

PAPER

MHND: Multi-Homing Network Design Model for Delay Sensitive Applications

Akio KAWABATA^{†a)}, Senior Member, Bijoy CHAND CHATTERJEE^{††}, Nonmember, and Eiji OKI^{†††}, Fellow

SUMMARY When mission-critical applications are provided over a network, high availability is required in addition to a low delay. This paper proposes a multi-homing network design model, named MHND, that achieves low delay, high availability, and the order guarantee of events. MHND maintains the event occurrence order with a multi-homing configuration using conservative synchronization. We formulate MHND as an integer linear programming problem to minimize the delay. We prove that the distributed server allocation problem with MHND is NP-complete. Numerical results indicate that, as a multi-homing number, which is the number of servers to which each user belongs, increases, the availability increases while increasing the delay. Noteworthy, two or more multi-homing can achieve approximately an order of magnitude higher availability compared to that of conventional single-homing at the expense of a delay increase up to two times. By using MHND, flexible network design is achieved based on the acceptable delay in service and the required availability.

key words: delay sensitive service, network design, availability, distributed computing, conservative synchronization

1. Introduction

According to recent trends in networking, launching the fifth-generation (5G) service facilitates communications with low delay and high bandwidth [1]. In particular, telecommunications carriers are actively developing technologies for low delay communication, such as all photonics networks [2]. In addition, various Internet of Things (IoT) services are being provided via networks. Recently, data centers that used only a few locations are now being deployed as many widely distributed edge data centers across the country. These environmental changes accelerate providing IoT applications with low delay and high bandwidth. These conditions are expected to enable mission-critical applications, such as telesurgery, multiple drone control, and autonomous vehicles, which were previously difficult to achieve, to be provided via networks.

When providing applications with low delay and high availability using multiple servers distributed over a wide area, a design model for a network consisting of multiple servers with low delay and high availability is required. At

network design for high availability, redundancy must be ensured so that a single failure does not render an application unavailable. A question grabs our attention: how can we provide higher availability of network service with low delay even in a network failure?

To address the above question, this paper proposes a multi-homing network design model (MHND). In MHND, each user belongs to multiple servers so that applications can continue to be used even in the event of user-server link failures or server failures; the order of event occurrence is guaranteed by using conservative synchronization [3]. It is ensured that the delay is not affected in the event of link failure or server failure, which is determined by the delay of the link with the largest delay among the multiple user-server links. We formulate MHND as an integer linear programming (ILP) problem to minimize the delay, with the number of belonging servers at multi-homing as a given parameter. We prove that the distributed server allocation problem with MHND is NP-complete. We evaluate MHND in terms of delay and service availability and compare it to conventional single-homing. Numerical results indicate that MHND is effective to design networks for balancing low delay and high availability. In addition, the measured computation time of MHND by solving the ILP problem indicates that MHND can be used in practical scenarios. The main contributions of MHND are summarized as follows:

- Regardless of the users' location, application processing proceeds to keep the order in which events occur for all users. This means that all users using the application share the same time space.
- High redundancy can be achieved by specifying the number of servers to which users belong. For example, if each user belongs to three servers, the service can continue in the case of triple failures.
- Even if the servers are switched over due to failure, the maximum delay is not changed; the delay is still acceptable for the service quality, even if multiple failures occur.

The proposed scheme is intended to be implemented as a lower layer of applications such as operating systems (OS) or middleware. Since applications receive events from the middleware implemented with the proposed scheme in occurrence order, the processing proceeds without the rollback process to guarantee the order of events. This is not burdensome to the application and has a wide range of applications. In addition, we evaluate that the proposed scheme

Manuscript received March 22, 2023.

Manuscript revised May 27, 2023.

Manuscript publicized July 24, 2023.

[†]The author is with the Toyohashi University of Technology, Toyohashi-shi, 441-8580 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.2023EBP3046

is expected to be more versatile by verifying it with two different network models. Therefore, in terms of the burden on applications and the versatility of the network topology, the proposed scheme is a versatile scheme that can be realized based on a general infrastructure consisting of a wide-area network and multiple edge clouds.

This paper is an extended version of the work [4]. The main additions are as follows. We present related works and the originality of our work. We additionally investigate evaluations of delay and availability in a wide area network throughout Japan. We also evaluate the variances of the computation time for all evaluations.

The rest of the paper is organized as follows. Section 2 presents related works and the originality of our work. Section 3 presents the prerequisite of the proposed model, its formulation as an optimization problem, and its computational time complexity. In Sect. 4, we evaluate the proposed model in terms of delay, availability, and computation time. Finally, Sect. 5 concludes this paper.

2. Related Work

We discuss related works in terms of delay and availability of distributed processing with processing events on occurrence order. Next, we discuss the position and originality of our work.

Research works that guarantee the order of events have been studied in parallel and distributed processing. These works are mainly classified into two categories: conservative synchronization and optimistic synchronization [3]. In conservative synchronization, time information is given to events, and the events are rearranged in the order of occurrence before processing the application, thereby guaranteeing the order of the events. In optimistic synchronization, events are processed in the order of arrival, and if past events are received, the status is rolled back, and the processing result is corrected. Time Warp is known for implementing a rollback process [5]. As for research on distributed processing that guarantees the order of events focusing on delay, a server selection model that minimizes the delay of distributed processing systems using conservative synchronization has been studied without considering any failures [6]. The work in [7] introduced a server selection model with preventive start-time optimization by sharing backup server resources to minimize the delay in switching the belonging server after a single server failure, which can cause service interruptions in the server switching operation due to the backup sharing nature.

In terms of availability, the works mentioned above [3], [5], [6] are based on the single-homing, which provides less availability of services and may not provide service continuity in user-server link and server failures. The work in [8] introduced trailing state synchronization (TSS) for multi-player games with low latency but strong consistency requirements. It is based on an optimistic synchronization mechanism and provides low-latency, consistent game-play through the use of multiple copies of the game state and

rollbacks. The work has multiple mirrors of the application state but does not consider availability in case of network failure. A new transport protocol, named latency-controlled end-to-end aggregation protocol (LEAP) [9], achieves multiple paths for the transmit data and satisfies delay constraints using Forward Error Correction (FEC). The work [10] introduced a 6G-based edge intelligence solution for ultra-reliable low-delay applications such as multiple drone control, border surveillance, and telesurgery, where the delay of milliseconds is not tolerable. The work mentioned that merging 6G with edge intelligence overcomes the issues of delay, security, and reliability. The work in [11] introduced how the network resources can be optimized and key enablers to achieve low delay and high availability reliability for use cases characterized by edge networking. The availability of service is essential and typically can be improved with the increase in the multi-homing. The work in [12] introduced a service function chaining for virtualized network functions considering delay and availability. These efforts [8]–[12] various studies using edge computing to achieve low delay and high availability are a recent trend in providing mission-critical applications. However, there is no study on how to design networks that contribute to low delay and high availability. Our work is based on a network design model to achieve these requirements and to balance delay and availability according to service requirements.

