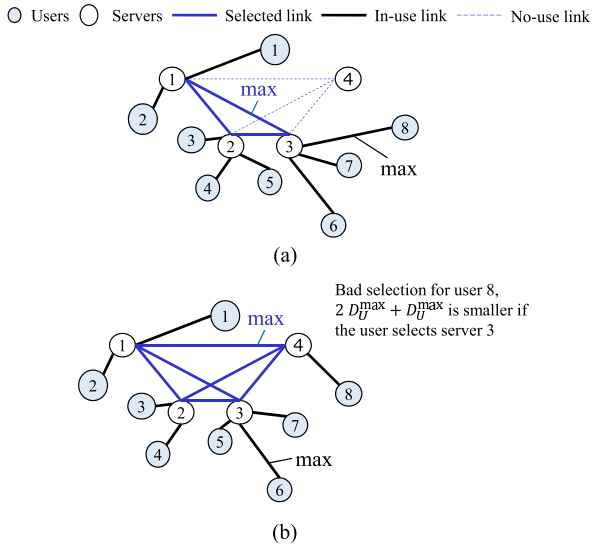

FIGURE 4. Examples of server selection. (a) Best selection. (b) Feasible solution, but not best selection.

which is the nearest one to user 5. This is because server 3 can accommodate at most three users. Another alternate selection option is that user 6 can select server 2 and user 5 can select server 3. In this case, the delay between user 6 and server 2 is set to $D_U^{\max}$, which is not allowed to obtain the minimum user correction time. If user 5 selects server 2 instead of server 3, delay between user 5 and server 2 increases. Note that the selection of server 2 for user 5 does not affect the minimum user correction time. In the feasible selection, as show in Fig. 4(b), all the users select the nearest servers, but the user correction time is not optimum. As user 8 selects server 4 instead of server 3, all links among servers are active and hence the user correction time is larger than that of the best solution in Fig. 4(a). Therefore, a proper server selection is required in the distributed processing communication approach in order to minimize the delay. Note that the distributed system introduces extra delay to synchronize servers and processing control packets, which is not required in the centralized system.

IV. PROPOSED DELAY-SENSITIVE COMMUNICATION APPROACH BASED ON DISTRIBUTED PROCESSING

The proposed delay-sensitive communication approach based on distributed processing minimizes the delay considering both admissible delay and delay variation rate for real-time applications that provide interactive services for multiple users. The proposed approach considers two scenarios, namely simultaneous participation and successive participation, which are discussed in sections IV-B and IV-C.

A. NETWORK MODEL AND NOTATION

The network is represented as undirected graph $G(V, E)$, where V is the set of nodes and E is the set of non-directional links. $V_U \subseteq V$ is the set of all users, and a user is denoted by $p \in V_U$. $V_S \subseteq V$ is the set of servers, and a server is denoted

by $i \in V_S$. $V_U \cup V_S = V$ since a node is either a user or a server, and $V_U \cap V_S = \emptyset$ since a node is not both user and server. $E_U \subseteq E$ is the set of links between user and server, and a link between user $p \in V_U$ and server $i \in V_S$ is denoted by $(p, i) \in E_U$. $(p, i) \in E_U$ exists between all users $p \in V_U$ and all servers $i \in V_S$. A user is allowed to select a server that is not the nearest one to the user. $E_S \subseteq E$ is the set of links between server and server, and a link between server i and server j is denoted by $(i, j) \in E_S$. $(i, j) \in E_S$ exists between all servers $i \in V_S$ and server $j \in V_S$ $(i \neq j)$. $E_U \cup E_S = E$ since a link is either a user-server or server-server link, and $E_U \cap E_S = \emptyset$ since a link does not exist as both user-server and server-server links.

A delay of the link between user $p \in V_U$ and server $i \in V_S$ is denoted by d_{pi}. The maximum value of d_{pi} over in-use $(p, i) \in E_U$ in the proposed approach is denoted by $D_U^{\max}$. A delay of the link between server $i \in V_S$ and server $j \in V_S$ is denoted by d_{ij}. The maximum value of d_{ij} over in-use $(i, j) \in E_S$ in the proposed approach is denoted by $D_S^{\max}$. The maximum number of users that is allowed to belong to server $i \in V_S$ is denoted by M_i. x_{kl} is a binary variable for $(k, l) \in E_S$, where $x_{kl} = 1$ if (k, l) is used, and $x_{kl} = 0$ otherwise. y_i is a binary variable for $i \in V_S$, where $y_i = 1$ if server i is used, and $y_i = 0$ otherwise. $x_{kl} = 1$ if $y_k = 1$ and $y_l = 1$, and $x_{kl} = 0$ otherwise. In other words, $y_k \cdot y_l = x_{kl}$ is satisfied.

B. SIMULTANEOUS PARTICIPATION SCENARIO

In the simultaneous participation scenario, all users and servers are given, and the application is processed in different distributed servers. All users access suitable servers as a solution of the server selection problem.

1) SERVER SELECTION PROBLEM WITHOUT ADMISSIBLE DELAY AND DELAY VARIATION RATE

This subsection presents the proposed approach without considering the parameters of admissible delay and delay variation rate. An ILP problem is formulated to select the distributed servers in the following. It minimizes the delay for determining the virtual time; $(2D_U^{\max} + D_S^{\max})$ is minimized as the objective.

$$\text{Objective min } (2D_U^{\max} + D_S^{\max}) \tag{1a}$$

$$\text{s.t.} \sum_{i \in V_S} x_{pi} = 1, \quad \forall p \in V_U \tag{1b}$$

$$\sum_{p \in V_U} x_{pi} \leq M_i, \quad \forall i \in V_S \tag{1c}$$

$$\sum_{i \in V_S} y_i \leq Y_{\mathrm{MAX}} \tag{1d}$$

$$x_{pi} d_{pi} \leq D_U^{\max}, \quad \forall (p, i) \in E_U \tag{1e}$$

$$x_{ij} d_{ij} \leq D_S^{\max}, \quad \forall (i, j) \in E_S \tag{1f}$$

$$y_i \geq x_{pi}, \quad \forall p \in V_U, \ i \in V_S \tag{1g}$$

$$y_i + y_j - 1 \leq x_{ij}, \quad \forall (i, j) \in E_S \tag{1h}$$

$$x_{ij} \leq y_i, \quad \forall i \in V_S, \ (i,j) \in E_S \tag{1i}$$

$$x_{ij} \leq y_j, \quad \forall j \in V_S, \ (i, j) \in E_S \tag{1j}$$

$$x_{ij} \in \{0, 1\}, \quad \forall (i, j) \in E_S \cup E_U \tag{1k}$$

$$y_i \in \{0, 1\}, \quad \forall i \in V_S \tag{1l}$$

The decision variables are $D_U^{\max}$, $D_S^{\max}$, and x_{pi}. The given parameters are d_{pi}, d_{ij}, Y_{MAX}, and M_i. The objective function in Eq. (1a) is the difference between the virtual time and the current time. Eq. (1b) states that one link is used between a user and a server. Eq. (1c) expresses that the number of users that belongs to server i does not exceed M_i. Eq. (1d) indicates that the maximum number of distributed servers used by users is restricted by Y_{MAX}. As Y_{MAX} increases, the complexity to synchronize the distributed servers becomes larger, which may causes extra delay. The extra delay for synchronization is typically restricted by Y_{MAX}. Eq. (1d) express that the maximum number of servers in the distributed servers. Eq. (1e) indicates that the maximum value of the delay between a user and a server over all user-server pairs that satisfy the admissible delay does not exceed $D_U^{\max}$. Eq. (1f) states that the maximum value of d_{ij} for all $(i, j) \in E_S$ does not exceed $D_S^{\max}$. Eq. (1g) expresses that server i is used, $y_i = 1$, if it is used by at least one user. Eqs. (1h)-(1j) represent $y_k \cdot y_l = x_{kl}$ and $y_k \cdot y_l = x_{kl}$ in the linear model. Eqs. (1k)-(1l) state that x_{ij} and y_i are binary variables.

