PAPER

A Network Design Scheme in Delay Sensitive Monitoring Services

Akio KAWABATA[†a)], ***Senior Member***, **Takuya TOJO**[††], ***Member***, **Bijoy CHAND CHATTERJEE**[†††], ***Nonmember***, ***and*** **Eiji OKI**[††††], ***Fellow***

SUMMARY Mission-critical monitoring services, such as finding criminals with a monitoring camera, require rapid detection of newly updated data, where suppressing delay is desirable. Taking this direction, this paper proposes a network design scheme to minimize this delay for monitoring services that consist of Internet-of-Things (IoT) devices located at terminal endpoints (TEs), databases (DB), and applications (APLs). The proposed scheme determines the allocation of DB and APLs and the selection of the server to which TE belongs. DB and APL are allocated on an optimal server from multiple servers in the network. We formulate the proposed network design scheme as an integer linear programming problem. The delay reduction effect of the proposed scheme is evaluated under two network topologies and a monitoring camera system network. In the two network topologies, the delays of the proposed scheme are 78 and 80 percent, compared to that of the conventional scheme. In the monitoring camera system network, the delay of the proposed scheme is 77 percent compared to that of the conventional scheme. These results indicate that the proposed scheme reduces the delay compared to the conventional scheme where APLs are located near TEs. The computation time of the proposed scheme is acceptable for the design phase before the service is launched. The proposed scheme can contribute to a network design that detects newly added objects quickly in the monitoring services.

key words: *IoT service, network design, edge computing, data analysis, low-delay network*

1. Introduction

Internet-of-Things (IoT)-based systems provide various services via a wide-area network using cloud services [1], [2], advancing edge computing technology [3], and fifth-generation (5G) services [4]. One of the typical IoT services is provided by a monitoring system that processes two types of data (reference data and monitoring data) to determine actions to be taken. The reference data is usually stored such as database (DB) and used by the application. The sensor collects the monitoring data to be used by the application. For example, cameras are used to detect specific people or vehicles [5] or to detect noise to monitor the normality of factory production lines [6]. Other examples include a growth management system in which an optimal growth condition is determined from the reference data such as air temperature, weather, etc., and the monitoring data from farm sensors. Also, in the monitoring systems that require decisions based on the latest data, such as the detection of fleeing criminals by surveillance cameras or automated driving, the application must process the latest reference data as new data is added.

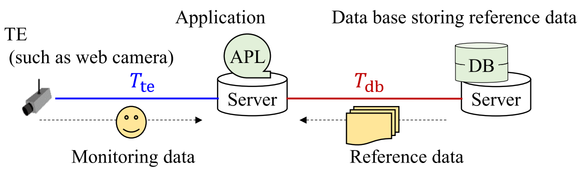
When the service is provided over a wide area network, the reference data is typically located at the center of the network, and the applications are typically located at the edge of the network [7], [8]. Sensors are generally distributed over a wide area as terminals of a network. To detect the latest data quickly, it is required to receive the reference data and monitoring data at the application quickly and accelerate the application processing. Either the computer's performance needs to be improved or the computing resources should be reserved to accelerate the application processing. The server performance can be controlled by the service provider. The processing delay may be reduced with sufficient server resources. On the other hand, the transmission distance needs to be reduced to suppress delay. Since the route of optical fiber cannot be changed immediately, it is necessary to design a network with a short transmission distance by devising the allocation of application (APL) and DB. For example, there is about 2000 km from one end to the other in Japan, and the transmission delay of optical fiber alone is equivalent to a delay of 10 ms. The impact of delays is even greater when the country is as large as the United States or China. This transmission delay may not be negligible in mission-critical applications.

When reducing transmission delay to detect newly added reference data quickly, it is necessary to focus on the transmission route of the reference and monitoring data. Fig. 1(a) shows the data exchange of the reference and monitoring data among DB, APL, and IoT devices located at terminal endpoints (TEs). TEs indicate sensors, such as monitoring cameras or in-vehicle global positioning systems (GPS), and IoT devices, like sprinklers at farms or automobile navigation systems. DB indicates a centralized data source that stores a large number of reference data, such as personal data of fleeing criminals or location data of disasters. Usually, DB and APL are allocated to servers with CPU, memory, and storage. The monitor data collected at TE and the updated reference data at DB are sent to APL. In Fig. 1(a), $T_{\rm te}$ indicates a delay until the monitoring data

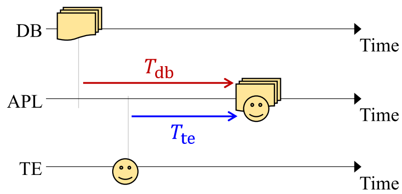
Manuscript received January 19, 2023.
Manuscript revised March 15, 2023.
Manuscript publicized April 19, 2023.

[†]The author is with the Toyohashi University of Technology, Toyohashi-shi, 441-8580 Japan.
[††]The author is with the NTT Network Service Systems Laboratories, Musashino-shi, 180-8585 Japan.
[†††]The author is with the South Asian University, New Delhi, India.
[††††]The author is with the Kyoto University, Kyoto-shi, 606-8501 Japan.
a) E-mail: kawabata.akio.oc@tut.jp

DOI: 10.1587/transcom.2023EBP3010



(a) Data route of reference data and monitoring data



(b) Time from reference data update to the start processing

Fig. 1 Route of reference data and monitoring data.

arrives at APL, and T_{db} indicates a delay until the updated reference data arrives at APL. The delay between the latest update of reference data and the start of processing at APL is expressed as $\max(T_{te}, T_{db})$. Therefore, to reduce the delay in detecting newly added reference data, it is necessary to reduce $\max(T_{te}, T_{db})$.

The delay in the carrier network, i.e., the delay between TEs on both ends, consists of the queuing delay of network equipment and the transmitting delay of optical fibers. In this paper, we assume TE and IoT devices are co-located and connected by a fixed-line. In some use cases, TE and IoT devices may be connected through wireless. When the delay between TE and IoT devices is affected by wireless conditions, we target only use cases to prevent delay degradation by managing power supply status, the number of connections, etc. Delay-sensitive mission-critical services are usually provided by networks whose delay is guaranteed by service level agreements (SLAs). Network equipment with sufficient resources can control traffic congestion and queuing delays. However, the transmitting delay of an optical fiber cannot be reduced unless the transmission distance is shortened. This transmitting delay depends on the network topology among DB, APLs, and TEs. A network design scheme that reduces the transmission distances of the reference data and monitoring data is required to reduce the transmission delay of an optical fiber.

To address the above issue, this paper proposes a network design scheme to minimize the transmitting delay. In the proposed scheme, the network design selects the servers to which DB and APL are allocated from multiple candidate servers and determines the server to which TE belongs. We tend to think that the case where TE belongs to the nearest server is the one with the minimum delay. However, a network in which TEs belong to the nearest server (conventional scheme) may not have a minimum delay. The proposed scheme minimizes the maximum transmission delay.

In the proposed scheme, the network design scheme is formulated as an integer linear programming problem (ILP) that minimizes $\max(T_{te}, T_{db})$. We express the value of $\max(T_{te}, T_{db})$ as T_{delay}. We evaluate the proposed scheme in terms of delay for two basic network topologies and an assumed monitoring camera system network located at stations of Super Express (Shinkansen) in Japan. In basic network topologies, we assume that the servers are located at the node locations of COST239 [9] and JPN Kanto region [10]. TEs are uniformly distributed in the square area, including each node. Numerical results observe that the proposed scheme reduces T_{delay} compared to the conventional scheme in which TEs select the nearest server. We also evaluate the computation time of solving the proposed scheme. These results indicate that the computation times are acceptable for a network design process before launching the service.

This paper is an extended version of the work [11]. The main additions are as follows. We present related works and the originality of our work. We extensively investigate the dependency of computation time on the number of TEs and servers and provide the delay and computation time with an assumed monitoring camera system network which is more similar to an actual service.

The rest of the paper is organized as follows. Section 2 presents the related works and the position of work. The system model, effective use-case, and formulation for the proposed scheme are presented in Sect. 4. In Sect. 5, we evaluate the proposed scheme in terms of delay and computation time and compare it to the conventional scheme using typical network topologies (COST239 and JPN) and a network of the assumed monitoring camera system network. Finally, Sect. 6 concludes the paper.

2. Related Works

Several types of research works have been studied on delay reduction in IoT services using multiple clouds via a wide-area network [12]–[18]. These works are based on the distributed allocation of functions at multiple clouds and focus on controlling the transmitting and processing delays.