In terms of network dynamic design utilizing Network Functions Virtualization (NFV) and Mobile Edge Computing (MEC) technologies is attracting attention. The work in [12] introduced a scheduling method to achieve low delay and high availability in a VNF environment considering availability and acceptable end-to-end delay. The integer non-linear programming (INLP) problem approach to finding optimal solutions and the heuristic approach to solving the problems efficiently are presented. The two presented approaches were compared in terms of the acceptance rate of processing requests, the number of selected nodes, and the processing time. It was shown that the presented heuristic approach achieves less computation time and achieves the performance close to the optimal solution obtained by INLP. When using MEC to provide low delay services, the issue is which edge server is selected by the user.

The work in [13] introduced a method to select edge servers and control the flow by considering the processing delay and the communication delay when a service request is processed. The acceptance rate of processing requests was compared for three methods of edge server selection: random selection, selection of the edge server with the lowest delay, and selection of the edge server with the lowest processing load. The method of selecting the edge server with the lowest delay resulted in a higher acceptance rate but also resulted in a higher rate of decision errors.

The work in [14] introduced a concept for Medical Cyber-Physical Systems (MCPS) using Fog computing. The work aims to manage network resources while maintaining quality cost-effectively. It minimizes costs by considering base station selection, subcarrier assignment, VM deploy-

ment, and task distribution. The work is formulated as a mixed integer nonlinear problem and extended to mixed integer linear programming and two-phase heuristic methods as a response to computational complexity. The presented two-phase heuristic achieves a cost reduction ratio close to the optimal solution compared to the greedy method.

The above-mentioned works [12]–[14] contribute to network design that considers delay, availability, bandwidth, etc. when multiple candidate servers are available to users, such as MEC. Particularly in virtualized networks, determining VNF placement and user attribution prior to service launch is an important topic. There are two critical differences between these efforts and our works. One is that all events are sorted in occurrence order before application processing to eliminate unfairness due to the delay difference (fairness of events). The second is that the maximum delay does not change as servers are switching over.

In terms of Content Delivery Network (CDN) network design, various efforts are being made to achieve both quality and availability. Content multi-homing refers to using multiple content delivery networks on a network for delivery. The work in [15] introduced a selection method of optimal CDN from among multiple candidates in terms of quality and cost. The combination of two functions has reduced costs and quality degradation. One is computed at the distributor, which calculates the optimal CDN assignment considering cost, availability, and performance. The other is adaptation algorithms providing the ability for viewers to utilize multiple CDN servers efficiently. Multi-CDN federations, where standalone CDNs are interconnected, allow dynamically selecting a higher-performing CDN while maintaining redundancy. On the other hand, there are challenges to the real-time interconnection of multiple CDNs. To address these challenges, CDN semi-federation [16] was introduced as a method that can be implemented on existing CDNs. The method identified the impact on CDN performance from traffic patterns and reduced the cumulative delay of the CDN with optimal traffic dispatch, resulting in a 20% reduction in delivery delay.

These efforts [15], [16] contribute to configuring the network with high availability and delay quality in mind. In a CDN network, content is delivered in the direction of the user, and thus latency differences between users are acceptable. Our approach, on the other hand, aims to reduce delay when an application is used by multiple users keeping the order in which events occur regardless of the user’s location.

3. Proposed Model

In this section, first, we discuss the prerequisite of the communication and processing process to guarantee processing events in occurrence order. Secondly, we discuss the prerequisite of multi-homing and the communication and processing of multi-homing. Thirdly, we discuss the formulation of MHND. Finally, we discuss the computational time complexity of MHND by proving that the decision version of MHND is NP-complete.

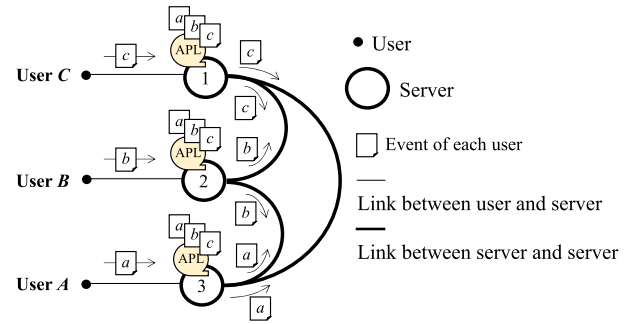


Fig. 1 Communication and processing model between servers.

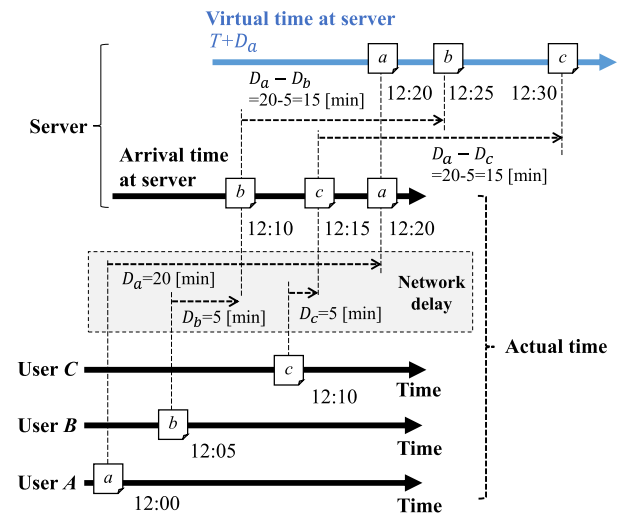


Fig. 2 Example of correcting order of events between user and server.

3.1 Prerequisite of Communication and Processing Process between Servers

MHND uses conservative synchronization, which rearranges the events of all users before processing the application. As shown in Fig. 1, user events are assumed to be multicasted between servers for distributed processing. Each server processes the events of all users. In other words, all users’ events are processed in parallel on all servers.

3.2 Prerequisite of Guaranteeing Order of Events

The concept of virtual time is introduced and events of all users are rearranged by virtual time in the order in which they occur [6]. Figure 2 shows the order guarantee of events using virtual time. Events *a*, *b*, and *c* occur at 12:00, 12:05, and 12:10, respectively. The network delays between user *A*-server, user *B*-server, and user *C*-server are $D_a=20$ [min], $D_b=5$ [min], and $D_c=5$ [min], respectively. Therefore, the time when events *a*, *b*, and *c* are received at the server is 12:20, 12:10, and 12:15, respectively, and an order reversal occurs. As shown in Fig. 2, all events are rearranged with $T+20$ [min] at the virtual time by adding $D_a=20$ [min], the maximum value of user-server delay, to the current time *T*.

If the maximum user-server delay denotes D_U^{\max} , the user-server event correction is performed at $T + D_U^{\max}$.