For the conventional centralized processing approach, the user correction time can be obtained by considering $Y_{\mathrm{MAX}} = 1$ in Eq. (1d). This is because the number of servers are available in one. Further, if the central server is determined as a specific server, the user correction time can be obtained by considering $M_i = 0$ for other serves.

2) TIME COMPLEXITY ANALYSIS OF DISTRIBUTED SERVER SELECTION

To analyze the computational time complexity of the distributed server selection problem, the decision version of the distributed server selection problem is defined in the following.

Definition 1: Given a set of N servers, a set of P users, the distances between servers, the distances between servers and users, the allowable number of users per server M_i, and a number h, is it possible to make an assignment of the users to the servers, and have a maximum delay $w \leq h$?

Theorem 1: The distributed server selection decision (DSSD) problem is NP-complete.

Proof: The DSSD problem is in NP. Given a DSSD instance, we can verify if it is a yes instance within a polynomial time. We check that each user p is connected to a server i and compute the maximum delay between users and servers, D_U, in $O(P)$. We compute the maximum delay between servers, D_S, in $O(N^2)$. Then, we compute w, and verify if w is at most h in $O(1)$. Therefore, the overall time complexity is $O(N^2 + P)$.

To prove that the DSSD problem is an NP-complete, we show that the 3-SAT problem, which is proved to be an NP-complete problem [34], is polynomial time reducible to the DSSD problem. The 3-SAT problem can be stated as:

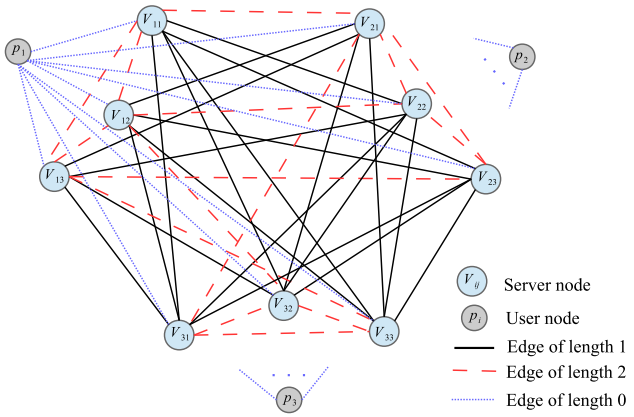

FIGURE 5. Graph *G* corresponding to 3-SAT problem with three clauses.

Given a set of n boolean variables, and k clauses of three elements each, is there a truth assignment that satisfies all clauses?

We construct an instance of the DSSD problem from any instance of the 3-SAT problem in the following. The schematic image of the construction is depicted in Fig. 5.

- Create graph G with $3k$ server nodes grouped into k sets of three nodes V_{ij}, where $i = 1, 2, \cdots, k$ and $j = 1, 2, 3$, and k user nodes.
- All server nodes are connected by an edge
  - The length of an edge $(V_{ij}, V_{i'j'})$ is set to 1 whenever $i \neq i'$, and the element of V_{ij} and $V_{i'j'}$ are not each other's negations. In other words, the edge represents two nodes corresponding to elements that have a compatible true assignment.
  - Otherwise, the edge length is set to 2.
- Each user node is connected to all server nodes with edges with length 0.

To complete the DSSD instance we set the capacity of each server M_i to 1. The value h is set to 1.

Next, we show that the DSSD instance is feasible if and only if there is a satisfiable 3-SAT assignment.

Suppose that there is a satisfiable 3-SAT assignment. Then we can select k nodes, one corresponding to a *true* assignment from each clause, which are all connected in G with edges of length 1. We assign the k users to the k selected server nodes, one each. The maximal delay w is 1, which satisfies $w \leq h$. Therefore, the DSSD instance is feasible.

Conversely, suppose that the DSSD instance is feasible. Then, there is a set of k fully connected server nodes with edges of length at most 1 between them. By the definition of graph G, these nodes correspond to variables with compatible true assignment. Therefore, the truth assignment that sets the variable corresponding to the k nodes to *true* satisfies all the clauses. Thus, the 3-SAT instance is satisfiable. ■

3) SERVER SELECTION PROBLEM CONSIDERING ADMISSIBLE DELAY AND DELAY VARIATION RATE

This subsection considers the parameters of admissible delay and delay variation rate for the server selection problem in order to suppress the end-to-end delay of real-time applications.

We assume that the real-time applications are processed under the admissible end-to-end delay. The parameter D_{lim} is introduced as the maximum value of $2D_U^{\max} + D_S^{\max}$. Note that small value of D_{lim} indicates that the application is sensitive to delay, and large value of D_{lim} indicates that the application can be in tolerable with long delay. The number of users that are excluded, since D_{lim} is not satisfied, is denoted by N_{out}. z_{pi} is a binary variable for $(p, i) \in E_U$, where $z_{pi} = 1$ if the user link (p, i) does not satisfy the admissible delay, and $z_{pi} = 0$ otherwise. q_p is a binary variable for $p \in V_U$, where $q_p = 1$ if user p does not satisfy the admissible delay, and $q_p = 0$, which means that user p is excluded, otherwise. We have a relationship of $q_p \cdot x_{pi} = z_{pi}$.

On the other hand, the parameter of delay variation has a significant impact on the end-to-end delay, when network traffic passes through a congested network. Therefore, we also consider the delay variation in our formulation; the parameter of delay variation f_{pi} is introduced. We define the maximum delay of link (p, i) in a congested network as $d_{pi} \times (1 + f_{pi})$, introducing the parameter of delay variation f_{pi}. For example, d_{pi} at link $(p, i) \in E_U$ is 1 ms when network is not congested, the maximum delay of link (p, i) in a congested network is 2.5 ms = 1 ms × (1+1.5) at the condition of $f_{pi} = 1.5$.

In this subsection, we replace d_{pi} with $d_{pi}(1 + f_{pi})$, and the user selects the server with least delay to minimize the effect of delay variation. This suppresses the increase of the user terminal correction time, which depends on delay; selecting the server with least delay also shortens the link length to be used between the user terminal and the server.