Distributed processing environments, such as the edge and Fog cloud, are used in event detection and anomaly detection systems, and research is being conducted to optimize the amount of computation and processing delays [12]. A partitioning model of the data stream was presented and shown to enable highly parallel processing with low delay by limiting the buffering imposed on event detection when there are dynamic variations in load [13]. A big data platform for low-delay traffic flow analysis based on real-time data was launched, and low-delay detection and classification of events at highways using the big data platform was reported.

In distributed processing, a method for satisfying delay constraints by managing tradeoffs between delay and resources of computation and network was reported in [16]. Distributed processing systems that offload processing to reduce delay face the issue of not being able to process heavy tasks due to a lack of available resources or limited computing power. To address this issue, an optimized offload solution for efficiently allocating in the case of resource contention was presented in; this proposal has shown its potential to reduce the average delay in heterogeneous clouds [17]. The challenge of distributed processing in real-time IoT applications is using limited resources to control processing and queuing delays effectively. In [18], a two-stage tandem queuing model was developed to predict an average end-to-end delay in two classes of networks, which are edge sensor networks (ESNs) and data center networks (DCNs). This effort can be assumed to predict queuing delays for conformance and monitoring for delay-sensitive IoT applications.

In an IoT-based camera analysis system, the work in [19] presented the smart cameras and base station task offloading scheme, and the uplink-downlink bandwidth allocation are jointly optimized to minimize the delay. The works in [20] presented an approach that distributes tasks to edge servers based on estimated waiting time (queuing delay and processing delay) to minimize the delay. The work in [21] presented that an end-to-end delay is minimized by distributing jobs and communicating with each other among distributed edges. Q. Fan et al. [22] introduced a workload allocation scheme to minimize the response time of applications, which considers both the network delay and computing delay. The work in [23] presented a service chaining scheme to minimize service time by optimizing the number of clusters in the multiple clustered edge cloud. The work in [24] presented a workload allocation scheme to guarantee power consumption and delay for the balance between delay and power consumption.

The above works target the suppression of processing and queuing delays and the optimal offloading in processing systems. For mission-critical and delay-sensitive applications, it is usual to provide using a delay-guaranteed network and sufficient processing power. There is no proposal for a network design scheme focusing on transmission delay, which is the target of this paper. With telecommunication carriers providing selectable multiple transmission routes and cloud providers offering selectable multiple cloud locations, IoT application providers need to design a low-delay network from multiple servers and transmission paths. The proposed scheme is positioned as a solution for this network design issue used by IoT service providers.

3. Motivation and Prerequisite

3.1 Motivation of Network Topology Design

In this section, we discuss the necessity of a network design scheme for reducing the delay. Figure 2 shows an example topology using TEs and servers, where Server B is assumed as a central cloud server. Figure 3 shows two examples of the network configuration. DB is allocated on server B in both configurations. In configuration A, APLs are allocated to the servers nearest TE. APLs are allocated to servers A and C, since TEs 1 and 2 select nearest server A, and TEs 3 and 4 select nearest server C. T_{delay} of configuration A is 10 seconds. In configuration B, the servers that allocate APLs and connect TEs are determined to minimize T_{delay}. On one hand, if

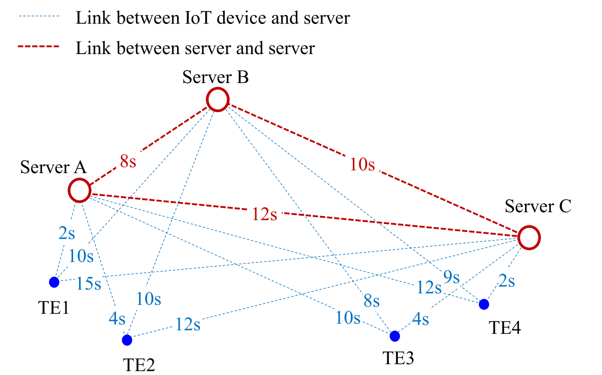


Fig. 2 Example topology using TEs and servers; each number attached to a link indicates a delay associated with that link.

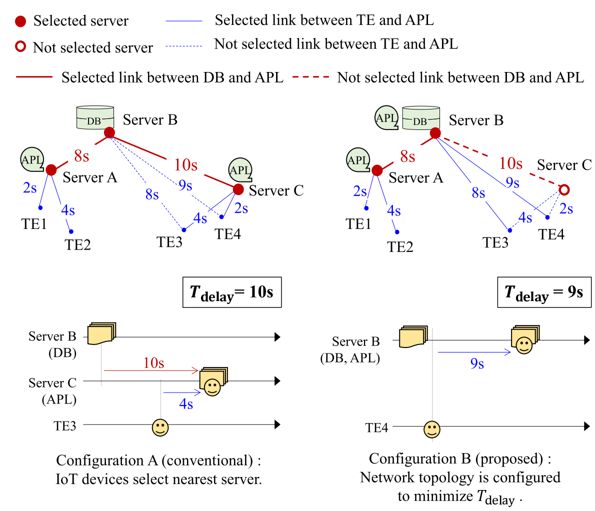


Fig. 3 Examples of network configuration of DB, APLs, and TEs.

TEs 3 and 4 select server C, T_{delay} is 10 seconds, since $T_{\text{te}}^{\text{max}}$ and $T_{\text{db}}^{\text{max}}$ are four and 10 seconds, respectively. $T_{\text{te}}^{\text{max}}$, which is four seconds, is determined by the link between TE 2-server A and the link between TE 3-server C. T_{delay}, which is 10 seconds, is determined by the link between server B-server C. On the other hand, if TEs 3 and 4 select server B, T_{delay} is nine seconds, since $T_{\text{te}}^{\text{max}}$ and $T_{\text{db}}^{\text{max}}$ are nine and eight seconds, respectively. $T_{\text{te}}^{\text{max}}$, which is nine seconds, is determined by the link between TE 4-server B. $T_{\text{te}}^{\text{max}}$, which is eight seconds, is determined by the link between server A-server B. Therefore, APLs are allocated to servers A and B, since TEs 1 and 2 select the nearest server A and TEs 3 and 4 select server B which is not the nearest server. T_{delay} of configuration B is nine seconds. From the above examples, we can summarize that locating APLs on servers near TEs does not guarantee reducing delay. Thus, an effective network design scheme is required for minimizing T_{delay}.

3.2 Prerequisites of Data Exchange for Processing Model

We discuss the prerequisites of the processing model for the monitoring system focusing on a data flow. Various IoT architectures have been presented, including four-layer [25] and five-layer models [26], [27]. In the commonality of these models, there are an input/output (I/O) device layer and an application layer offloaded from the cloud [12], [25]–[27]. It effectively reduces delay and bandwidth consumption in the network by offloading applications from the cloud to the edge. When determining an optimal control from a variety of information obtained by various sensors, such as vehicle traffic optimization or smart home, there may be no DB with reference data in the systems. On the other hand, in the monitoring systems, such as person detection of criminals or detection of abnormal noise in factories, there is usually a DB with reference data to be collated with data from sensors. In many cases, Message Queue Telemetry Transport (MQTT) [28] is adopted as a protocol for data exchange [25], and an asynchronous one-to-many publish/subscribe type communication is supported. Considering these conditions, as shown in Fig. 1, we assume that TEs as sensors invariably send monitoring data to APL and that the updated reference data at DB is sent to APL in a push type.

4. Proposed Scheme

4.1 Overview

The proposed scheme is intended to design a low-delay network considering the reference data and monitoring data. The maximum value of the transmission delay between DB and APL among all selected links is expressed as $T_{\mathrm{db}}^{\max}$. The maximum value of the transmission delay between APLs and TEs among all selected links is expressed as $T_{\mathrm{te}}^{\max}$. As shown in Fig. 1(b), the delay can be expressed as the larger value of $T_{\mathrm{te}}^{\max}$ and $T_{\mathrm{db}}^{\max}$. If $T_{\mathrm{db}}^{\max}$ is larger than $T_{\mathrm{te}}^{\max}$, the detecting using the newly added data starts $T_{\mathrm{db}}^{\max}$ after updating DB. If $T_{\mathrm{te}}^{\max}$ is larger than $T_{\mathrm{db}}^{\max}$, the detecting using the newly added data starts $T_{\mathrm{te}}^{\max}$ after the latest monitoring at TE. The delay between the latest update of the reference data and the start of detecting a newly added reference data, T_{delay}, is expressed as $\max(T_{\mathrm{te}}^{\max}, T_{\mathrm{db}}^{\max})$.