Similarly to the user-server event correction, order correction is performed for server-server multicast communication. If the maximum server-server delay is D_S^{\max} , the server-server event correction is performed at $T + D_S^{\max}$. Thus, as in the example in Fig. 1, each user's event is multicasted to all servers via the belonging server, so that each server processes all user events in parallel at $T + D_U^{\max} + D_S^{\max}$. When sending the processing result of each server to the user is sent, the maximum delay between the user and server, D_U^{\max} , is the queuing process for the network delay. The delay in arrival time for the processing results at each user is treated to there is no difference in delay regardless of the distance of each user. Fair application processing is achieved at the time of $T + 2D_U^{\max} + D_S^{\max}$ for all users, and the delay, T_{delay} , is expressed as follows:

$$T_{\text{delay}} = 2D_U^{\max} + D_S^{\max}. \quad (1a)$$

3.3 Prerequisite of Multi-Homing

In the communication and processing of Sect. 3.1, redundancy is ensured between servers, as all servers process all users' events in parallel. That is, even if one or more selected servers fail, parallel processing can be performed by the remaining selected servers. In MHND, redundancy is achieved by multi-homing, where each user belongs to multiple servers. Figure 3 shows examples of dual-homing and triple-homing. Thanks to multi-homing, the application can continue to be available even in the event of user-server link failures and server failures.

3.4 Communication and Processing of Multi-Homing

Each user belongs to multiple servers, as described in Sect. 3.3. All events are multicasted between servers and each server processes the events of all users, as described in Sect. 3.1. The event order is guaranteed by using conservative synchronization, as described in Sect. 3.2. When a user with single-homing, the delay of the user-server link is uniquely determined. On the other hand, in the case of multi-homing, the delay is determined using the selected

link with the largest delay. This is because the delay, which is calculated by the user-server link with the smallest delay, must be recalculated due to a user-server link failure or a server failure.

If a user belongs to multiple servers, the user multicasts each event to all belonging servers. In the examples of Figs. 3(a) and (b), user A sends the same event to servers 1 and 2, and user A sends the same event to servers 1, 2, and 3, respectively. The duplicate events are discarded since server 1 receives the event from user A in the link directly connected to user A and receives the event via other servers in duplicate. All events are rearranged by the time of occurrence at each server by adding the time information to the events at each user using highly accurate time information such as the precision time protocol (PTP) [17]. This time information and user information are used to discard duplicate events.

3.5 Formulation

MHND is formulated as an ILP problem. We consider a network described as an undirected graph $G(V, E)$. Let V and E denote a set of edges and a set of nodes, respectively. $V_U \subseteq V$ denotes the set of users, and $V_S \subseteq V$ denotes the set of servers. $V_U \cap V_S = \emptyset$ and $V_U \cup V_S = V$. $E_U \subseteq E$ denotes the set of user-server links, and $E_S \subseteq E$ denotes the set of server-server links. $E_U \cap E_S = \emptyset$ and $E_U \cup E_S = E$. A link between user $p \in V_U$ and server $i \in V_S$ is expressed as $(p, i) \in E_U$, and a link between server $i \in V_S$ and server $j \in V_S \setminus \{i\}$ is expressed as $(i, j) \in E_S$.

In the formulation, it is assumed that the user has logical connectivity to all servers and can select an optimal server. It is also assumed that there is a full mesh of logical connectivity between servers and can select optimal links. MHND determines an optimal network topology that satisfies the constraints, including multi-homing, and minimizes T_{delay} .

Table 1 shows the given parameters and decision variables. The given parameters in the ILP problem are defined as follows. d_{pi} denotes the delay of link $(p, i) \in E_U$ and is given for all candidate links between the user and the server. d_{ij} denotes the delay of link $(i, j) \in E_S$ and is given for all candidate links between two servers. m denotes the multi-homing number. Each user belongs on m servers in a multi-homing topology. M_i denotes the maximum number of users that server $i \in V_S$ can accommodate and is given for all candidate servers. Y_{\max} denotes the maximum number of servers that can be selected for the network and is set to one value for the network. The decision variables in the ILP problem are defined as follows. D_U^{\max} denotes the maximum delay between the user and server for the designed network. D_S^{\max} denotes the maximum delay between the server and server for the designed network. 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. x_{kl} is decided for all candidate links. y_i is a binary variable for $i \in V_S$, where $y_i=1$ if server i is selected, and $y_i=0$ otherwise. y_i is decided for all candidate servers.

MHND is formulated by:

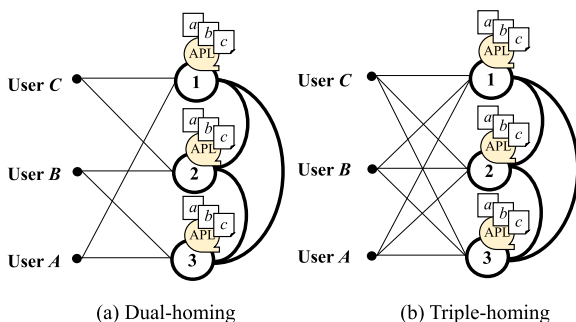


Fig. 3 Multi-homing model between user and server.

Table 1 Given parameters and decision variables used in MHND.

Given parameters	d_{pi}	Delay between user p and server i
	d_{ij}	Delay between server i and server j
	m	Number of servers to which each user belongs (multi-homing number)
	M_i	Maximum number of TEs who belong to server i
	Y_{\max}	Maximum number of servers in the network
Decision variables	D_U^{\max}	Maximum delay between use and server
	D_S^{\max}	Maximum delay between server and server
	x_{kl}	link $(k, l) \in E$ is selected or otherwise $x_{kl} = 1$ is selected, $x_{kl} = 0$ is not selected
	y_i	Server i is selected or otherwise $y_i = 1$ is selected, $y_i = 0$ is not selected

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

$$\text{s.t. } \sum_{i \in V_S} x_{pi} = m, \forall p \in V_U, \quad (2b)$$

$$\sum_{p \in V_U} x_{pi} \leq M_i, \forall i \in V_S, \quad (2c)$$

$$\sum_{i \in V_S} y_i \leq Y_{\max}, \quad (2d)$$

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

$$x_{ij} d_{ij} \leq D_S^{\max}, \forall (i, j) \in E_S, \quad (2f)$$

$$y_i \geq x_{pi}, \forall p \in V_U, i \in V_S, \quad (2g)$$

$$y_i + y_j - 1 \leq x_{ij}, \forall (i, j) \in E_S, \quad (2h)$$

$$x_{ij} \leq y_i, \forall i \in V_S, (i, j) \in E_S, \quad (2i)$$

$$x_{ij} \leq y_j, \forall j \in V_S, (i, j) \in E_S, \quad (2j)$$

$$x_{kl} \in \{0, 1\}, \forall (k, l) \in E, \quad (2k)$$

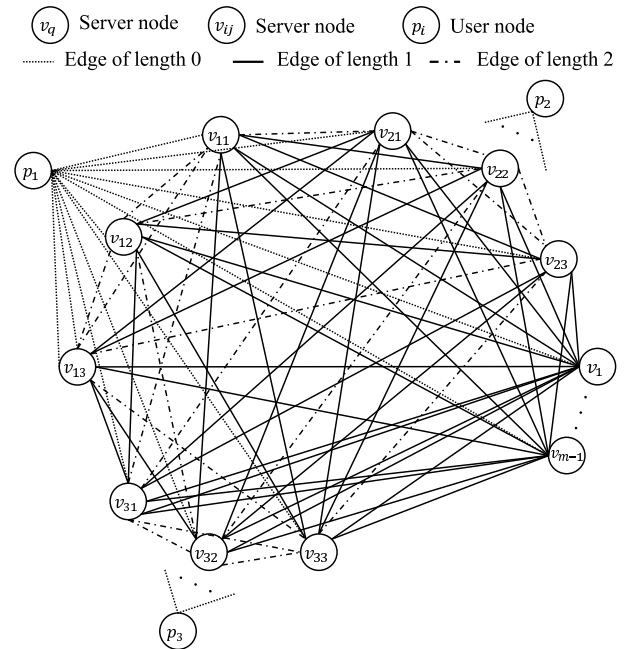
$$y_i \in \{0, 1\}, \forall i \in V_S. \quad (2l)$$