The sum of the delay for all users is denoted by F_f. We further consider F_f in the optimization formulation in which the user terminal selects the nearest server. Selecting nearest server can be effective for a network design method as it reduces the line length of a wired link, which provides the stable quality in radio link. In the following, we formulate an ILP problem considering delay variation rate and the sum of the delay of all users.

$$\text{Objective min} \quad \{N_{\text{out}} + \alpha(2D_U^{\max} + D_S^{\max}) + \beta F_f\} \tag{2a}$$

$$\textbf{s.t.}\ d_{pi}(1 + f_{pi})(x_{pi} - z_{pi}) \leq D_U^{\max}, \quad \forall (p, i) \in E_U \tag{2b}$$

$$2D_U^{\max} + D_S^{\max} \leq D_{\text{lim}} \tag{2c}$$

$$N_{\text{out}} = \sum_{p \in V_U} q_p \tag{2d}$$

$$F_f = \sum_{(p,i) \in E_U} x_{pi} d_{pi}(1 + f_{pi}) \tag{2e}$$

$$z_{pi} \leq x_{pi}, \quad \forall (p, i) \in E_U \tag{2f}$$

$$z_{pi} \leq q_p, \quad \forall (p, i) \in E_U \tag{2g}$$

$$z_{pi} \leq x_{pi} + q_p - 1, \quad \forall (p, i) \in E_U \tag{2h}$$

$$z_{pi} \in \{0, 1\}, \quad \forall i \in E_U \tag{2i}$$

$$q_p \in \{0, 1\}, \quad \forall i \in V_U \tag{2j}$$

Eqs. (1b) - (1d), Eq. (1f), Eqs. (1g) - (1j), and
Eqs. (1k) - (1l)

The first term of the objective function in Eq. (2a) is the number of users that are excluded since the delay-time constraint is not satisfied, N_{out}. The second term in Eq. (2a) is the difference between the virtual time and the current time, $2D_U^{\max} + D_S^{\max}$. The third term in Eq. (2a) is the sum of delay of all users, F_f. Eq. (1a) is replaced with Eq. (2a). α is a constant that expresses the weight of the first term in Eq. (2a) and β is a constant that expresses the weight of the second term in Eq. (2a). The parameter of α is set so that the relationship of $N_{\text{out}} \gg \alpha\ (2D_U^{\max} + D_S^{\max})$ must be satisfied. The parameter of β is set so that the relationship of $(2D_U^{\max} + D_S^{\max}) \gg \beta F_f$ must be satisfied. Eq. (2b) determines $D_U^{\max}$ considering the delay variation. Eq. (2c) represents that the difference between the virtual time and the current time, $2D_U^{\max} + D_S^{\max}$, does not exceed the admissible delay D_{lim}. Eq. (2d) indicates that the number of users that are excluded by exceeding the admissible delay is N_{out}. Eq. (2e) denotes the sum of delay of all users considering the delay variation ratio f_{pi}. Eqs. (2f) - (2h) represent $z_{pi} = x_{pi} \cdot q_p$ in the linear model. Eqs. (2i) - (2j) state that z_{pi} and q_p are binary variables.

In this formulation, the decision variables are $D_U^{\max}$, $D_S^{\max}$, N_{out}, F_f, q_p, y_i, x_{pi}, z_{pi}, and the given parameters are Y_{MAX}, M_i, D_{lim}, d_{pi}, d_{ij}.

C. SUCCESSIVE PARTICIPATION SCENARIO

This subsection considers successive participation scenario for different users, which will be added sequentially with variation of time. There is a possibility of interruption of application if the in-use user is moved to a different server. In the successive participation, new users can either select the server which is already selected by in-use users or some different server without interrupting the application. In the successive participation scenario, we introduce a method that accommodates users sequentially in a greedy manner without interrupting an application that is currently executing, which is discussed in section IV-C1.

1) FORMULATION OF SERVER SELECTION PROBLEM FOR SUCCESSIVE PARTICIPATION METHOD

Along with the conditions of section IV-B1, we further consider a set of nodes, which represents the in-use users, denoted by V_U^1, and a set of nodes, which represents the new participation users, denoted by V_U^2. A set of servers, which is selected by the set of in-use users, is denoted by V_S^1 and, the set of servers that can be selected by the set of new users is expressed by V_S^2. A set of in-use links between user terminals and the server-to-server is denoted by E_U^1 and the set of links connecting new user terminals to the servers is expressed by E_U^2.

In this setting a user, respectively a link, is either in V_U^1 or V_U^2, respectively E_U^1 or E_U^2, and therefore, $V_U^1 \cup V_U^2 = V_U$, $E_U^1 \cup E_U^2 = E_U$, $V_U^1 \cap V_U^2 = \emptyset$, $E_U^1 \cap E_U^2 = \emptyset$. In section IV-A, we already mentioned that x_{kl} is used as a binary variable for $(k, l) \in E_S$, where $x_{kl} = 1$ if (k, l) is used, and $x_{kl} = 0$ otherwise. y_i is a binary variable for $i \in V_S$, where $y_i = 1$ if server i is used, and $y_i = 0$ otherwise. The value of x_{pi} for link $(p, i) \in E_U^1$ between user terminal $p \in V_U^1$ and server $i \in V_S$ is given and represented by x_{pi}^{given}. The value y_i is determined by whether an in-use user terminal is connected to server $i \in V_S^1$, and is represented by y_i^{given}. For $i \in V_S^2$, y_i^{given} is defined as $y_i^{\text{given}} = 0$, since the considered servers are inactive.

In the following, we formulate an ILP problem in order to promote the selection of available servers that are already in-use.

Objective Eq. (1a)

$$\textbf{s.t.} \sum_{i \in V_S} x_{pi} = 1, \quad \forall p \in V_U^2 \tag{3a}$$

$$x_i \geq x_{pi}^{\text{given}}, \quad \forall (p, i) \in E_U^1 \tag{3b}$$

$$y_i \geq y_i^{\text{given}}, \quad \forall i \in V_S$$

Eqs. (1c)- (1l) (3c)

The server selection in the successive participation method is intended for the new participating user $p \in V_U^2$. In the successive participation method, initially, the set of all users and the set of links, each of which connects a user to a server, are initialized to null. When a set of new users joins the server selection problem, a set of links is generated, which connects newly joined users to all servers. Thereafter, we solve the optimization problem, which is formulated in section IV-C1, in order to find the suitable servers considering delay. x_{ij} for in-use user is treated as a given parameter and replaced with x_{ij}^{given}. Similarly, y_i for in-use server is treated as a given parameter, and replaced with y_i^{given}. The set of new users and the set of links are updated considering newly joined users. This process is continued when new users join in the successive participation method.

In the successive participation method, the decision variables are $D_U^{\max}$, $D_S^{\max}$, N_{out}, F_f, q_p, y_i, x_{pi}, z_{pi}, and the given parameters are Y_{MAX}, M_i, D_{lim}, d_{pi}, d_{ij}, x_{pi}^{given}, y_i^{given}.

The detail steps of the successive participation method are given in the following.

- Step 1:
  - Initialization, $V_U = 0$, $E_U = 0$.
- Step 2:
  - Participation of a set of new users V_U^2, and generate E_U^2 form V_U^2.
  - If $V_U^2 = \emptyset$, generation is finished, if $V_U^2 \neq \emptyset$, go to Step3.
- Step 3:
  - Solve the optimization problem formulated in section IV-C1.
- Step 4:
  - The value of x_{kl} for considered user is substituted with x_{kl}^{given}.

- The value of y_i for considered user is substituted with y_i^{given}.
- $V_U^1 \leftarrow V_U^1 \cup V_U^2$
- $E_U^1 \leftarrow E_U^1 \cup E_U^2$
- Go to Step 2.

2) FORMULATION OF SERVER SELECTION PROBLEM CONSIDERING REDUCING WAITING TIME

The waiting time for selecting suitable servers in the successive participation scenario should be reduced for better performance. The waiting time is the time between the moment a user puts a participation request and the moment it joins the communication. A suitable server is selected by solving the optimization problem. We limit the calculation range of optimization problem, when a new user, $p \in V_U^2$, participates. By limiting the parameter at the optimization problem, we can reduce the calculation time, and thus the waiting time for server selection is suppressed.

