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

Akio Kawabata
Toyohashi University of Technology
Aichi, Japan
kawabata.akio.oc@tut.jp

Bijoy Chand Chatterjee
South Asian University
New Delhi, India
bijoycc@ieee.org

Eiji Oki
Kyoto University
Kyoto, Japan
oki@i.kyoto-u.ac.jp

***Abstract*—When mission-critical applications are provided over a network, high availability is required in addition to a low delay network. This paper proposes a multi-homing network design model, named MHND, to balance a low delay and high availability when distributed processing applications use multiple processing servers. 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. Two or more multi-homing can achieve approximately an order of magnitude higher availability compared to that of the conventional single-homing at the expense of a delay increase of 1.25 times. By using MHND, flexible network design is achieved based on the acceptable delay in service and the required availability.**

***Index Terms*—Delay sensitive service, network design, availability, distributed computing, conservative synchronization**

I. 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 telemedicine and network-based autonomous driving, which were previously difficult to achieve, to be provided via networks.

When mission-critical applications are provided via a network, high availability is required in addition to a low delay network. In particular, redundancy must be ensured so that a single failure does not render an application unavailable. Our work aims to realize a network design model to balance end-to-end delay (delay) and high availability when an application is being processed by multiple servers located in a wide area network.

When providing services using multiple servers distributed over a wide area, an optimal network configuration of multiple servers is an issue. Furthermore, these applications require fairness of delay regardless of the user's location, such as an unfair larger delay due to being far from the application server.

Research works that guarantee the order of events have been studied in parallel and distributed processing 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 [4] 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 [5]. The work in [6] 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. The works mentioned above [3]–[5] 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 availability of service is essential and typically can be improved with the increase in the multi-homing. The work in [7] introduced a service function chaining for virtualized network functions considering delay and availability. The work is effective in providing a service using multiple virtualized functions but the fairness of events among users is not considered.

It is desirable to design a network with the redundancy of dual-homing or more for mission-critical applications that require high availability. A question grabs our attention: how can we provide higher availability of service continuity in user-server link and server failures under the condition of acceptable delay?

To address the above question, this paper, for the first time, proposes a multi-homing network design model, named MHND, with multi-homing configurations. In MHND, each user belongs to multiple servers so that applications can

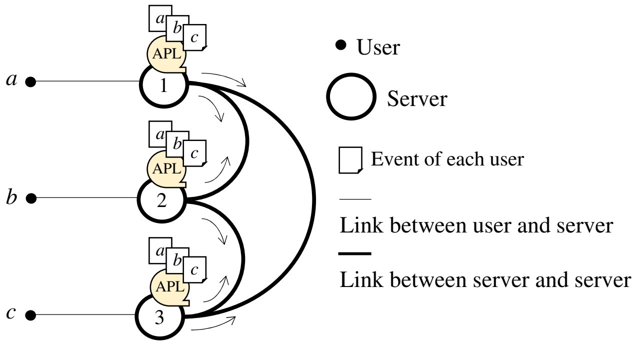


Fig. 1. Communication and processing model between servers.

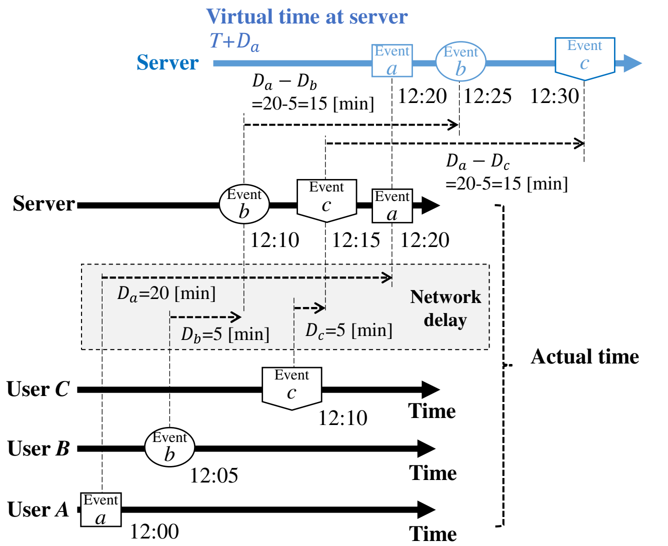


Fig. 2. Example of correcting order of events between user and server.

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 the delay and service availability by solving the ILP problem of MHND and comparing it to the conventional single-homing. Numerical results indicate that MHND can be used for network designing considering delay and availability, and it is effective 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 rest of the paper is organized as follows. Section II presents the prerequisite of the proposed model, its formulation as an optimization problem, and its computational time complexity. In Section III, we evaluate the proposed model in terms of delay, availability, and computation time for two types of networks. Finally, Section IV concludes this paper.

II. Proposed Model

A. 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, it is assumed that user events are multicasted between servers for distributed processing. Each server processes the events of all users. In other words, all users' events are processed in parallel at all servers.