Equation (2a) minimizes the objective function, T_{delay} . Equation (2b) indicates that the sum of the number of user-server links per user is m and that each user belongs to m servers. Equation (2c) indicates that the sum of the number of users belonging to server i is less than or equal to M_i . Equation (2d) indicates that the sum of the number of servers in the network is Y_{\max} or less. Equation (2e) indicates that the maximum value of the delay of the selected user-server link is D_U^{\max} . Equation (2f) indicates that the maximum value of the delay of the selected server-server link is D_S^{\max} . Equation (2g) indicates that if the user-server link $(p, i) \in E_U$ is selected, then the server $i \in V_S$ is also selected. Equations (2h)–(2j) are linear representation of $x_{ij} = y_i \cdot y_j$. Equations (2k)–(2l) show that x_{kl} and y_i are binary decision variables.

3.6 Computational Time Complexity

We analyze the computational time complexity of the distributed server allocation problem with MHND (DSA-MHND). The decision version of DSA-MHND is defined by:

Definition 1: Given a set of servers, V_S , a set of users, V_U ,


Fig. 4 Graph G corresponding to 3-SAT problem with three clauses.

the delay between each pair of a user and a server, the delay between each pair of servers, the capacity of a server, a multi-homing number of m , and a number of h , is it possible to make an assignment of the users to the servers to have the largest maximum delay among selected user-server links $w \leq h$?

Theorem 1: The DSA-MHND problem is NP-complete.

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

We show that the 3-SAT problem, which is known as an NP-complete problem [18], is polynomial-time reducible to the DSA-MHND problem. The 3-SAT problem is stated: given a set of k clauses, each of length three, over a set of x boolean variables, does a satisfying truth assignment exist?

We construct an instance of the DSA-MHND problem from any instance of the 3-SAT problem. Note that this construction is inspired by the proof of NP-completeness for the server allocation problem with preventive start-time optimization against single server failures [7]. The schematic image of the construction is depicted in Fig. 4.

- Create graph G with k user nodes and $3k + m - 1$ server nodes, including k sets of three server nodes v_{ij} , where $i = 1, 2, \dots, k$ and $j = 1, 2, 3$, and $m - 1$ server nodes v_q , where $q = 1, 2, \dots, m - 1$, i.e., $|V_U| = k$ and $|V_S| = 3k + m - 1$.
- We define $V_1 = \{v_{ij} | i = 1, 2, \dots, k, j = 1, 2, \dots, k\}$ and

$V_2 = \{v_q | q = 1, 2, \dots, m-1\}$, where $V_S = V_1 \cup V_2$.

- All server nodes are connected by an edge.
 - For all v_{ij} in V_1 , the length of edge $(v_{ij}, v_{i'j'})$ is set to 1 whenever $i \neq i'$, and the element of v_{ij} and $(v_{ij}, v_{i'j'})$ are not negotiations of each other. In other words, the edge represents two nodes corresponding to elements that have a compatible true assignment.
 - The length of edge (v_q, v_{ij}) is set to 1.
 - Otherwise, the edge length is set to 2.
- Each user node is connected to all server nodes with edge with a length 0.
- Set the capacity of each server in V_1 to 1 and that of each server in V_2 to $|V_U|$, and $h = 1$.

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

Suppose that there is a yes-instance of the 3-SAT problem. We can select k node from $v_{ij} \in V_1$, one corresponding to *true* assignment from each clause, which is all connected in G with edges of length 1. Firstly we assign the k users to the k selected server nodes in V_1 . Secondly, we assign each user to $m-1$ nodes in V_2 . In the assignment, m -homing is achieved and the largest maximum delay w is 1, which satisfies $w \leq h$. Therefore, the DSA-MHND instance is a yes instance.

Conversely, suppose that the DSA-MHND instance is a yes instance. Considering m -homing, each user is connected to one node in V_1 and $m-1$ nodes in V_2 . There is a set of k fully connected server nodes with edges of length at most 1 between two nodes in V_1 . By the definition of graph G , these nodes in V_1 correspond to variables with the compatible *true* assignment. Therefore, the truth assignment that sets the variable corresponding to the k nodes in V_1 to *true* satisfies all the clauses. Thus the 3-SAT problem is a yes instance.

Since the DSA-MHND problem is NP and the 3-SAT problem is polynomial-time reducible to the DSA-MHND problem, the DSA-MHND problem is NP-complete. ■

The problem of multi-homing network design model (NHND) presented in Sect. 3.5 is NP-hard, since the DSA-MHND problem is proved to be NP-complete. No polynomial-time algorithm to an NP-hard problem has been found so far. It is reasonable to solve NHND by using a solver that handles ILP in case that the computation time is acceptable.

4. Numerical Evaluation and Discussion

We evaluate the delay and availability of MHND compared to that of conventional single-homing. First, we evaluate MHND under the condition that 100 and 1000 users are randomly distributed in a square region, and servers are located in the nodes of the Kanto area of the Japan Photonic Network (KJPN) [19] and nodes of COST239 (COST) [20] node, as basic performance for typical networks. Next, we evaluate MHND under the condition that users are located at

the latitude and longitude of all cities in Japan [21], and the server is located at the location of the Japan Photonic Network node [19] as a wide area network (WAN) with realistic user distribution.

We set the delays so that they are independent of how the equipment is set up or how much traffic there is. Since MHND 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 [22], [23]. 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 [24], transmission delay calculation using packet timestamps of network time protocol (NTP) [25] or PTP [17], 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 users and servers; each delay of a logical link in the proposed model reflects the network equipment configuration and traffic conditions.