We set the above delays so that they are independent of how the equipment is set up or how much traffic there is. Since the proposed scheme targets networks for delay-sensitive services, we assume a network with guaranteed delay. Communication carriers can control the equipment resources in the network and guarantee SLAs [29], [30]. In other words, carriers know how long it takes to get from one user-network interface in the network to another and manage their equipment resources to meet the standards set by SLAs. We assume that transmission delays are measured in advance, such as round-trip time (RTT) measurement with ping [31], transmission delay calculation using packet timestamps of network time protocol (NTP) [32] or precision time protocol (PTP) [33], in various situations including congestion and non-congestion periods. These transmission delays typically incorporate the queuing delay in network equipment. With these assumptions, we achieve a network design of the logical topology of TEs and servers; each delay of a logical link in the proposed scheme reflects the network equipment configuration and traffic conditions.

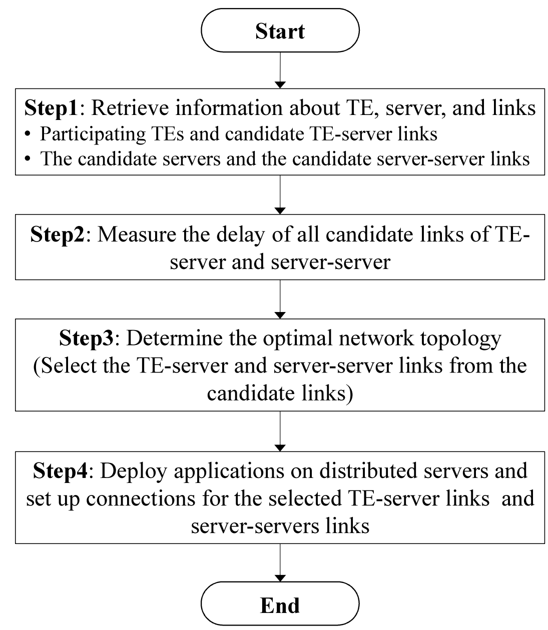


Fig. 4 Assumed network design steps before launching an application.

As shown in Fig. 4, we assume four steps in network design before launching an application. Step 1 is retrieving information about the TE, server, and links. The information consists of the participating TEs, the candidate TE-server links, the candidate servers, and the candidate server-server links, etc. Step 1 is outside the scope of this study, as it completely depends on the service provider's operation. Note that the time taken for Step 1 may not be the dominant time since the service provider retrieves its own customer and network information held by itself. Step 2 is measuring the delay of all candidate links of TE-server and server-server. If the delay of all links is contained in the information of Step 1, Step 2 can be skipped. The number of these measurements increases linearly with the number of links. Normally, when measuring RTT using ping [31], one measurement can be completed in ms order or less in the managed network in Japan. If the measurement time is 100 ms for one measurement, the measurement time of 1000 links will be completed in about 100 seconds, so it will not be the dominant time in network design. Furthermore, if the measurements can be performed in parallel, the time is expected to be reduced. Step 3 is determining the optimal network topology. The proposed scheme targets Step 3. It is required that the optimal design is completed in an appropriate amount of time allowed before launching the service. Step 4 is deploying applications on distributed servers and setting up connections for the selected TE-server and server-servers links. The time required for Step 4 is outside the scope of this study, as it completely depends on the service provider's operation. Note that Step 4 can process in parallel since all selected links and servers to use have been determined in Step 3.

4.2 Formulation for Proposed Scheme

In this section, we formulate the proposed network design scheme as an ILP problem to minimize the delay. The network is modeled as an undirected graph $G(V,E)$, where V and E indicate sets of nodes and non-directional links, respectively. The set of TEs is represented by $V_T \subseteq V$. The set of servers is represented by $V_S \subseteq V$. $V_T \cup V_S = V$ since a node is either a TE or a server, and $V_T \cap V_S = \emptyset$ since no node is both a TE and a server. The set of links between TE and server is represented by $E_T \subseteq E$; a link between TE $t \in V_T$ and server $i \in V_S$ is symbolized by $(t,i) \in E_T$. The set of links between server and server is represented by $E_S \subseteq E$; a link between server $i \in V_S$ and server $j \in V_S$ is symbolized by $(i,j) \in E_S$. $E_T \cup E_S = E$ since a link is either a link between TE and server or a link between server and server, and $E_T \cap E_S = \emptyset$ since no link is both a link between TE and server and a link between server and server. The delay of link $(t,i) \in E_T$ is represented by d_{ti}. The delay of link $(i,j) \in E_S$ is denoted by d_{ij}.

The maximum number of TEs that accommodate server $i \in V_S$ is denoted by M_i. K_i is a binary parameter for $i \in V_S$, where K_i=1 if server i can be allocated DB, and K_i=0 otherwise. x_{kl} is a binary variable for $(k,l) \in E$, where x_{kl}=1 if link (k,l) is selected, and x_{kl}=0 otherwise. y_i is a binary variable for $i \in V_S$, where y_i=1 if sever i is allocated APL, and y_i=0 otherwise. z_j is a binary variable for $j \in V_S$, where z_j=1 if sever i is allocated DB, and z_j=0 otherwise. The maximum delay of a link between TEs $t \in V_T$ and server $i \in V_S$ that accommodates an application is denoted by $T_{te}^{max} = \max_{(t,i)\in E_T}(d_{ti} \cdot x_{ti})$. The maximum delay of a link between server i that accommodates an application and server j that has the database is denoted by $T_{db}^{max} = \max_{(i,j)\in E_S}(d_{ij} \cdot x_{ij})$. T_{delay} is denoted by $T_{delay} = \max(T_{te}^{max}, T_{db}^{max})$ as the objective function.

DB and APL allocation problem in the proposed scheme is formulated as an ILP problem by:

$$\text{Objective} \quad \min \quad T_{delay} \tag{1a}$$

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

$$\sum_{t\in V_T} x_{ti} \leq M_i, \forall i \in V_S \tag{1c}$$

$$\sum_{i\in V_S} z_i = 1 \tag{1d}$$

$$z_i \leq K_i, \forall i \in V_S \tag{1e}$$

$$y_i \geq x_{ti}, \forall i \in V_S, (t,i) \in E_T \tag{1f}$$

$$d_{ti} x_{ti} \leq T_{te}^{max}, \forall (t,i) \in E_T \tag{1g}$$

$$d_{ij} x_{ij} \leq T_{db}^{max}, \forall (i,j) \in E_S \tag{1h}$$

$$T_{te}^{max} \leq T_{delay}, \forall (t,i) \in E_T \tag{1i}$$

$$T_{db}^{max} \leq T_{delay}, \forall (i,j) \in E_S \tag{1j}$$

$$x_{ij} \geq z_j + y_i - 1, \forall (i,j) \in E_S \tag{1k}$$

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

Table 1 Given parameters and decision variables.

Given parameters	d_{ti}	Delay between TE t and server i
	d_{ij}	Delay between server i and server j
	M_i	Maximum number of TEs who belong to server i
	K_i	Server i can be allocated DB or otherwise
Decision variables	T_{delay}	Delay between latest update of reference data and the start of processing updated data
	T_{te}^{max}	Maximum delay between TEs and APLs
	T_{db}^{max}	Maximum delay between DB and APLs
	x_{kl}	Link between node k and l is selected or otherwise
	y_i	Server i is selected or otherwise
	z_i	Server i is allocated DB or otherwise

$$x_{ij} \leq z_j, \forall (i,j) \in E_S \tag{1m}$$

$$x_{kl} \in \{0,1\}, \forall (k,l) \in E \tag{1n}$$

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

$$z_i \in \{0,1\}, \forall i \in V_S. \tag{1p}$$

Table 1 shows the given parameters and decision variables. Equation (1a) expresses the objective function, which minimizes T_{delay}. Equation (1b) represents that one link is selected between TE and server. Equation (1c) represents that the sum of the number of TEs that select server $i \in V_S$ does not exceed M_i. Equation (1d) represents that one DB can be allocated to the network. Equation (1e) represents that DB is allocated to servers i, i.e., K_i=1. Equation (1f) represents that APL is allocated to a server, if the server is selected by at least one TE. Equation (1g) expresses that the maximum delay of a link between TE and APL is T_{te}^{max}. Equation (1h) expresses that the maximum delay of the selected link between APL and DB is T_{db}^{max}. Equation (1i) expresses that the maximum delay of the selected link between TE and APL is T_{delay}. Equation (1j) expresses that the maximum value of the selected link between APL and DB is T_{delay}. Equations (1k)–(1m) indicate that $y_i \cdot z_j = x_{ij}$ is satisfied at sever $i \in V_S$, server $j \in V_S$, and link $(i,j) \in E_S$. Equations (1n)–(1p) state that x_{kl}, y_i, and z_i are binary variables.