$D_U^{\max}$ is already decided for in-use user and is denoted by $D_U^{\max,\text{given}}$. The number of in-use users that selected server i is denoted by M_i^{given}, which is $\sum_{p \in V_U^1} x_{pi}, \forall i \in V_S$. In the following, we formulate an ILP problem in order to limit the calculation range for new participating users.

$$\text{Objective Eq. (1a)}$$

$$\textbf{s.t.} \sum_{p \in V_U^2} x_{pi} \leq (M_i - M_i^{\text{given}}), \forall i \in V_S \tag{4a}$$

$$x_{pi} d_{pi} \leq D_U^{\max}, \quad \forall (p,i) \in E_U^2 \tag{4b}$$

$$D_U^{\max,\text{ given}} \leq D_U^{\max}$$

$$\text{Eq. (1d), Eqs. (1f)- (1l), Eqs. (3a)-(3c)} \tag{4c}$$

Eq. (4a) excludes the in-use users in calculation. Eq. (4b) and Eq. (4c) limits the calculation range for new participating users. In the successive participation method considering reducing waiting time, the decision variables are $D_U^{\max}, D_S^{\max}, N_{\text{out}}, F_f, q_p, y_i, x_{pi}, z_{pi}$, and the given parameters are Y_{MAX}, M_i, M_i^{given}, $D_U^{\max,\text{given}}$, y_i^{given}, d_{pi}, x_{pi}^{given}, d_{pi}, d_{ij}.

V. EVALUATION

This section evaluates the performance of the proposed delay-sensitive communication approach based on distributed processing. The server selection problem in the proposed approach is NP-Complete. Thanks to the progress of high-performance optimization solvers, we solve the introduced ILP within a practical time for 800 users. The proposed approach is evaluated in the following conditions. The ILP model is solved by the IBM(R) ILOG(R) CPLEX(R) Interactive Optimizer 12.6.1.0 [35] using Intel (R) Xeon (R) CPU E5-2609 2.5 GHz 8-core, 64 GB memory. We use the Monte-Carlo simulation for distributing users's location randomly. We consider the following assumptions for the purpose of evaluation. (i) The communication delay is mainly caused by the transmission distance, which is proportional to the transmission distance. (ii) The communication delays of user-to-server and server-to-server are measured in advance. (iii) The processing delay at the server is much smaller than that of the communication delay. (iv) All users can directly communicate to any distributed server, (v) The transmission medium between user and server is either wireless or wire; a wire link is typically considered between server and server.

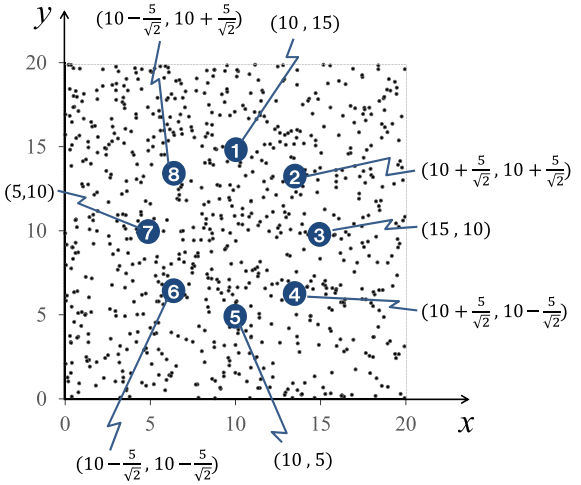

FIGURE 6. Coordinates of user and server locations.

In the following, we evaluate both simultaneous participation and successive participation scenarios of the proposed approach.

A. SIMULTANEOUS PARTICIPATION

We first focus our discussion on the proposed approach without admissible delay and delay variation rate followed by the proposed scheme with admissible delay and delay variation rate.

1) PROPOSED APPROACH WITHOUT ADMISSIBLE DELAY AND DELAY VARIATION RATE

We first consider a simple model, where all servers and users are located in a square. The servers are located in an octagon shape considering the center of the server topology at the intersection of the diagonal, which is the center of the square. The location of all distributed servers is defined in the coordinates, as shown in Fig. 6; we assume 800 users for the purpose of simulation, and the user location follows a random uniform distribution in the square region.

Figure 7 shows the server selection results of the proposed approach for 800 users for different server topologies without considering admissible delay and delay variation rate. In the star topology, as shown in Fig. 7(a), server 1 and server 5 are selected to minimize the delay. In this topology, by sharing the link between servers 1 and 5 among all users, $D_S^{\max}$ becomes the smallest, and thus the value of $2D_U^{\max} + D_S^{\max}$ is minimized. The value of $2D_U^{\max} + D_S^{\max}$ is 32.2 [ms], where $D_U^{\max} = 11.1$ [ms] and $D_S^{\max} = 10$ [ms]. In the mesh topology, as shown in Fig. 7(b), all servers, except server 1,

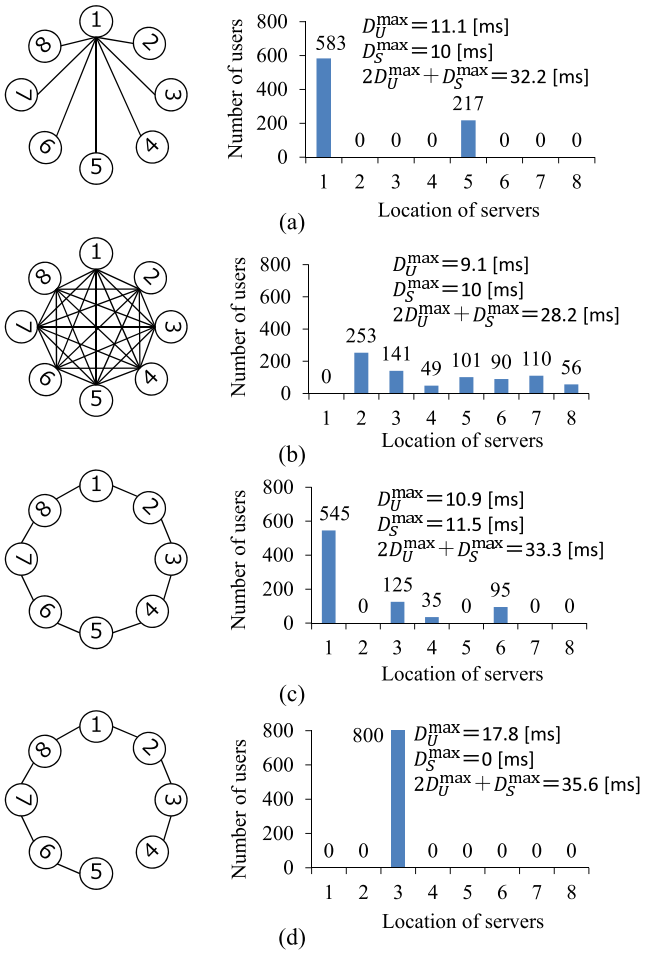

FIGURE 7. Server selection results of 800 users. (a) Star topology. (b) Mesh topology. (c) Ring topology. (d) Line topology.