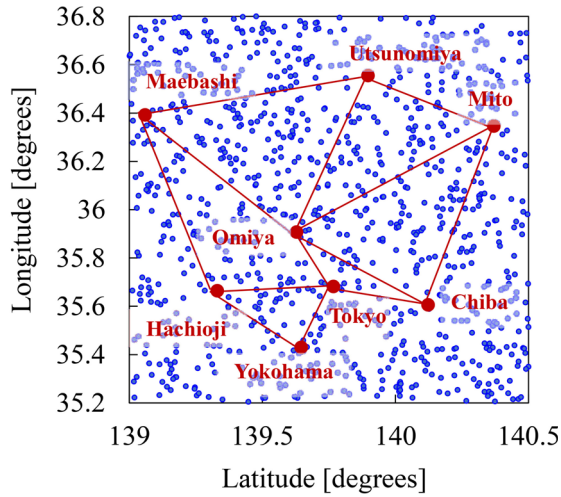
In a commercial network, the transmission distance is not proportional to the delay because traffic passes through transport devices, routers, and switches. Note that, typically, the longer the transmission distance, the greater the number of devices passed through; for this reason, we assume that the delay is proportional to the distance in this evaluation. When a service provider uses the proposed model, the delay should be determined according to the previous paragraph. Based on this assumption, we treat a 100 km transmission delay as a 0.5 ms delay in each link for the evaluation in this paper.

We assume the user and server are connected by a fixed-line. In some use cases, a user may be connected by wireless. When the delay between the user and server is affected by wireless conditions, we target only use cases to prevent delay degradation by managing power supply status, the number of connections, etc.

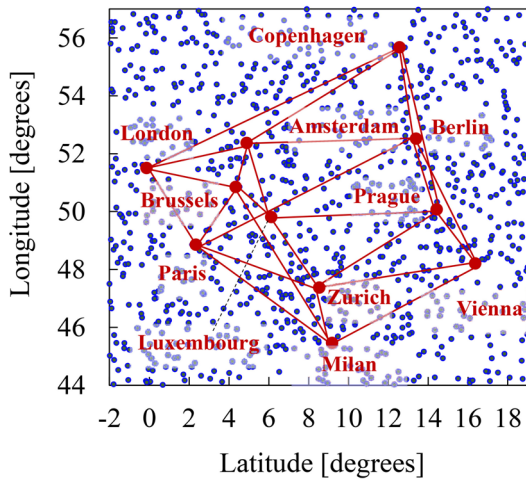
We evaluate the availability of MHND as the availability of service which all users continue to use the application. The availability of service when each user belongs to m servers is expressed as A_m . We assume that the service is available, i.e., it can continue against most failures of $m-1$ servers simultaneously when each user belongs to m servers. p_s represents the unavailability of a server. Therefore, when there are $|V_S|$ servers. A_m can be expressed by:

$$A_m = \sum_{k=0}^{m-1} \binom{|V_S|}{k} p_s^k (1-p_s)^{|V_S|-k}. \quad (3a)$$

In this evaluation, computation time is the average time considered by solving the ILP problem five times. Our evaluations are performed on an Intel(R) Xeon(R) Gold 6132 CPU 2.60GHz, 128 GB memory environment using CPLEX [26].



(a) Location of servers for KJPN.



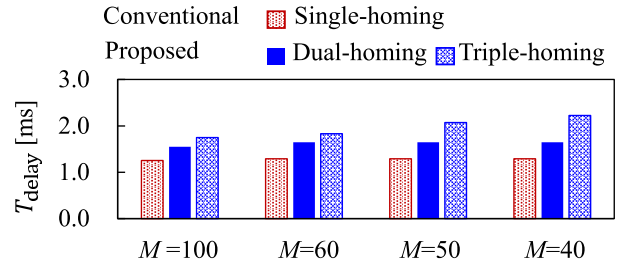
(b) Location of servers for COST.

Fig. 5 User and server location for KJPN and COST.

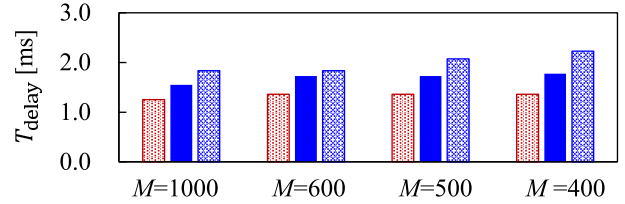
4.1 Basic Performance for Typical Network

Figures 5(a) and (b) show the network for evaluation. Servers are assumed to be located in the nodes of KJPN and nodes of COST, respectively. We assume that all servers are logically connected in a full mesh, e.g., servers in Yokohama and Omiya are connected via a Tokyo node. In KJPN shown in Fig. 5(a), we assume that users are randomly distributed in an area with a longitude of 139 degrees to 140.5 degrees and a latitude of 35.2 degrees to 36.8 degrees. In COST shown in Fig. 5(b), we assume that users are randomly distributed in an area with a longitude of -2.0 degrees to 19.0 degrees and latitude of 44.0 degrees to 57.0 degrees. We assume that users are connected to each server by a linear distance on the coordinate axis. We assume that all servers have the same failure rate of each server, λ , and the same mean time to repair (MTTR) of two hours; the unavailability of a server is expressed as $p_s = \frac{2\lambda}{1+2\lambda}$.

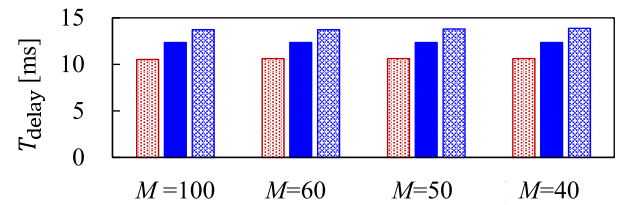
Figures 6(a) and (b) show T_{delay} of single, double, and



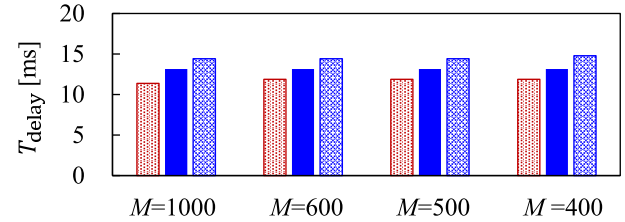
(a) 100 users for KJPN



(b) 1000 users for KJPN



(c) 100 users for COST



(d) 1000 users for COST

Fig. 6 T_{delay} of 100 and 1000 users at single, dual, and triple-homing.

triple-homing for 100 and 1000 users in KJPN, respectively. Figures 6(c) and (d) show T_{delay} of single, double, and triple-homing for 100 and 1000 users in COST, respectively. In this evaluation, we assume the same values for all servers; $M_i = M, \forall i \in V_s$, where $M=100, 60, 50, 40$ for 100 users and $M=1000, 600, 500, 400$ for 1000 users.

For 100 and 1000 users on both networks (KJPN and COST), T_{delay} is worse as the multi-homing number increases. This is because T_{delay} is determined using the link with the largest delay among the multiple user-server links in the case of multi-homing. Regarding the effect of M on the delay, the delay tends to worsen as M decreases. This is because the restriction of M prevents the selection of servers with a lower delay since the user connects to multiple servers in the case of multi-homing.