5. Evaluation

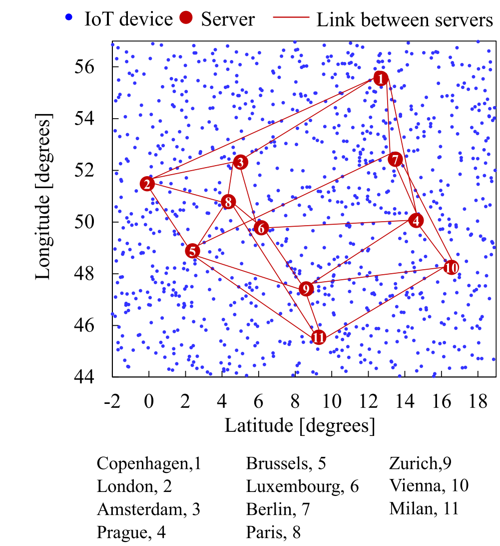
We assess the performance of the proposed scheme in terms of delay and computation time, and compare it with the conventional scheme. Each TE can select any server under the restriction. This assumption is based on the following conditions. TE accommodates an exchange office of the network via an optical fiber or Ethernet cables and connects to the distributed server via a carrier network. Any logical connection between TE and the server is possible in the carrier network. Considering these conditions, we simply assume that TE can select any server.

In a commercial network, the transmission distance is not proportional to the delay because traffic passes through transport devices, routers, and switches. Note that, typically, the longer the transmission distance, the greater the number of devices passed through; for this reason, we assume that the delay is proportional to the distance in this evaluation. When a service provider uses the proposed scheme, the delay should be determined according to the last paragraph of Sect. 4.1. Based on this assumption, we treat a 100 km transmission delay as a 0.5 ms delay in each link. Computation time is the average time considered by solving the ILP problem five times. T_{delay} of the conventional scheme is obtained by using the ILP problem in (A· 1a)–(A· 1b) at Appendix. We solve the ILP problem mentioned in (1a)–(1p) using Solving Constraint Integer Programs (SCIP) [34]; the computer is configured with the Intel(R) Xeon(R) Gold6132 CPU 2.60 GHz, and 196 GB memory is used.

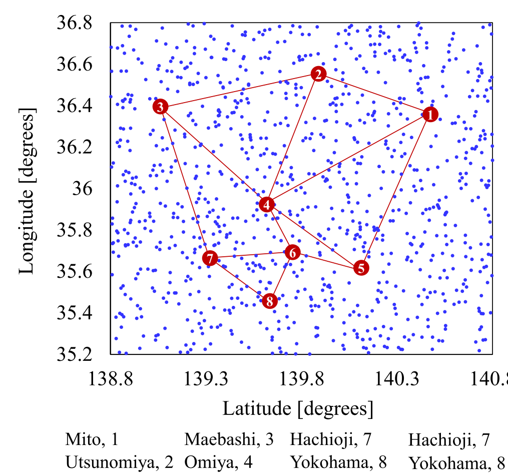
5.1 Basic Performance for Typical Network

We evaluate the proposed scheme with two different topologies and observe the topology dependency. We assume that the servers are located at the nodes of COST239 [9] and JPN Kanto region [10], as shown in Fig. 5. The link distance in COST239 and JPN Kanto region is treated as a linear distance on the coordinate axes based on the latitude and longitude of each node. Logically, all servers can communicate with each other by links of their shortest routes. For example, London is connected to Luxembourg via Brussels. The delay of a link between a TE and a server is treated as a linear distance on the coordinate axes based on the latitude and longitude. As the delay of the link, one unit of the coordinate axes is treated as 100 km. We assume that groups of 1000 TEs are uniformly distributed in each square area, as shown in Fig. 5. For COST239, we set V_{T}=1000, V_{S}=11, E_{T}=11000, and E_{S}=55. For JPN, we set V_{T}= 1000, and V_{S} = 8. We set E_{T}=8000, V_{S}=8, and E_{S}=28.

We discuss the delay performance of the proposed and conventional schemes using the two topologies. Figures 6(a) and (b) show T_{delay} of the proposed and conventional schemes with M_i=1000, where $i \in V_{\mathrm{S}}$. The red bars indicate the value of the conventional scheme, and the number at the horizontal axis indicates the server location where DB is allocated; the mapping of the server number and its location is mentioned in Fig. 5. For example, the first red bar (sever 1) indicates T_{delay} at the condition that DB is allocated to the server at Copenhagen node. From Fig. 6(a), we observe that T_{delay} using the proposed scheme is smaller than that of the conventional scheme; DB is allocated to any server in the conventional scheme. T_{delay} of the proposed scheme is 78 percent compared to that of the conventional scheme for server 3 in COST239. As shown in Fig. 6(b), T_{delay} of the proposed scheme is 80 percent compared to that of the conventional scheme for server 5 in JPN. The delay is reduced because each TE selects the optimal APL server considering the delay between DB and APL. In the conventional scheme, TE selects the nearest server as APL without considering the delay between DB and APL. These results indicate that the proposed scheme reduces the delay regardless of the network topology of servers. These results indicate that, as shown in



(a) Location of servers and IoT devices in COST239



(b) Location of servers and IoT devices in JPN

Fig. 5 Location of servers and TEs.

Figs. 3(a) and (b), there exists a server selection that reduces the delay rather than selecting the nearest server, and the network design using the proposed scheme achieves a server selection that minimizes the delay.

We discuss the results of the delay performance with M_i limitation. We choose two conditions for M_i limitation, considering the value of the total accommodatable number of TEs by all servers, $\sum_{i \in V_{\mathrm{S}}} M_i$. In the first condition, $\sum_{i \in V_{\mathrm{S}}} M_i$ is 1001 for COST239, where $M_i = 91$, and 1000 for JPN, where $M_i = 125$. In the second condition, $\sum_{i \in V_{\mathrm{S}}} M_i$ is 1320 for COST239, where $M_i = 120$, and 1600 for JPN, where $M_i = 200$. In the first condition, $\sum_{i \in V_{\mathrm{S}}} M_i$ is approximately equal to the total number of TEs. The second condition is more relaxed than the first one in terms of M_i. Figures 7(a) and (b) show T_{delay} of the proposed and conventional schemes with M_i limitation for COST239 and JPN, respectively, where $i \in V_{\mathrm{S}}$. Tables 2 and 3 show the servers where DB and APL are allocated in the proposed and conventional schemes. T_{delay} of the proposed scheme in Fig. 7(a)

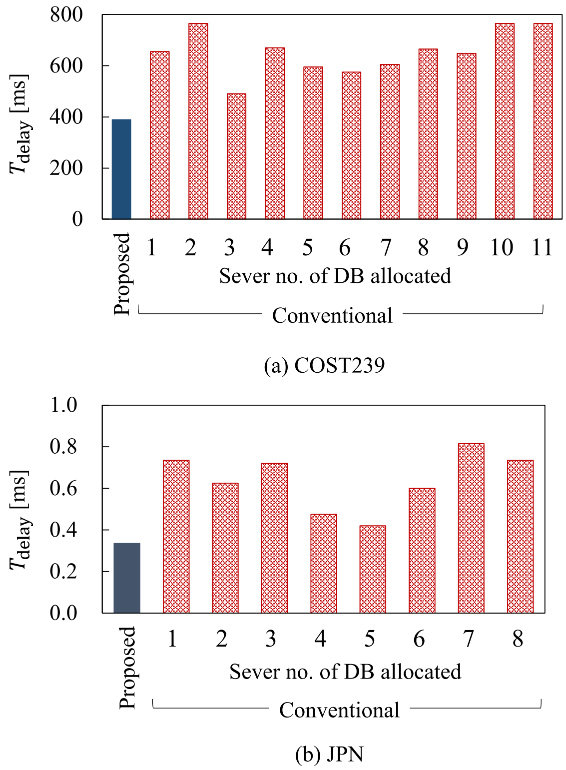


Fig. 6 T_{delay} using proposed and conventional schemes.