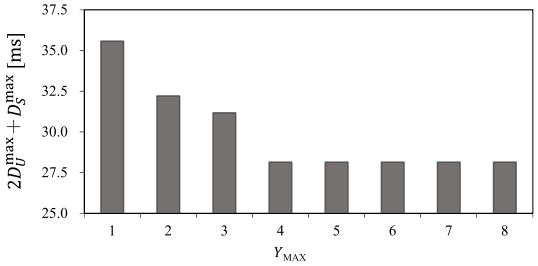

FIGURE 8. Dependence of Y_{MAX} on $2D_U^{max}+D_S^{max}$ for mesh topology shown in Fig. 7(b).

are selected to minimize the user correction time; the number of users that select server 1 is zero. This is because D_S^{max} and D_U^{max} are not affected by whether users select server 1 or not in the following reasons. (i) D_S^{max} is the longest diagonal length of the octagon whether users select server 1 or not, (ii) Server 1 is located in the top of the octagon as shown in Fig. 7(b). Users whose nearest server is server 1 can choose server 2 or server 8 without affecting D_U^{max}. The optimization solver may have multiple optimum solutions with and without server 1. In this evaluation, the solver provides the optimum solution without server 1 as one of multiple optimum solutions. The value of $2D_U^{max}+D_S^{max}$ is 28.2 [ms], where $D_U^{max}=9.1$ [ms] and $D_S^{max}=10$ [ms]. Server 1, server 3, server 4, and server 6 are selected to minimize the value of $2D_U^{max}+D_S^{max}$ in the ring topology, as shown in Fig. 7(c). Note that, in the ring topology, the selection of servers in diagonal way is avoided to minimize the value of $2D_U^{max}+D_S^{max}$; the value of $2D_U^{max}+D_S^{max}$ is 33.2 [ms], where $D_U^{max}=11.1$ [ms] and $D_S^{max}=10$ [ms]. Finally, in the line topology, as shown in Fig. 7(d), there is no existence of the distributed processing approach; only server 3 is selected for all users. In this topology, selecting the nearest server with detour route increases the value of $2D_U^{max}+D_S^{max}$; the value of $2D_U^{max}+D_S^{max}$ is 35.6 [ms], where $D_U^{max}=17.8$ [ms] and $D_S^{max}=0$ [ms].

The solving time of the ILP introduced in section IV-B1 for mesh, star, ring and line topologies are 9.4 [sec], 15.3 [sec], 7.2 [sec] and 6.0 [sec], respectively.

Figure 8 depicts the dependence of Y_{MAX} on $2D_U^{max}+D_S^{max}$ for mesh topology shown in Fig. 7(b). In this evaluation, we consider a mesh topology as it provides a lower value of $2D_U^{max}+D_S^{max}$ compared to any other topology. We observe that, when $Y_{MAX}=1$, which is referred as the centralized processing approach, the value of $2D_U^{max}+D_S^{max}$ is the highest. The value of $2D_U^{max}+D_S^{max}$ decreases with increase in Y_{MAX}. In other words, we can say that the delay is reduced by adapting the distributed processing approach. Furthermore, we notice that the distributed processing approach is not effective when $Y_{MAX}>4$. When Y_{MAX} becomes larger than four, the value of D_U^{max} is not decreased.

From the above discussion, we summarize that the performance of the proposed approach mainly depends on the server selection, and the selection of servers varies with topologies. In the simple model, we observe that the performance of the proposed approach in terms of $2D_U^{max}+D_S^{max}$ is the best in the mesh topology, whereas the line topology provides the worst performance. Evaluation of the proposed approach considering the server selection problem in a typical backbone network is indispensable, which is performed in the following.

Further, we consider a typical backbone network for evaluating the proposed approach. For this purpose, we consider Japan Photonic Network (JPN48) model [36], as shown in Fig. 9. In JPN48 model, all servers are logically connected to form a full mesh topology; the link between Okinawa and Sapporo, is connected by the shortest route on the topology. We assume that four users are connected to each JPN48 node, as shown in Fig. 9. The delays among four users to each node are considered as 0.1 [ms], 0.5 [ms], 1 [ms], and 2 [ms], respectively; we consider the distance of 1 km is equivalent to delay of 5 μs. The delays of 0.1 [ms] and 2 [ms] are introduced when the distances between a user and a server are considered 20 km and 400 km, respectively. In this evaluation, the number of nodes is 48, each of which is eligible to be a server, the number of users is 192 ($=48\times4$), and the number of at most server-to-server links is 1128 ($=\frac{48\times47}{2}$).

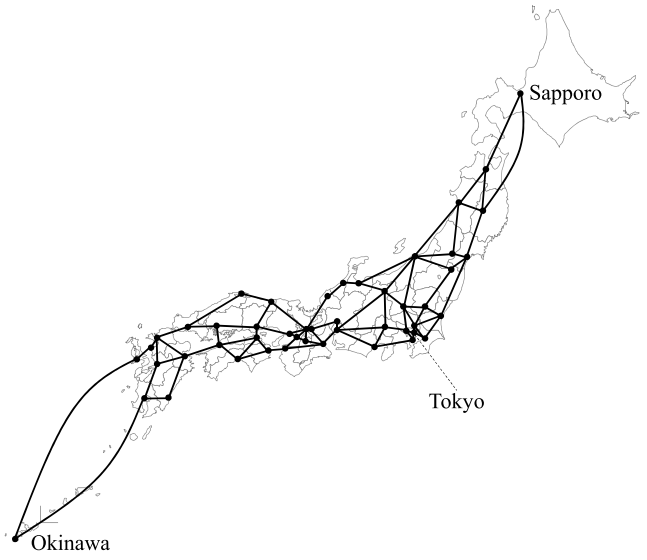

FIGURE 9. Topology of JPN48 model.

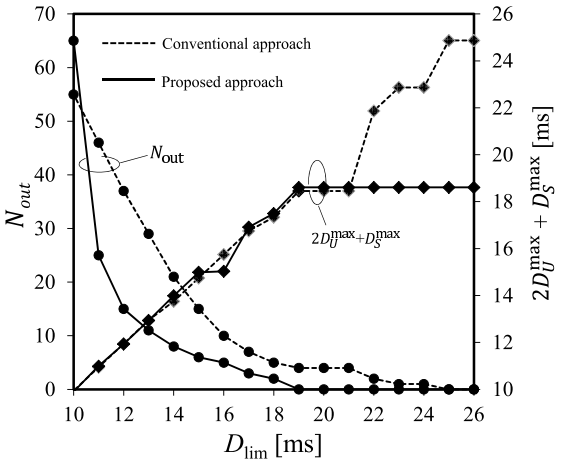

FIGURE 10. Dependency of N_{out} and $2D_U^{max} + D_S^{max}$ on admissible delay, D_{lim} for proposed and conventional approaches.

From the evaluation results, we observe that the value of $2D_U^{max} + D_S^{max}$ using the proposed approach becomes 18.6 [ms]. On the other hand, when we use the conventional centralized processing approach at Tokyo node, the value of $2D_U^{max} + D_S^{max}$ becomes 24.8 [ms], which is 25% higher than that of the proposed approach.

Note that, in JPN48 model, the solving time of ILP introduced in section IV-B1 of the proposed approach and the conventional centralized processing approach that considers the Tokyo center are 236.6 [sec] and 0.4 [sec], respectively.

From these results, we summarize that the distributed processing communication approach has an excellent property in terms of end-to-end delay in JPN48 model compared to the conventional centralized processing approach. In addition, when natural disasters, such as earthquake or tsunami, happen, the distributed processing approach can still survive by processing the application using other server located in non-disaster areas. In case of centralized processing approach, if the server is located in the disaster areas, the entire system will be collapsed.

2) PROPOSED APPROACH WITH ADMISSIBLE DELAY AND DELAY VARIATION RATE

This subsection evaluates the proposed approach considering the parameter of admissible delay and delay variation rate; the distributed server selection problem is solved by using the optimization problem introduced in section IV-B3. In this evaluation, we consider Japan Photonic Network (JPN48) model [36], as shown in Fig. 9.