Figure 7 shows the dependency of server failure rate, λ , on the unavailability, $1-A_m$, at single ($m = 1$), dual ($m = 2$), and triple ($m = 3$) homing. In the single-homing, as the server failure rate increases, the availability of service, A_m ,

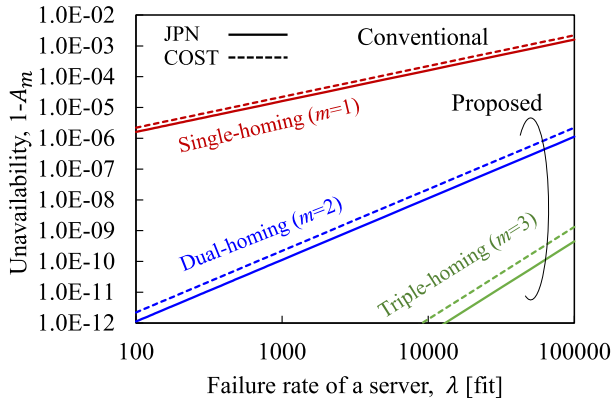


Fig. 7 Server failure rate dependence of Unavailability, $1-A_m$, in single, dual, and triple-homing at MTTR = 2 [hour].

Table 2 Computation times [sec] and variances [sec^2] in 1000 users for KJPN.

	Conventional		Proposed			
	Single		Dual		Triple	
	Time	Var.	Time	Var.	Time	Var.
$M=1000$	3.5	0.002	2.5	0.009	2.1	0.004
$M=600$	4.3	0.035	11.9	0.157	9.3	0.045
$M=500$	7.2	0.035	13.3	0.068	10.2	0.003
$M=400$	9.4	0.156	16.2	0.028	7.9	0.001

Table 3 Computation times [sec] and variances [sec^2] in 1000 users for COST.

	Conventional		Proposed			
	Single		Dual		Triple	
	Time	Var.	Time	Var.	Time	Var.
$M=1000$	5.5	0.002	4.1	0.016	5.9	0.009
$M=600$	6.9	0.005	16.3	0.340	35.4	0.051
$M=500$	8.9	0.009	16.0	0.047	83.8	0.471
$M=400$	10.1	0.233	25.8	0.057	72.6	1.136

decreases more sharply compared to multi-homing with $m \geq 2$. In the dual and triple-homing, the availability of service, A_m remains above 0.99999 (five-nines) even with a server failure rate of 100000 fits. From these results, it is desirable to design a network with the redundancy of dual-homing or more for mission-critical applications to achieve the required availability.

Tables 2 and 3 show the computation time and variances in 1000 users to determine the delay for KJPN and COST, respectively. The computation time is the average value over five trials. The maximum computation time is 16.2 [sec] for KJPN and 83.8 [sec] for COST, respectively. In addition, as M decreases, the computation time increases due to the added restrictions. These computation times are acceptable in practical scenarios for network designing before launching a service.

Numerical results of MHND show that the availability of service is improved with the increase in the multi-homing number while the delay worsens. In particular, when there is no restriction of M in the case of dual-homing, the delay

increase is 1.25 times that of conventional single-homing. At the same time, the availability of service is improved from 0.99998 (four-nines) to 0.999999999 (nine-nines) under the condition of a server failure rate of 1000 [fit], as shown in Fig. 7. If the delay of service is acceptable, high availability is ensured by dual-homing. From these results, MHND is an effective network design model for balancing the delay and availability in service. In addition, the computation time is considered to be a practical time for the network design compared to the time required for application installation, user registration, and preparation before service startup.

4.2 Performance for Wide Area Network (WAN)

We evaluate MHND under the condition that users are located at the latitude and longitude of all cities in Japan [21] and the server is located at the location of the JPN node [19]. Figures 8(a) and (b) show the location of 703 users and the location of servers and the link between two servers, respectively. The location of 48 servers and the distance of each link are based on JPN [19]. We assume that all servers have the same failure rate, λ , and the same MTTR.

The same as Sect. 4.1, we assume that all servers are logically connected in a full mesh and each user is logically connected to all servers. The distance between two servers is treated as the shortest route for the JPN links. The distance between a user and a server is calculated based on the model shown in Fig. 9(a). As shown in Fig. 9(b), from the nearest server to the fifth nearest server, the distance between a user and a server is treated as the linear distance on the coordinate axes. As shown in Fig. 9(c), farther than the sixth nearest server, the distance between a user and a server is treated as the sum of the distance between the user and the nearest server and the distance between the nearest server and the target server. This assumption is based on the following conditions. Typically, each user has accommodated a near exchange office of the network via an optical fiber or Ethernet cable and connects to the distributed server via a carrier network. In addition, we assume that there are five candidate exchange offices where users will be accommodated in this evaluation model. Any logical connection between a user and a server is possible in the carrier network. Considering these conditions, we simply assume that a user can select any server. In this evaluation, we assume that a user can select the nearest five exchange offices of the network for the redundancy routes. We assume the same values for all servers; $M_i = M, \forall i \in V_S$, where $M=703$ and 100 users.

Figures 10(a) and (b) show T_{delay} of WAN at single, double and triple-homing for $Y_{\text{MAX}}=24$ and 48, respectively. Figures 11 and 12 show the dependency of server failure rate, λ , on the unavailability, $1-A_m$, at single ($m=1$), dual ($m=2$), and triple ($m=3$) homing. The evaluation conditions of Fig. 11 is $Y_{\text{MAX}}=24$ and 48, MTTR=2. The evaluation conditions of Fig. 12 is $Y_{\text{MAX}}=48$, MTTR=2 and 12. These results indicate that, as in Sect. 4.1, the availability improves significantly for a slight increase in delay (less than 1.1 times) when the number of multi-homing exceeds

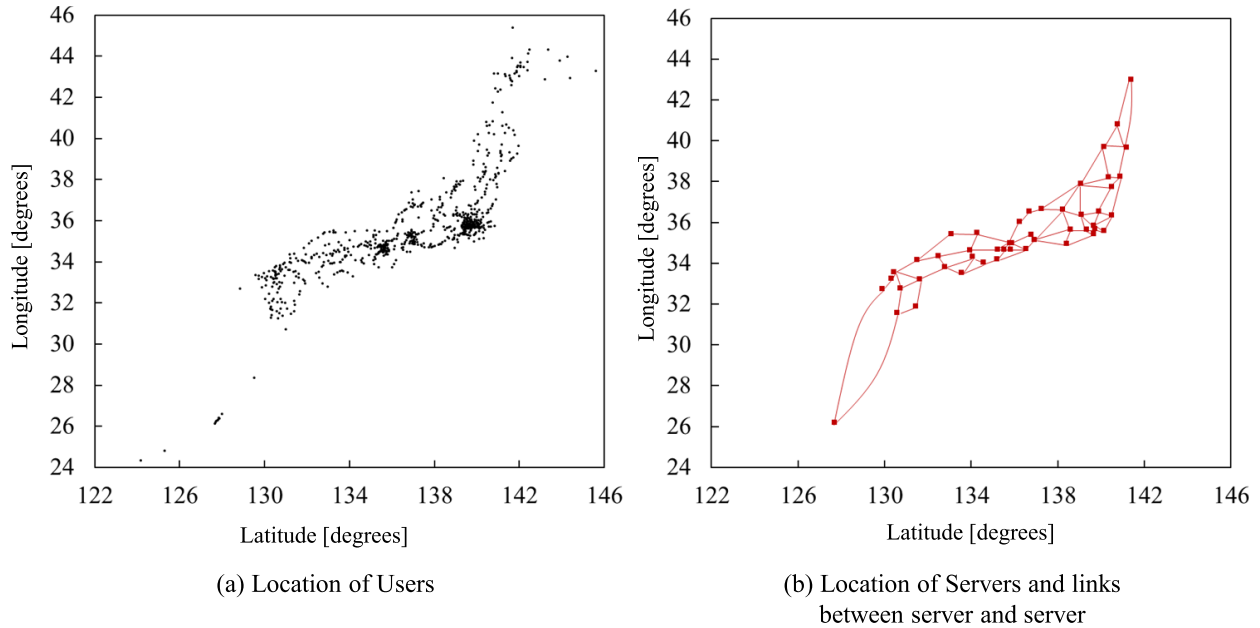


Fig. 8 User and server location for WAN.