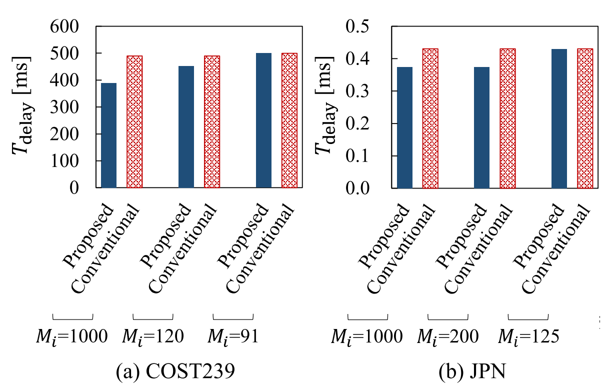


Fig. 7 T_{delay} with M_i limitation using proposed and conventional schemes, where $i \in V_S$.

increases due to a decrease in M_i, but that of the conventional scheme does not increase. This is because the number of TEs per server is averaged since TE selects servers near TEs in the conventional scheme. In Fig. 7(a), T_{delay} of the proposed scheme is reduced with $M_i = 1000, 120$ compared to that of the conventional scheme. As shown in Fig. 7(b), the delay reduction of the proposed scheme is also achieved in JPN. As shown in M_i=91 of Fig. 7(a) and M_i=125 of Fig. 7(b), when M_i constraints become stronger, the selectable server will be limited and the merits of the proposed scheme will decrease. These results indicate that the proposed scheme is effective for reducing T_{delay} under the condition of lenient M_i limitation, regardless of the server topology. The reason for

Table 2 Allocation of DB and APL using proposed and conventional schemes in COST239.

M_i	Proposed		Conventional	
	APL	DB	APL	DB
1000	2, 3, 4, 5, 6 8, 9	6	1, 2, 3, 5, 7 8, 9, 10	3
120	3, 4, 5, 6, 7 8, 9, 10, 11	9	1, 2, 3, 5, 6 7, 8, 9, 10, 11	3
91	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11	3	1, 2, 3, 4, 5 6, 7, 8, 9, 10, 11	3

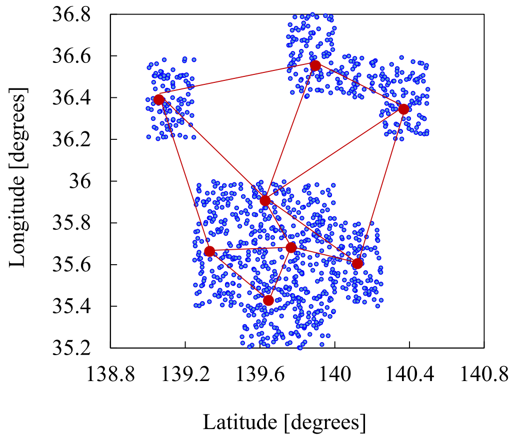
Table 3 Allocation of DB and APL using proposed and conventional schemes in JPN.

M_i	Proposed		Conventional	
	APL	DB	APL	DB
1000	1,4	2	1, 2, 4, 5, 6, 8	5
200	1, 4, 5, 6, 7, 8	5	1, 2, 3, 4, 5 6, 7, 8	4
125	1, 2, 3, 4, 5 6, 7, 8	4	1, 2, 3, 4, 5 6, 7, 8	4

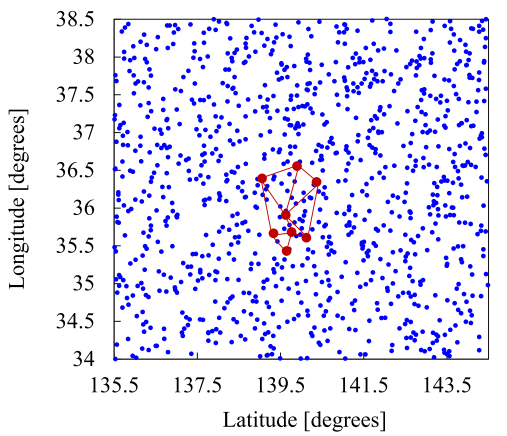
these results is that under the lenient M_i limitation, there are many candidates of the sever selection that reduces the delay rather than selecting the nearest server. The service provider usually has the server resources that process the application without delay. However, if the service provider recognizes the delay degradation caused by M_i limitation during the network design phase, it can use the proposed scheme to augment server resources and reduce the delay.

We discuss the effective case of the proposed scheme from the perspective of TE distribution. Since $T_{\text{te}}^{\max}$ is minimized in the conventional scheme, the effective case of the proposed scheme needs to satisfy the condition that T_{delay} is determined by $T_{\text{db}}^{\max}$. Figures 8(a) and (b) show two different types of TE distribution. In Type-A, 1000 TEs are located near the servers. In Type-B, TEs are located far from servers. Figure 9 shows the ratio of the proposed and conventional schemes for T_{delay} at three types of TE distribution: (i) Type-A in Fig. 8, (ii) Fig. 5(b), and (iii) Type-B in Fig. 8. It is evaluated with M_i=1000 and K_i=1 for all servers for the proposed and conventional schemes. As shown in Fig. 9, the proposed scheme is effective to reduce T_{delay} under the condition that T_{delay} is determined by $T_{\text{db}}^{\max}$; $T_{\text{db}}^{\max} \geq T_{\text{te}}^{\max}$. The proposed scheme is effective to reduce T_{delay} under the condition of $T_{\text{db}}^{\max} \geq T_{\text{te}}^{\max}$. These results indicate that the proposed scheme is effective for a wide-area network with multiple distributed clouds.

We discuss the computation time of the proposed scheme. We investigate the dependency of computation time on the number of TEs and servers. Figures 10(a) and (b) show the location of 10 TEs and 100 TEs. Figures 10(c) and (d) show the location of the 3 servers and 40 servers. In Fig. 10(d), four cities with larger populations in each prefecture are added as new server locations. The location of new servers is based on the latitude and longitude information of the city. We assume that the new servers connect to the representative node in each prefecture shown in Fig. 5(b). Table 4 shows the computation time for the



(a) IoT devices located near to servers (Type-A)



(b) IoT devices located far from servers (Type-B)

Fig. 8 Different patterns of TEs' location.

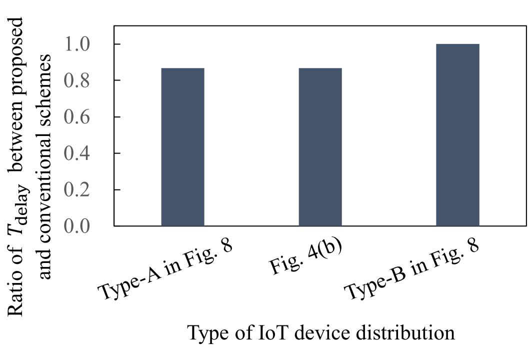
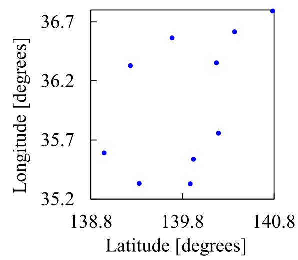
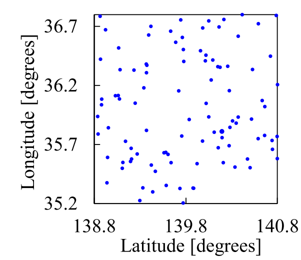


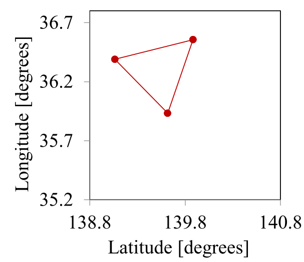
Fig. 9 Ratio of T_{delay} obtained by proposed scheme to that of conventional scheme



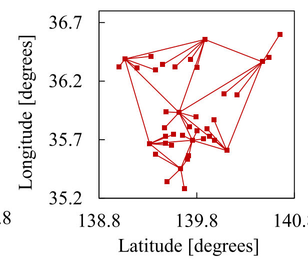
(a) 10 TEs



(b) 100 TEs



(c) 3 servers



(d) 40 servers

Fig. 10 Locations of TEs and servers.

Table 4 Average computation time [sec] and variance [sec^2] for combined patterns of each number of TEs and servers.

No. of servers	No. of TEs: 10		100		1000	
	Time	Var.	Time	Var.	Time	Var.
3	0.01	0.000	0.03	0.000	0.25	0.000
8	0.09	0.000	0.88	0.000	29	0.034
40	3.7	0.001	50.5	0.162	9709	2269.590

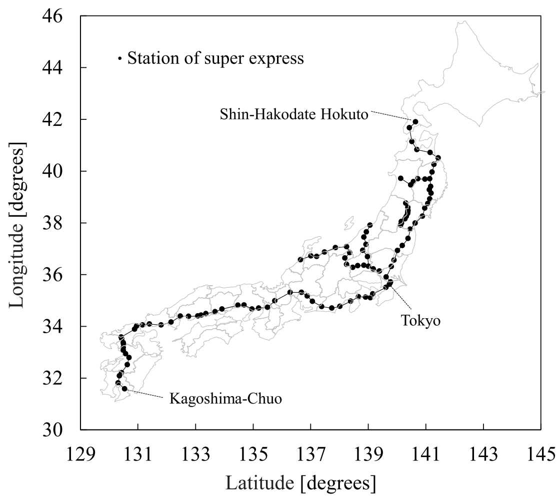
Table 5 Average computation time [sec] and variance [sec^2] for proposed scheme.