Figure 10 shows the dependency of N_{out} and $2D_U^{max} + D_S^{max}$ on admissible delay, D_{lim}, for the proposed and conventional approaches. We notice that N_{out} decreases with increase in admissible delay. The value of N_{out} using the proposed approach is lower than that of using the conventional centralized processing approach. This is because the proposed approach uses distributed servers to maximize the number of in-service users. Additionally, we perceive that the values of N_{out} using both approaches are comparable, when the admissible delay is set to more than 19 [ms], where no excluded user exits. The proposed approach suppresses the delay for determining the virtual time for users' satisfaction under given D_{lim}. Furthermore, we observe that the delay for determining the virtual time using the proposed approach is lower than that of the conventional approach due to the effect of the second term of the objective functions formulated in section IV-B3.

From our observations in Fig. 10, the proposed approach is effective in terms of N_{out} when $10 \leq D_{lim} \leq 19$ [ms], and in terms of the delay, $2D_U^{max} + D_S^{max}$ when $D_{lim} > 19$ [ms], compared to the conventional approach.

Further, we discuss the performance of the proposed approach considering the admissible delay and delay variation rate. In this evaluation, the distributed server is located at the node in Kanto-region of JPN48 model. All servers are logically connected each other by the shortest route on the link of JPN48 model shown in Fig. 11. In this evaluation, 200 users are assumed, which are uniformly distributed in the participation area. The participation area is defined as X-axis longitude from 139 to 140.5 and Y-axis latitude from 35.2 to 38.8, as shown in Fig. 11.

Figure 12(a) shows the link between the server and the user, determined by the ILP introduced in section IV-B1. Figure 12 (b) shows the link between the server and the user, determined by the ILP introduced in section IV-B3 considering the delay variation at the condition of $f_{pi} = 1.5$ in the link via the Omiya node and Yokohama node. Both figures indicate that users located in the same area select a different server. When the third objective function of ILP formulated in section IV-B3 is considered (see Fig. 12(b)), we observe that users select their nearest server compared to without considering third objective function (see Fig. 12(a)). This is because the vicinity of selected servers is limited considering the maximum delay with delay variation rate.

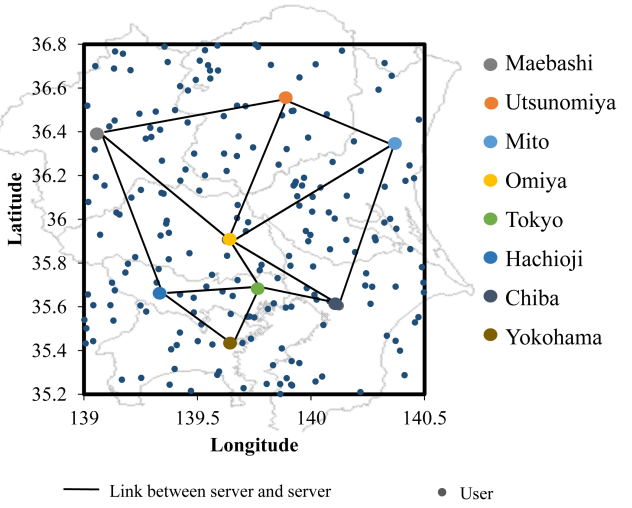

FIGURE 11. User and server location in JPN48 Kanto region.

B. SUCCESSIVE PARTICIPATION

This subsection evaluates the effect of the proposed approach considering successive participation scenario. In this evaluation, we consider that the server is located at Kanto region in JPN48 model, and 200 users are uniformly distributed in that region, as shown in Fig. 11, and these users participate in the server selection problem in a greedy manner.

Figure 13 shows the user correction time versus number of participated users for different methods. In this figure, blue and green dots are used to indicate the results of conventional centralized processing approach and the distributed server selection problem formulated in section IV-B1, respectively, where all users participate in the server selection problem simultaneously. Note that when all users participate in the server selection problem simultaneously, we named it as a simultaneous participation method. For the successive participation method, we use three different patterns of users' participation. In the random pattern, users are participated randomly in the server selection problem; all the users with different delays are uniformly distributed. We ran the simulation with 100 different seeds for the random pattern. For the user correction time of the random pattern, simulation results are obtained with a 95% confidence interval and the margin of error is within 5% of the reported average results. In patterns A and B, users are participated in descending and ascending order of their delays (d_{pi}), respectively; these delays are considered according to the distance between the user and the closest server. We observe that the user correction time for any pattern of the successive participation method is lower than that of the conventional centralized processing, when 200 users participated. In pattern A, the correction time rapidly increases in the region of small number of users, and then it slightly increases with the number of users. This is because, in pattern A, users are participated in descending order of their delays, and thus initially participated users have a significant impact on the correction time. The user correction time using pattern B constantly increases with increase

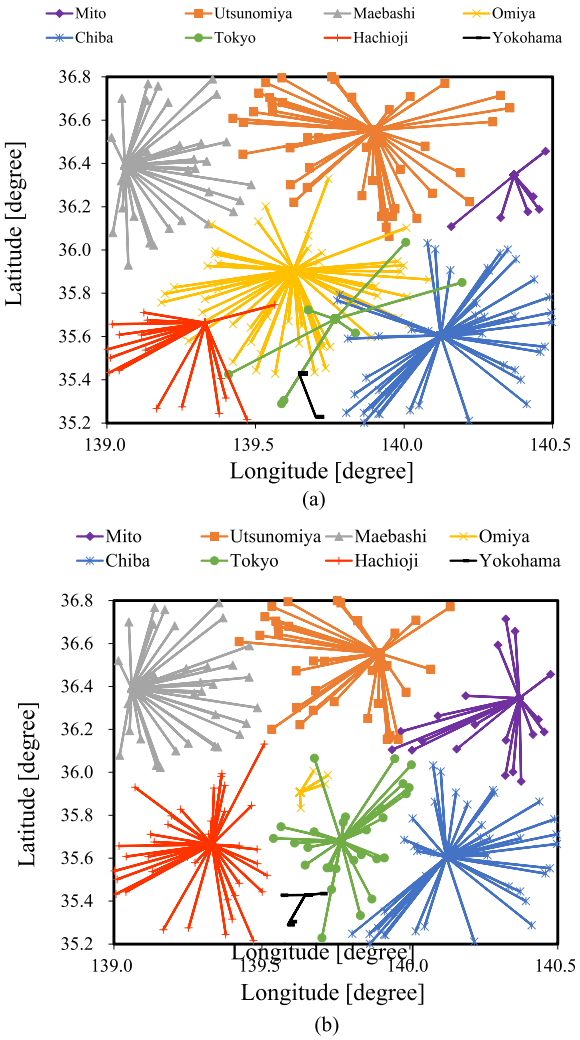

FIGURE 12. Selected servers (a) without and (b) with admissible delay and delay variation rate.

in the number of users. This is because, in pattern B, the users are participated in the server selection problem sequentially according to the increasing order of their delays. The user correction times using the random pattern is the worse among all patterns, when all 200 users were participated, as the users participated in the server selection problem randomly, without maintaining any order or sequence.