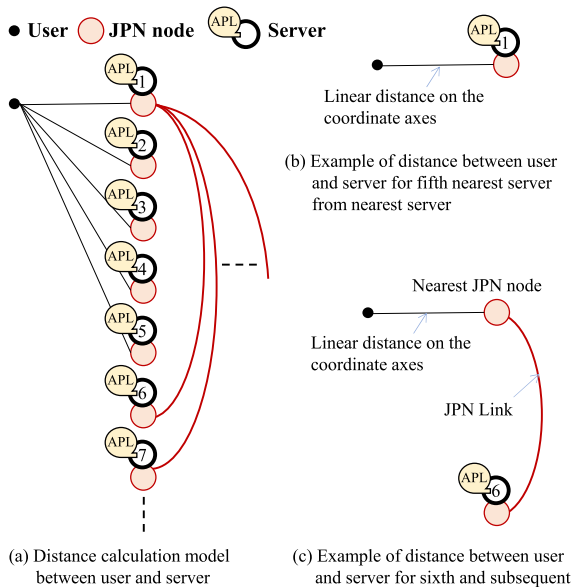


Fig. 9 Distance calculation model between user and server.

dual ($m = 2$). As shown in Fig. 11, it is indicated that increasing the number of multi-homing is expected to improve availability more than reducing the total number of servers (Y_{MAX}). As shown in Fig. 12, it is indicated that the unavailability of dual-homing with $MTTR = 12$ [hour] is less than that of single-homing with $MTTR = 2$ [hour]. For a service provider, $MTTR$ is related to the storage location of maintenance components in case of failure. These results suggest that multi-homing topology can require fewer storage locations for the maintenance components.

Table 4 shows the computation time and variances of WAN at $Y_{MAX}=48$ and 24, respectively. The computation

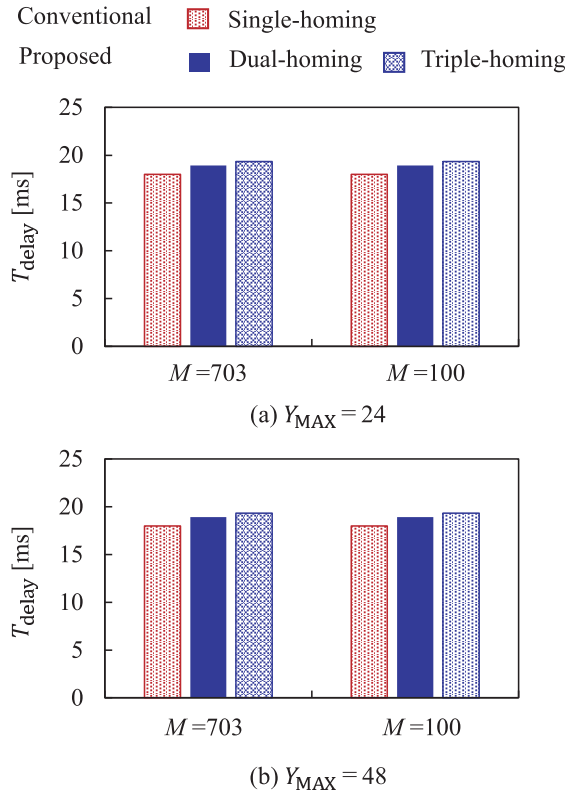


Fig. 10 T_{delay} at single, dual, and triple-homing in WAN.

time is the average value over five trials. The maximum computation time is 7945.8 [sec] (2.2 hours) with the variances of 581.3 [sec²] (the maximum computation time in five trials is 7967.3 [sec]). It is acceptable in practical scenarios for network designing before launching a service, but in scenarios where the service is started as soon as the user

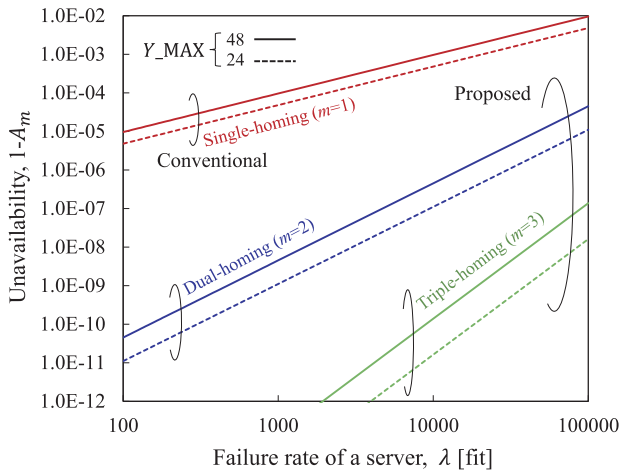


Fig. 11 Server failure rate dependence of unavailability, $1-A_m$, for WAN at $MTTR = 2$ [hour].

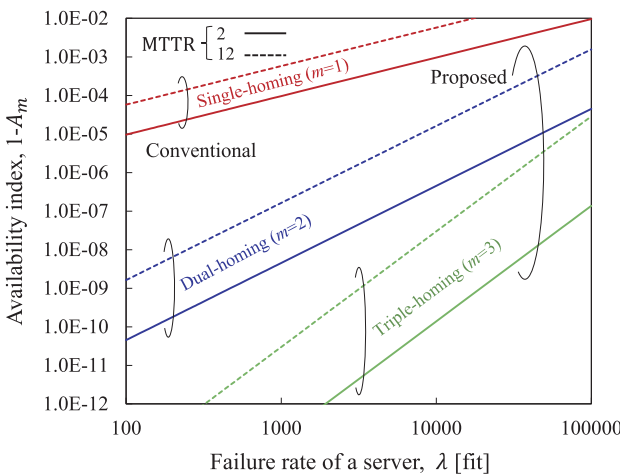


Fig. 12 Server failure rate dependence of unavailability, $1-A_m$, for WAN at $Y_{MAX} = 48$.

Table 4 Computation times [sec] and variances [sec²] of WAN.