COST239			JPN		
M_i	Time	Var.	M_i	Time	Var.
1000	76.1	0.2	1000	43.5	0.4
120	1429.3	46.8	200	70.2	0.7
91	401.8	17.2	125	65.6	0.4

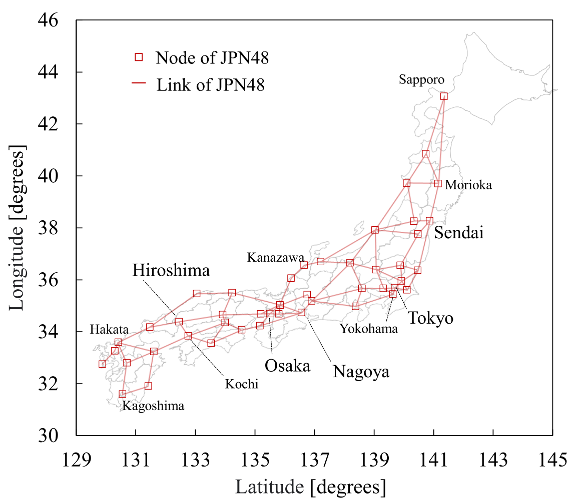
combined patterns of each number of TEs and servers. The maximum computation time with 1000 TEs and 40 servers is 9709 [sec] (2.7 hours) with the variance of 2270 [sec^2]. This is usually an acceptable network design time before the service launch. However, for applications with thousands of TEs where network design must be determined in real-time, it is an issue to reduce the computation time. Table 5 shows the computation time and variance at evaluation of M_i limitation of the proposed scheme for COST239 and JPN. For both COST239 and JPN with M_i limitation, the computation times are less than 25 [min] (1500 [sec]) with the variance of less than 50 [sec^2].

5.2 Network of Assumed Monitoring Camera System

In this section, we compare the proposed and conventional schemes under conditions more similar to an actual service network. We consider a use-case where surveillance cameras are located at railway stations. DB and APL can be allocated on servers on the cloud located throughout Japan. The location of TEs has assumed the station of the super express, as shown in Fig. 11(a). The number of stations is 108. The location of servers and the links between servers are assumed the node and link of JPN48 (remove Okinawa node) [10], respectively, as shown in Fig. 11(b). The number of serves is 47. We assume that each TE can be connected to all servers. For example, the camera at the Sin-Hakodate



(a) Locations of super express stations



(b) Network topology of JPN48

Fig. 11 Locations of TEs and servers for assumed monitoring camera system network.

Hokuto Station can be connected to all servers. We assume that all servers are logically connected via JPN48 links. For example, the Sapporo node is connected to the Sendai node via the Morioka node. In this network, the number of candidate links between TEs and servers is 5076, and the number of candidate links between servers is 1081. The distance between TEs and JPN48 nodes is assumed to be three times the straight line distance on the map based on the latitude and longitude. This assumption is based on a typical local network with a ring-topology. The distance between servers is assumed the link distance of JPN48.

We discuss the network topology of the monitoring camera system network. Figures 12(a) and (b) show the network topologies for the conventional and proposed schemes, respectively. In both schemes, DB is allocated to the Tokyo server. In the conventional scheme, as shown in Fig. 12(a), $T_{\rm te}^{\rm max}$ is 1.8 ms between TE of the Shin-Hakodate Hokuto station and the Sapporo server. $T_{\rm db}^{\rm max}$ is 7.2 ms between the Kagoshima server (APL) and the Tokyo server (DB). $T_{\rm delay}$ is 7.2 ms. In the proposed scheme, as shown in Fig. 12(b), $T_{\rm te}^{\rm max}$ is 4.9 ms between TE of the Kagoshima-Chuo station and the Kochi server. $T_{\rm db}^{\rm max}$ is 4.8 ms between the Kochi server (APL) and the Tokyo server (DB). $T_{\rm delay}$ is 4.9 ms.

We discuss the results of the delay performance under the condition that DB location is fixed. Figure 13 shows $T_{\rm db}^{\rm max}$ and $T_{\rm te}^{\rm max}$ in the conventional and proposed schemes. In Fig. 13, DBs are allocated on Sendai, Tokyo, Nagoya, Osaka, and Hiroshima servers. Using the proposed scheme, the delay of the proposed scheme, where DBs are allocated at Sendai, Tokyo, Nagoya, Osaka, and Hiroshima servers, are 65.4, 67.6, 70.5, 82.4, and 86.8 percent, respectively, compared to that of the conventional scheme. In the conventional scheme, as shown in Fig. 12(a), TEs select a nearest server; $T_{\rm te}^{\rm max}$ and $T_{\rm db}^{\rm max}$ are 1.8 ms and 7.2 ms, respectively. In the proposed scheme, as shown in Fig. 12(b), TEs select the server to minimize $T_{\rm delay}$; $T_{\rm te}^{\rm max}$ and $T_{\rm db}^{\rm max}$ are 4.8 ms and 4.9 ms, respectively. From these results, $T_{\rm te}^{\rm max}$ of the proposed scheme is increased from 1.8 ms to 4.9 ms but $T_{\rm db}^{\rm max}$ of the proposed scheme is reduced from 7.2 ms to 4.8 ms. Therefore, $T_{\rm delay}$ of the proposed scheme is reduced from 4.9 ms to 4.8 ms. TEs, located near DB that is not involved in determining $T_{\rm delay}$, select the nearest server in the same way as the conventional scheme. For example, TEs in the vicinity of the Tokyo server select the nearest Tokyo server for the conventional and proposed schemes since $T_{\rm delay}$ is the same whether they select either the Tokyo or the Yokohama server. Regarding the location of DB, the delay reduction of the proposed scheme is less effective if the location of DB is near the center of the inter-server network. If DB is allocated far from the center of the inter-server network, $T_{\rm delay}$ could be reduced by optimizing the location of APLs.

We then discuss the results of the delay performance under the condition that DB location is not fixed. In Fig. 13, when the location of DB is not fixed, it is observed that $T_{\rm delay}$ is minimized compared to the results where the location of DB is fixed. The delay of the proposed scheme when the location of DB is not fixed is 53 percent compared to that of the conventional scheme in which DB is allocated at the Tokyo server. In the proposed scheme, when the location of DB is not fixed, DB is allocated at the Nagoya server. It is an optimal DB location in which $T_{\rm delay}$ is minimized at the condition that the location of DB is not fixed. If there is no constraint on the location of DB, the network topology including the location of DB can be determined to minimize $T_{\rm delay}$. When the location of DB is not fixed, it can be observed that $T_{\rm delay}$ is reduced compared to the case where DB location is not fixed. It can be reduced $T_{\rm delay}$ by locating DB in an area near the center of the inter-server network so that the delay between DB and APL is reduced.

We discuss the computation time of the proposed scheme. Table 6 shows the computation time and variance of the proposed scheme. The computation times are less than 25 [sec] when the position of DB is fixed. When the location of DB is not fixed, the computation time is 96 [sec] since the location of DB needs to be optimized. In case the

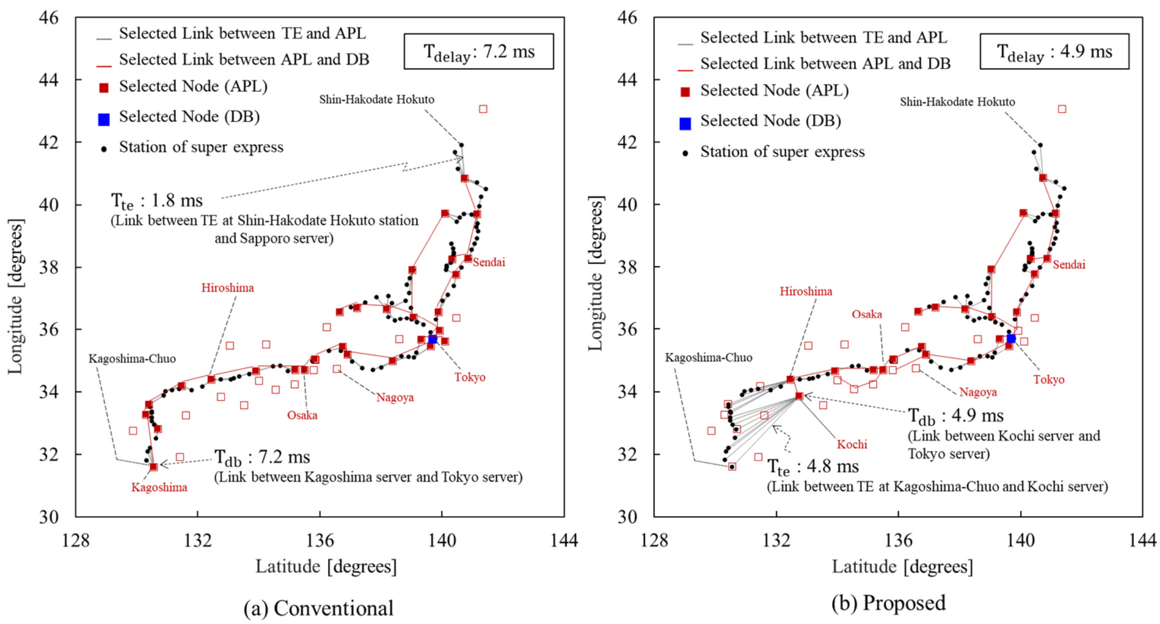


Fig. 12 Determined network topologies using conventional and proposed schemes when DB is allocated at Tokyo server.