Figure 14 shows the required calculation time, which is normalized by the conventional centralized processing approach, for both simultaneous and successive user participation methods. In this evaluation, we consider two types of successive participation method; they are basic model, formulated in section IV-C1, and extended model, formulated in section IV-C2. We observe that, in simultaneous participation, the calculation time using the conventional centralized processing approach is lower than that of the proposed distributed processing approach. This is because the conventional centralized processing approach considers only one server, and hence the finding alternate servers for users is not required, which suppresses the calculation time. Fur-

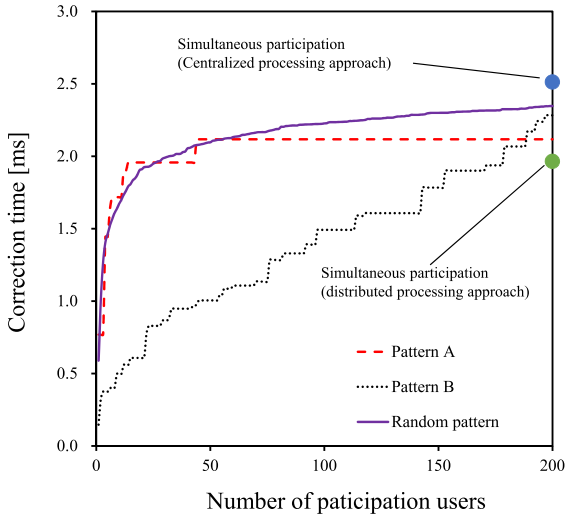

FIGURE 13. User correction time versus number of participated users for different approaches.

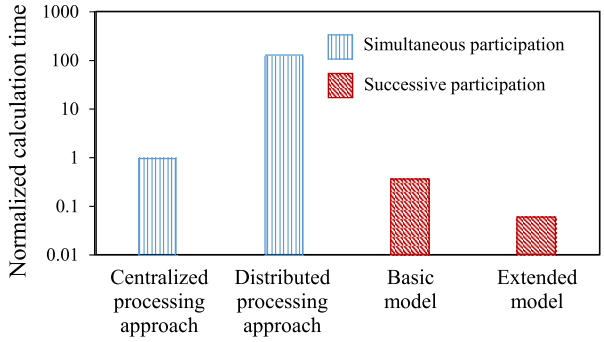

FIGURE 14. Normalized calculation times using simultaneous and successive participation methods.

ther we observe that, in the successive participation method, the calculation time using the extended model is 70% is less than that of the basic model.

VI. CONCLUSIONS AND FUTURE WORKS

This paper proposed a delay-sensitive communication approach based on distributed processing for real-time applications that provide interactive services for multiple users in order to minimize the delay considering both admissible delay and delay variation rate. The proposed approach considers two scenarios, namely simultaneous participation and successive participation. In the simultaneous participation, all users and servers are given, and the application is processed in different distributed servers; a user accesses a suitable server as a solution of the server selection problem. In the successive participation, where all servers are given, different users will be participated sequentially in a greedy manner with variation of time and without interrupting applications that are currently executing. We formulated an ILP problem in the simultaneous participation for the distributed server selection when all users and servers are given considering the parameter of admissible delay and delay-variation rate. We proved that the distributed server selection problem is NP-complete. By using a high-performance optimization solver, we solve the introduced ILP problem within a practical time for 800 users. The proposed approach was evaluated for both scenarios using different topologies and typical backbone network, namely Japan Photonic Network (JPN48) model. We observed that the proposed approach with simultaneous participation scenario outperforms the conventional centralized processing approach in terms of delay. Furthermore, we observed that the successive participation scenario with all patterns outperforms the conventional centralized processing approach in terms of delay.

This paper mainly focuses on a delay-sensitive communication approach based on distributed processing for real-time applications in order to clarify the effect of incorporation of distributed servers on overall delay. The capital and operational expenditures of the proposed scheme are not evaluated in this work, which needs further research.

REFERENCES

[1] R. Mijumbi, J. Serrat, J.-L. Gorricho, N. Bouten, F. De Turck, and R. Boutaba, "Network function virtualization: State-of-the-art and research challenges," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 236–262, 1st Quart., 2016.

[2] B. Han, V. Gopalakrishnan, L. Ji, and S. Lee, "Network function virtualization: Challenges and opportunities for innovations," *IEEE Commun. Mag.*, vol. 53, no. 2, pp. 90–97, Feb. 2015.

[3] Q. Zhang, L. Cheng, and R. Boutaba, "Cloud computing: State-of-the-art and research challenges," *J. Int. Serv. Appl.*, vol. 1, no. 1, pp. 7–18, 2010.

[4] A. Kawabata and E. Oki, "Distributed processing communication scheme for real-time application," *IEICE Trans. Commun.*, vol. J99-B, no. 4, pp. 356–367, 2016.

[5] C. Demichelis and P. Chimento, *IP Packet Delay Variation Metric for IP Performance Metrics (IPPM)*, document IETF RFC 3393, Nov. 2002.

[6] A. Kawabata, B. C. Chatterjee, and E. Oki, "Distributed processing communication scheme for real-time applications considering admissible delay," in *Proc. IEEE CQR*, Stevenson, WA, USA, May 2016, pp. 1–6.

[7] S. Ba, A. Kawabata, B. C. Chatterjee, and E. Oki, "Computational time complexity of allocation problem for distributed servers in real-time applications," in *Proc. APNOMS*, Kanazawa, Japan, Oct. 2016, pp. 1–4.

[8] M. Shahraeini, M. H. Javidi, and M. S. Ghazizadeh, "Comparison between communication infrastructures of centralized and decentralized wide area measurement systems," *IEEE Trans. Smart Grid*, vol. 2, no. 1, pp. 206–211, Mar. 2011.

[9] S. A. Barnett and G. J. Anido, "A cost comparison of distributed and centralized approaches to video-on-demand," *IEEE J. Sel. Areas Commun.*, vol. 14, no. 6, pp. 1173–1183, Aug. 1996.

[10] R. M. Fujimoto, M. Hunter, A. Biswas, M. Jackson, and S. Neal, "Power efficient distributed simulation," in *Proc. ACM SIGSIM*, Singapore, 2017, pp. 77–88.

[11] S. Bhaumik *et al.*, "CloudIQ: A framework for processing base stations in a data center," in *Proc. ACM MobiCom*, Istanbul, Turkey, 2012, pp. 125–136.

[12] Y. Jiang, "A survey of task allocation and load balancing in distributed systems," *IEEE Trans. Parallel Distrib. Syst.*, vol. 27, no. 2, pp. 585–599, Feb. 2016.

[13] G. F. Coulouris, J. Dollimore, and T. Kindberg, *Distributed Systems: Concepts and Design*. London, U.K: Pearson Education, 2005.

[14] M. D. Dikaiakos, D. Katsaros, P. Mehra, G. Pallis, and A. Vakali, "Cloud computing: Distributed Internet computing for IT and scientific research," *IEEE Internet Comput.*, vol. 13, no. 5, pp. 10–13, Sep./Oct. 2009.

[15] R. M. Fujimoto, *Parallel and Distributed Simulation Systems*. New York, NY, USA: Wiley, 2000.

[16] R. M. Fujimoto, "Research challenges in parallel and distributed simulation," *ACM Trans. Model. Comput. Simul.*, vol. 26, no. 4, pp. 22:1–22:29, 2016.

[17] D. R. Jefferson, "Virtual time," *ACM Trans. Program. Lang. Syst.*, vol. 7, no. 3, pp. 404–425, 1985.