	Conventional		Proposed			
	Single		Dual		Triple	
	Time	Var.	Time	Var.	Time	Var.
$Y_{MAX}=48$						
$M=703$	135.2	2.4	1026.6	20.9	1183.7	35.7
$M=100$	1196.9	13.4	4583.6	1864.8	6407.0	1477.2
$Y_{MAX}=24$						
$M=703$	118.5	6.6	248.8	1.0	1398.6	6.6
$M=100$	1943.6	997.1	1778.9	225.2	7945.8	581.3

participation is confirmed, reducing the computation time is an issue to be addressed in the future.

5. Conclusion

In this paper, we proposed a multi-homing network design model named MHND, balancing low delay and high availability in distributed processing of applications. We for-

mulated MHND as an integer linear programming problem. We proved the NP-completeness for the considered problem. Numerical results indicated that two or more multi-homing networks designed with MHND can achieve more than an order of magnitude higher availability compared to that of conventional single-homing network. The evaluation of a wide area network has shown that the proposed model can be used to design networks considering the delay, network availability, MTTR (related to the location of the maintenance parts), and the number of servers to be utilized.

Acknowledgments

This work was supported by JSPS KAKENHI JP23K16867. 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] "NTT DOCOMO to Launch 5G Service in Japan on March 25," https://www.nttdocomo.co.jp/english/info/media_center/pr/2020/03_18_00.html, online; accessed 16 Feb. 2023.
- [2] A. Kawabata and Y. Aoyagi, "Network-service technology enabled by the all-photonics network," *NTT Technical Review*, vol.19, no.10, pp.18–24, 2021.
- [3] R.M. Fujimoto, *Parallel and Distributed Simulation Systems*, Wiley, New York, NY, USA, 2000.
- [4] A. Kawabata, B.C. Chatterjee, and E. Oki, "MHND: Multi-homing network design model for delay sensitive distributed processing applications," *IEEE Consumer Communications & Networking Conference (CCNC)*, pp.348–353, 2023.
- [5] D.R. Jefferson, "Virtual time," *ACM Trans. Program. Lang. Syst. (TOPLAS)*, vol.7, no.3, pp. 404–425, 1985. DOI: 1145/3916.3988
- [6] A. Kawabata, B.C. Chatterjee, S. Ba, and E. Oki, "A real-time delay-sensitive communication approach based on distributed processing," *IEEE Access*, vol.5, pp.20235–20248, 2017. DOI: 10.1109/ACCESS.2017.2758803
- [7] S. Yanase, S. Masuda, F. He, A. Kawabata, and E. Oki, "Heuristic approach to distributed server allocation with preventive start-time optimization against server failure," *IEICE Trans. Commun.*, vol.E104-B, no.8, pp.942–950, Aug. 2021. DOI: 10.1587/transcom.2020EBP3145
- [8] E. Cronin, B. Filstrup, A.R. Kurc, and S. Jamin, "An efficient synchronization mechanism for mirrored game architectures," *1st Workshop on Network and System Support for Games*, pp.67–73, 2002. DOI: 10.1145/566500.566510
- [9] F. Chiariotti, S. Kucera, A. Zanella, and H. Claussen, "Analysis and design of a latency control protocol for multi-path data delivery with pre-defined QoS guarantees," *IEEE/ACM Trans. Netw.*, vol.27, no.3, pp.1165–1178, 2019. DOI: 10.1109/TNET.2019.2911122
- [10] R. Gupta, D. Reebadiya, and S. Tanwar, "6G-enabled edge intelligence for ultra-reliable low latency applications: Vision and mission," *Computer Standards & Interfaces*, vol.77, p.103521, 2021. DOI: 10.1016/j.csi.2021.103521
- [11] M.S. Elbamby, C. Perfecto, C.F. Liu, J. Park, S. Samarakoon, X. Chen, and M. Bennis, "Wireless edge computing with latency and reliability guarantees," *Proc. IEEE*, vol.107, no.8, pp.1717–1737, 2019. DOI: 10.1109/JPROC.2019.2917084
- [12] S. Yang, F. Li, R. Yahyapour, and X. Fu, "Delay-sensitive and availability-aware virtual network function scheduling for NFV," *IEEE Trans. Services Comput.*, vol.15, pp.188–201, 2019. DOI: 10.1109/TSC.2019.2927339
- [13] K. Intharawijitr, K. Iida, H. Koga, and K. Yamaoka, "Simulation

study of low-latency network model with orchestrator in MEC,” *IEICE Trans. Commun.*, vol.E102-B, no.11, pp.2139–2150, Nov. 2019. DOI: 10.1587/transcom.2018EBP3368

- [14] L. Gu, D. Zeng, S. Guo, A. Barnawi, and Y. Xiang, Y, “Cost efficient resource management in fog computing supported medical cyber-physical system,” *IEEE Trans. Emerg. Topics Comput.*, vol.5, no.1, pp.108–119, 2055. DOI: 10.1109/TETC.2015.2508382
- [15] H.H. Liu, Y. Wang, Y.R. Yang, H. Wang, and C. Tian, C, “Optimizing cost and performance for content multihoming,” *Proc. ACM SIGCOMM 2012 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, pp.371–382, 2012.
- [16] H. Wang, G. Tang, K. Wu, and J. Fan, “Speeding up multi-CDN content delivery via traffic demand reshaping,” *2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS)*, pp.422–433, 2018. DOI: 10.1109/ICDCS.2018.00049
- [17] Technical Committee, “IEEE standard for a precision clock synchronization protocol for networked measurement and control systems,” *IEEE Std.*, pp.1588–2008, 2008.
- [18] R.M. Karp, “Reducibility among combinatorial problems,” *Complexity of Computer Computations*, pp.85–103, Springer, 1972.
- [19] “Japan Photonic Network Model,” <http://www.ieice.org/cs/pn/jpn/jpnm.html>, online; accessed 16 Feb. 2023.
- [20] M. O’Mahony, “Ultrahigh capacity optical transmission network: European research project cost 239,” *Information, Telecommunications, Automata Journal*, vol.12, pp. 33–45, 1993.
- [21] “National Database of Latitude and Longitude Locations of Prefectures, Cities, Towns, and Villages,” <https://www.vector.co.jp/soft/dl/data/home/se156040.html> (Japanese), online; accessed 16 Feb. 2023.
- [22] “Global IP Network,” <https://www.ntt.com/en/services/network/gin/sla.html>, online; accessed 16 Feb. 2023.
- [23] “Service Level Agreements of NTT WEST,” <https://www.ntt-west.co.jp/business/solution/sp/04> (Japanese), online; accessed 16 Feb. 2023.
- [24] J. Postel, “Internet control message protocol,” RFC 792, 1981.
- [25] D. Mills, J. Burbank, and W. Kasch, “Network time protocol version 4: Protocol and algorithms specification,” RFC 5905, 2010.
- [26] “CPLEX Optimization Studio,” https://www.ibm.com/products/ilog-cplex-optimization-studio?mhsrc=ibmsearch_a&mhq=cplex, online; accessed 16 Feb. 2023.



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.



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.