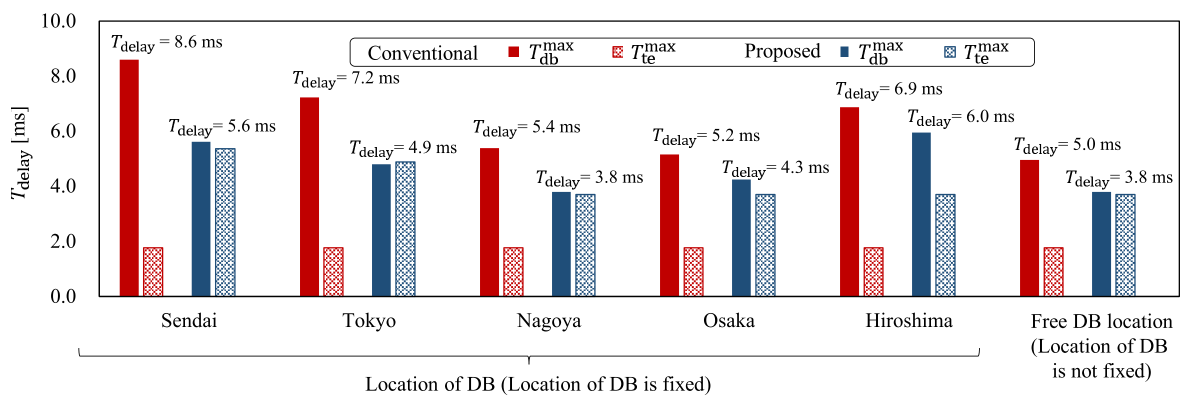


Fig. 13 T_{delay}, $T_{\text{db}}^{\max}$ and $T_{\text{te}}^{\max}$, of conventional and proposed schemes.

Table 6 Average computation time [sec] and variance [sec^2] for proposed scheme.

Location of DB	Time	Var.
Sendai	22.9	0.103
Tokyo	15.4	0.058
Nagoya	5.3	0.006
Osaka	2.1	0.002
Hiroshima	2.2	0.001
Free DB	96.0	0.070

location of DB is fixed, the number of candidate servers for DB is one, and the number of candidate servers for APL is 47. In the non-fixed DB location case, the number of candidate servers for DB and APL is 47. The above results show that the computation time is acceptable for a network design process before the service is launched.

6. Conclusion and Future Direction

This paper proposed a network design scheme for detecting quickly in delay-sensitive monitoring services. In this paper, we consider reducing the delay between the latest update of reference data and the start of processing the updated data. The proposed scheme achieves minimal transmission delay by providing a network design scheme for databases, applications, and IoT devices. We evaluated the delay reduction and computing time of the proposed scheme. In JPN and COST239, the delays of the proposed scheme are 78 and 80 percent, respectively, compared to the conventional scheme. In the assumed monitoring camera system network, the delays of the proposed scheme where DB is allocated in the Tokyo server are less than 70 percent compared to that of the conventional scheme when DB is allocated in the Tokyo server. The delays of the proposed scheme where DB is not in the fixed location are less than 55 percent compared to that of the conventional scheme when DB is allocated in the Tokyo server. From these results, the proposed scheme contributes to designing an IoT network to detect the newly added reference data more quickly. When the proposed scheme is formulated as an ILP problem, the computation time is acceptable for a network design process before launching the service.

It is necessary to consider additional conditions for actual networks, such as redundancy, the bandwidth of links, the maximum resource of each server, and the terminal participation scenario. These additional conditions must be considered as the constraints for network design. Furthermore, to provide multiple different applications using the same network and the same server, the delay and the connection between TE-server and server-server should be determined on an application-by-application basis, and network resources should be treated in common resources. Developing the network design scheme that considers these additional conditions is left as part of future works.

Acknowledgments

This work was supported by ROIS NII Open Collaborative Research 2023-23S0201. The authors would like to thank Information Media Center (IMC) at Toyohashi University of Technology for providing computer resources and the kind support of its staff.