[18] D. Cingolani, A. Pellegrini, and F. Quaglia, "Transparently mixing undo logs and software reversibility for state recovery in optimistic PDES," *ACM Trans. Model. Comput. Simul.*, vol. 27, no. 2, pp. 11:1–11:27, 2017.
[19] K. Birman, A. Schiper, and P. Stephenson, "Lightweight causal and atomic group multicast," *ACM Trans. Comput. Syst.*, vol. 9, no. 3, pp. 272–314, 1991.
[20] R. Baldoni, S. Cimmino, and C. Marchetti, "A classification of total order specifications and its application to fixed sequencer-based implementations," *J. Parallel Dist. Comput.*, vol. 66, no. 1, pp. 108–127, 2006.
[21] W. Cai *et al.*, "A survey on cloud gaming: Future of computer games," *IEEE Access*, vol. 4, pp. 7605–7620 2016.
[22] J. Smed, H. Niinisalo, and H. Hakonen, "Realizing the bullet time effect in multiplayer games with local perception filters," *Comput. Netw.*, vol. 49, no. 1, pp. 27–37, 2005.
[23] J. Xu and B. W. Wah, "Concealing network delays in delay-sensitive online interactive games based on just-noticeable differences," in *Proc. IEEE ICME*, San Jose, CA, USA, Jul. 2013, pp. 1–6.
[24] F. Arslan, "Service oriented paradigm for massive multiplayer online games," *Int. J. Soft Comput. Softw. Eng.*, vol. 2, no. 5, pp. 35–47, 2012.
[25] C. Rottondi, C. Chafe, C. Allocchio, and A. Sarti, "An overview on networked music performance technologies," *IEEE Access*, vol. 4, pp. 8823–8843, 2016.
[26] T. Kämäräinen, M. Siekkinen, Y. Xiao, and A. Ylä-Jääski, "Towards pervasive and mobile gaming with distributed cloud infrastructure," in *Proc. IEEE NetGames*, Nagoya, Japan, Dec. 2014, pp. 1–6.
[27] S. Choy, B. Wong, G. Simon, and C. Rosenberg, "A hybrid edge-cloud architecture for reducing on-demand gaming latency," *Multimedia Syst.*, vol. 20, no. 5, pp. 503–519, 2014.
[28] P. G. Lopez *et al.*, "Edge-centric computing: Vision and challenges," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 45, no. 5, pp. 37–42, 2015.
[29] W. Shi and S. Dustdar, "The promise of edge computing," *Computer*, vol. 49, no. 5, pp. 78–81, May 2016.
[30] K.-W. Lee, B.-J. Ko, and S. Calo, "Adaptive server selection for large scale interactive online games," *Comput. Netw.*, vol. 49, no. 1, pp. 84–102, 2005.
[31] Y.-R. Chen, S. Radhakrishnan, S. K. Dhall, and S. Karabuk, "Server selection with delay constraints for online games," in *Proc. IEEE GLOBECOM*, Miami, FL, USA, Dec. 2010, pp. 882–887.
[32] Y.-R. Chen, S. Radhakrishnan, S. K. Dhall, and S. Karabuk, "On the game server network selection with delay and delay variation constraints," in *Proc. IEEE COMSNETS*, Bengaluru, India, Jan. 2011, pp. 1–10.
[33] N. Takahashi, H. Tanaka, and R. Kawamura, "Analysis of process assignment in multi-tier mobile cloud computing and application to edge accelerated Web browsing," in *Proc. MobileCloud*, San Francisco, CA, USA, Mar./Apr. 2015, pp. 233–234.
[34] R. M. Karp, *Reducibility Among Combinatorial Problems*. Boston, MA, USA: Springer, 1972.
[35] *Cplex Optimizer High-performance Mathematical Programming Solver for Linear Programming, Mixed Integer Programming, and Quadratic Programming*. Accessed: Mar. 8, 2017. [Online]. Available: http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/
[36] *Japan Photonic Network Model*. Accessed: Mar. 8, 2017. [Online]. Available: http://www.ieice.org/~pn/jpn/jpnm.html

AKIO KAWABATA received the B.E., M.E., and Ph.D. degrees from The University of Electro-Communications, Tokyo, Japan, in 1991, 1993, and 2016, respectively. In 1993, he joined Nippon Telegraph and Telephone Corporation (NTT) Communication Switching Laboratories, Tokyo, Japan, where he has been involved in developing switching systems and researching network design and switching system architecture. He served as a Senior Manager of the Research and Development Department, NTT, from 2011 to 2014. He is currently a Project Manager with Network Service System Laboratories, NTT. He is also associated with the Department of Communication Engineering and Informatics, The University of Electro-Communications, for research activities.

BIJOY CHAND CHATTERJEE (M'14) received the Ph.D. degree from the Department of Computer Science and Engineering, Tezpur University, in 2014. From 2014 to 2017, he was a Post-Doctoral Researcher with the Department of Communication Engineering and Informatics, The University of Electro-Communications, Tokyo, Japan, where he was involved in researching and developing high-speed flexible optical backbone networks. He is currently with the Indraprastha Institute of Information Technology, New Delhi, as a DST Inspire Faculty and a Visiting Faculty. He has authored over 45 journal/conference papers. He has authored the book *Routing and Wavelength Assignment for WDM-based Optical Networks: Quality-of-Service and Fault Resilience* (Cham, Switzerland: Springer International Publishing, 2016). His research interests include QoS-aware protocols, cross-layer design, fog networking, optical networks, and elastic optical networks.

Dr. Chatterjee is a Life Member of IETE. He was a recipient of several prestigious awards, including the DST Inspire Faculty Award in 2017, the ERCIM Postdoctoral Research Fellowship by the European Research Consortium for Informatics and Mathematics in 2016, the UEC Postdoctoral Research Fellowship from The University of Electro-Communications in 2014, and the IETE Research Fellowship from the Institution of Electronics and Telecommunication Engineers, India, in 2011.

SEYDOU BA received the M.E. and Ph.D. degrees in information and communication engineering from The University of Electro-Communications, Tokyo, Japan, in 2014 and 2017, respectively. He is currently with the NEC-AIST AI Cooperative Research Laboratory, National Institute of Advanced Industrial Science and Technology. His research interests include artificial intelligence, reinforcement learning, elastic optical networks, and software defined networks.

EIJI OKI (M'95–SM'05–F'13) received the B.E. and M.E. degrees in instrumentation engineering and the Ph.D. degree in electrical engineering from Keio University, Yokohama, Japan, in 1991, 1993, and 1999, respectively. In 1993, he joined Nippon Telegraph and Telephone Corporation (NTT) Communication Switching Laboratories, Tokyo, Japan. He has been researching network design and control, traffic-control methods, and high-speed switching systems. From 2000 to 2001, he was a Visiting Scholar with the Polytechnic Institute of New York University, Brooklyn, NY, where he was involved in designing terabit switch/router systems. He was involved in researching and developing high-speed optical IP backbone networks with NTT Laboratories. He was with The University of Electro-Communications, Tokyo, from 2008 to 2017. He joined Kyoto University, Kyoto, Japan, in 2017, where he is currently a Professor. He has been active in the standardization of the path computation element and GMPLS in the IETF. He has authored over ten IETF RFCs. He has authored or co-authored four books, *Broadband Packet Switching Technologies* (New York, NY, USA: Wiley, 2001), *GMPLS Technologies* (Boca Raton, FL, USA: CRC Press, 2005), *Advanced Internet Protocols, Services, and Applications* (New York, NY, USA: Wiley, 2012), and *Linear Programming and Algorithms for Communication Networks* (Boca Raton, FL, USA: CRC Press, 2012).

Prof. Oki is a fellow of IEICE. He was a recipient of several prestigious awards, including the 1998 Switching System Research Award and the 1999 Excellent Paper Award presented by IEICE, the 2001 Asia-Pacific Outstanding Young Researcher Award presented by the IEEE Communications Society for his contributions to broadband network, ATM, and optical IP technologies, the 2010 Telecom System Technology Prize by the Telecommunications Advanced Foundation, the IEEE HPSR 2012 Outstanding Paper Award, the IEEE HPSR 2014 Best Paper Award Finalist, First Runner Up, and the IEEE Globcom 2015 Best Paper Award.