References

[1] "AWS for the Edge," https://aws.amazon.com/edge/services/?nc1=h_ls (in Japanese), accessed Jan. 17. 2023.
[2] "Azure IoT Edge," https://azure.microsoft.com/en-us/services/iot-edge/, accessed Jan. 17. 2023.
[3] Y.C. Hu, M. Patel, D. Sabella, N. Sprecher, and V. Young, "Mobile edge computing — A key technology towards 5G," ETSI white paper, vol.11, no.11, pp.1–16, 2015.
[4] "NTT DOCOMO to Launch 5G Service in Japan on March 25," https://www.docomo.ne.jp/english/info/media_center/pr/2020/0318_00.html, accessed Jan. 17. 2023.
[5] N. Chen, Y. Chen, S. Song, C.T. Huang, and X. Ye, "Smart urban surveillance using fog computing," ACM/IEEE Symposium on Edge Computing (SEC), Washington, DC, USA, pp.95–96, 2016. DOI: 10.1109/SEC.2016.25
[6] Y. Yamato, Y. Fukumoto, and H. Kumazaki, "Analyzing machine noise for real time maintenance," The 2016 8th International Conference on Graphic and Image Processing (ICGIP 2016), Tokyo, Japan, vol.10225, pp.513–519, 2016. DOI: 10.1117/12.2266108
[7] S. Singh, "Optimize cloud computations using edge computing," International Conference on Big Data, IoT and Data Science (BID), Pune, India, pp.49–53, 2017. DOI: 10.1109/BID.2017.8336572
[8] W. Yu, F. Liang, X. He, W.G. Hatcher, C. Lu, J. Lin, and X. Yang, "A survey on the edge computing for the Internet of Things," IEEE Access, vol.6, pp.6900–6919, 2017. DOI: 10.1109/ACCESS.2017.2778504
[9] M. O'mahony, "Ultrahigh capacity optical transmission network: European research project cost 239," Information, Telecommunications, Automata Journal, vol.12, pp.33–45, 1993.
[10] "Japan Photonic Network Model," https://www.ieice.org/cs/pn/jpn/jpnm.html (in Japanese), accessed Jan. 17. 2023.
[11] A. Kawabata, T. Tojo, B.C. Chatterjee, and E. Oki, "An optimal allocation scheme of database and applications for delay sensitive IoT services," IEEE Global Communications Conference (GLOBECOM), Madrid, Spain, pp.1–6, 2021. DOI: 10.1109/GLOBECOM46510.2021.9685736
[12] A. Yahyaoui, T. Abdellatif, S. Yangui, and R. Attiai, "EAD-IoT: Reliable event and anomaly detection framework for the Internet of Things," IEEE Access, vol.9, pp.24168–24186, 2021. DOI: 10.1109/ACCESS.2021.3056149
[13] R. Mayer, B. Koldehofe, and K. Rothermel, "Predictable low-latency event detection with parallel complex event processing," IEEE Internet Things J., vol.2, no.4, pp.274–286, 2015. DOI: 10.1109/JIOT.2015.2397316
[14] A.I. Maarala, M. Rautiainen, M. Salmi, S. Pirttikangas, and J. Riekki, "Low latency analytics for streaming traffic data with Apache Spark," IEEE International Conference on Big Data (Big Data), Santa Clara, CA, USA, pp.2855–2858, 2015. DOI: 10.1109/BigData.2015.7364101
[15] H. Khazaei, R. Veleda, M. Litoiu, and A. Tizghadam, "Realtime big data analytics for event detection in highways," IEEE 3rd World Forum on Internet of Things (WF-IoT), Reston, VA, USA, pp.472–477, 2016. DOI: 10.1109/WF-IoT.2016.7845461
[16] P. Ntumba, N. Georgantas, and V. Christophides, "Scheduling continuous operators for IoT edge analytics," IEEE International Conference on Smart Computing (SMARTCOMP), Helsinki, Finland, pp.55–60, 2021. DOI: 10.1109/SMARTCOMP55677.2022.00026
[17] H. Tran-Dang and D.S. Kim, "FRATO: Fog resource based adaptive task offloading for delay-minimizing IoT service provisioning," IEEE Trans. Parallel Distrib. Syst., vol.32, no.10, pp.2491–2508, 2021. DOI: 10.1109/TPDS.2021.3067654
[18] A. Althoubi, R. Alshahrani, and H. Peyravi, "Delay analysis in IoT sensor networks," Sensors, vol.21, no.11, p.3876, 2021. DOI: 10.3390/s21113876
[19] Z. Fei, Y. Wang, R. Sun, and Y. Yuanfei, "Delay-oriented task scheduling and bandwidth allocation in fog computing networks," IEEE Global Communications Conference (GLOBECOM), Waikoloa, HI, USA, pp.1–6, 2019. DOI: 10.1109/GLOBECOM38437.2019.9013494
[20] A. Yousefpour, G. Ishigaki, and J.P. Jue, "Fog computing: Towards minimizing delay in the Internet of Things," IEEE International Conference on Edge Computing (EDGE), Honolulu, HI, USA, pp.17–24, 2017. DOI: 10.1109/IEEE.EDGE.2017.12
[21] W. Masri, I. Al Ridhawi, N. Mostafa, and P. Pourghomi, "Minimizing delay in IoT systems through collaborative fog-to-fog (F2F) communication," International Conference on Ubiquitous and Future Networks (ICUFN), Milan, Italy, pp.1005–1010, 2017. DOI: 10.1109/ICUFN.2017.7993950
[22] Q. Fan and N. Ansari, "Application aware workload allocation for edge computing-based IoT," EEE Internet Things J., vol.5, no.3, pp.2146–2153, 2018. DOI: 10.1109/JIOT.2018.2826006
[23] Y. Nam, S. Song, and J.M. Chung, "Clustered NFV service chaining optimization in mobile edge clouds," IEEE Commun. Lett., vol.21, no.2, pp.350–353, 2017. DOI: 10.1109/LCOMM.2016.2618788
[24] M. Guo, L. Li, and Q. Guan, "Energy-efficient and delay-guaranteed workload allocation in IoT-edge-cloud computing systems," IEEE Access, vol.7, pp.78685–78697, 2019. DOI: 10.1109/ACCESS.2019.2922992
[25] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, and W. Zhao, "A survey on Internet of Things: Architecture, enabling technologies, security and privacy, and applications," IEEE Internet Things J., vol.4, no.5, pp.1125–1142, 2017. DOI: 10.1109/JIOT.2017.2683200
[26] M. Wu, T.J. Lu, F.Y. Ling, J. Sun, and H.Y. Du, "Research on the architecture of Internet of Things," International Conference on Advanced Computer Theory and Engineering (ICACTE), Chengdu, vol.5, pp.V5-484–487, 2010. DOI: 10.1109/ICACTE.2010.5579493
[27] A. Rehman, A. Paul, M.A. Yaqub, and M.M.U. Rathore, "Trustworthy intelligent industrial monitoring architecture for early event detec-

tion by exploiting social IoT," Annual ACM Symposium on Applied Computing, pp.2163–2169, 2020. DOI: 10.1145/3341105.3373996

[28] "MQTT: The Standard for IoT Messaging," https://mqtt.org/, accessed Jan. 17. 2023.

[29] "Global IP Network," https://www.ntt.com/en/services/network/gin/sla.html, accessed Jan. 17. 2023.

[30] "Service Level Agreements of NTT WEST," https://www.ntt-west.co.jp/business/solution/sp/04 (in Japanese), accessed Jan. 17. 2023.

[31] J. Postel, "RFC 792: Internet control message protocol," 1981.

[32] D. Mills, J. Burbank, and W. Kasch, "RFC 5905: Network time protocol version 4: Protocol and algorithms specification," 2010.

[33] "IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems," IEEE Std 1588-2008, 2008.

[34] "Solving Constraint Integer Programs (SCIP)," https://www.scipopt.org/, accessed Jan. 17. 2023.

Appendix: Formulation of Conventional Scheme with M_i Limitation

In the conventional scheme, DB is allocated to the server which minimizes the maximum delay of a link between APL and DB. In terms of the decision of $T_{\mathrm{te}}^{\max}$, since it may not be possible to select the nearest server under the M_i limitation, we use the server selection method that minimizes the maximum delay of a link between TE and server.

The server selection problem in the conventional scheme with M_i limitation is formulated as an ILP problem by:

$$\text{Objective min} \quad T_{\mathrm{te}}^{\max} + \alpha T_{\mathrm{db}}^{\max} \qquad (\mathrm{A}\cdot 1\mathrm{a})$$

$$\text{s.t.} \quad (1\mathrm{b})\text{–}(1\mathrm{h}),\ (1\mathrm{k})\text{–}(1\mathrm{p}). \qquad (\mathrm{A}\cdot 1\mathrm{b})$$

The first term of the objective function in (A· 1a) is the maximum delay of the link between TEs and servers with APL. The second term in (A· 1a) is the maximum delay of the link between servers with APL and servers with DB. The parameter of α is set in such a way that the second term is a sufficiently small value and does not affect the first term. Table A· 1 shows the given parameters and decision variables.

Table A· 1 Given parameters and decision variables.

Given parameters	d_{ti}	Delay between TE t and server i
	d_{ij}	Delay between server i and server j
	M_i	Maximum number of TEs who belong to server i
	α	Sufficiently small constant
Decision variables	T_{delay}	Delay between latest update of reference data and the start of processing updated data
	$T_{\mathrm{te}}^{\max}$	Maximum delay between TEs and APLs
	$T_{\mathrm{db}}^{\max}$	Maximum delay between DB and APLs
	x_{kl}	Link between node k and l is selected or otherwise
	y_i	Server i is selected or otherwise
	z_i	Server i is allocated DB or otherwise

Akio Kawabata is a Professor at Toyohashi University of Technology, Aichi, Japan. He received the B.E., M.E., and Ph.D. degrees from the University of Electro-Communications, Tokyo, Japan, in 1991, 1993, and 2016, respectively. He was with Nippon Telegraph and Telephone Corporation (NTT) Laboratories, Tokyo, from 1993 to 2022. He is a Professor at Toyohashi University of Technology since 2022. He served as a senior manager of R&D Department at NTT East from 2011 until 2014. His research interests include distributed systems, network architecture, and switching systems.

Takuya Tojo is a senior research engineer at Nippon Telegraph and Telephone Corporation (NTT), Tokyo, Japan. He received the B.E., M.E., and Ph.D. degrees from Tokyo Denki University, Tokyo, Japan, in 2002, 2004, and 2007, respectively. In 2004, he joined Nippon Telegraph and Telephone Corporation (NTT) Service Integration Laboratories. He has been engaging in research on QoS and IP/optical network architecture.

Bijoy Chand Chatterjee is an Assistant Professor and DST Inspire Faculty at South Asian University (SAU), New Delhi, and an Adjunct Professor at the Indraprastha Institute of Information Technology Delhi (IIITD), New Delhi, India. Before joining SAU, he was with IIITD, Norwegian University of Science and Technology, Trondheim, Norway, and The University of Electro-Communications, Tokyo, Japan. His research interests include optical networks, QoS-aware protocols, optimization, and routing. He is a senior member of IEEE and a Fellow of IETE. More information can be found at https://www.bijoycc.site/.

Eiji Oki is a Professor at Kyoto University, Kyoto, Japan. He received the B.E. and M.E. degrees in instrumentation engineering and a Ph.D. degree in electrical engineering from Keio University, Yokohama, Japan, in 1991, 1993, and 1999, respectively. He was with Nippon Telegraph and Telephone Corporation (NTT) Laboratories, Tokyo, from 1993 to 2008, and The University of Electro-Communications, Tokyo, from 2008 to 2017. From 2000 to 2001, he was a Visiting Scholar at Polytechnic University, Brooklyn. He is a Professor at Kyoto University, Kyoto, Japan, since 2017. His research interests include routing, switching, protocols, optimization, and traffic engineering in communication and information networks. He is an IEEE Fellow.