B. Prerequisite of guaranteeing order of events

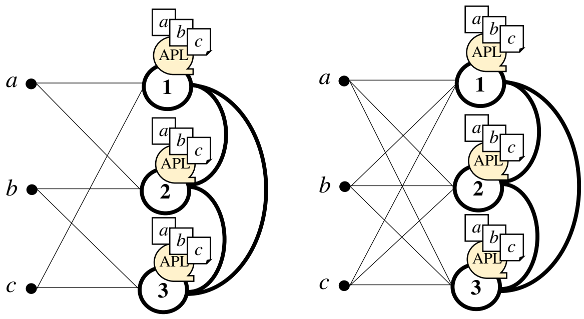
The concept of virtual time is introduced to guarantee the order of events, and at each server, events for all users are rearranged by virtual time in the order in which they occur [5]. Figure 2 shows the order guarantee of events using virtual time. Events a, b, and c occur at 12:00, 12:05, and 12:10, 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 is $D_{\mathrm{U}}^{\max}$, the user-server event correction is performed at $T + D_{\mathrm{U}}^{\max}$.

Similarly to the user-server event correction, order correction is performed for server-server multicast communication. If the maximum server-server delay is $D_{\mathrm{S}}^{\max}$, the server-server event correction is performed at $T + D_{\mathrm{S}}^{\max}$. 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 $T + D_{\mathrm{U}}^{\max} + D_{\mathrm{S}}^{\max}$. When sending the processing result of each server to the user is sent, the maximum delay between user and server, $D_{\mathrm{U}}^{\max}$, is the queuing process for the network delay. Fair application processing is achieved at the time of $T + 2D_{\mathrm{U}}^{\max} + D_{\mathrm{S}}^{\max}$ for all users, and the delay, T_{delay}, is expressed as follows:

$$T_{\mathrm{delay}} = 2D_{\mathrm{U}}^{\max} + D_{\mathrm{S}}^{\max}. \quad (1a)$$

C. Prerequisite of multi-homing

In the communication and processing of Section II-A, 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.



(a) Dual-homing (b) Triple-homing

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

D. Communication and processing of multi-homing

Each user belongs to multiple servers, as described in Section II-C. All events are multicasted between servers and each server processes the events of all users, as described in Section II-A. The event order is guaranteed by using conservative synchronization, as described in Section II-B. 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, all events are multicasted to all servers. In examples of Fig. 3, user a sends the same event to servers 1 and 2. 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) [8]. This time information and user information are used to discard duplicate events.

E. 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. $V_{\mathrm{U}} \subseteq V$ denotes the set of users, and $V_{\mathrm{S}} \subseteq V$ denotes the set of servers. $V_{\mathrm{U}} \cap V_{\mathrm{S}} = \emptyset$ and $V_{\mathrm{U}} \cup V_{\mathrm{S}} = V$. $E_{\mathrm{U}} \subseteq E$ denotes the set of user-server links, and $E_{\mathrm{S}} \subseteq E$ denotes the set of server-server links. $E_{\mathrm{U}} \cap E_{\mathrm{S}} = \emptyset$ and $E_{\mathrm{U}} \cup E_{\mathrm{S}} = E$. A link between user $p \in V_{\mathrm{U}}$ and server $i \in V_{\mathrm{S}}$ is expressed as $(p, i) \in E_{\mathrm{U}}$, and a link between server $i \in V_{\mathrm{S}}$ and server $j \in V_{\mathrm{S}}$ is expressed as $(i, j) \in E_{\mathrm{S}}$.

The given parameters are defined as follows. Let d_{pi} and d_{ij} denote the delay of the user-server link $(p, i) \in E_{\mathrm{U}}$ and server-server link $(i, j) \in E_{\mathrm{S}}$, respectively. The number of servers to which each user belongs, which we call a multi-homing number, is expressed as m. This means that each user selects m servers from $|V_{\mathrm{S}}|$ servers. Let M_i be the maximum number of users that server i can accommodate. Let $Y_{\max}$ be the maximum number of servers in the network.

The decision variables are defined as follows. Let x_{kl} expresses whether the link $(k, l) \in E$ is selected, $x_{kl} = 1$ if it is selected, $x_{kl} = 0$ if it is not selected. Let y_i expresses whether server $i \in V_{\mathrm{S}}$ is selected, $y_i = 1$ if it is selected, and $y_i = 0$ if it is not selected.

MHND is formulated by:

$$\text{Objective} \quad \min \quad 2D_{\mathrm{U}}^{\max} + D_{\mathrm{S}}^{\max}, \tag{2a}$$

$$\text{s.t.} \quad \sum_{i \in V_{\mathrm{S}}} x_{pi} = m, \forall p \in V_{\mathrm{U}}, \tag{2b}$$

$$\sum_{p \in V_{\mathrm{U}}} x_{pi} \leq M_i, \forall i \in V_{\mathrm{S}}, \tag{2c}$$

$$\sum_{i \in V_{\mathrm{S}}} y_i \leq Y_{\max}, \tag{2d}$$

$$x_{pi} d_{pi} \leq D_{\mathrm{U}}^{\max}, \forall (p, i) \in E_{\mathrm{U}}, \tag{2e}$$

$$x_{ij} d_{ij} \leq D_{\mathrm{S}}^{\max}, \forall (i, j) \in E_{\mathrm{S}}, \tag{2f}$$

$$y_i \geq x_{pi}, \forall p \in V_{\mathrm{U}}, i \in V_{\mathrm{S}}, \tag{2g}$$

$$y_i + y_j - 1 \leq x_{ij}, \forall (i, j) \in E_{\mathrm{S}}, \tag{2h}$$

$$x_{ij} \leq y_i, \forall i \in V_{\mathrm{S}}, (i.j) \in E_{\mathrm{S}}, \tag{2i}$$

$$x_{ij} \leq y_j, \forall j \in V_{\mathrm{S}}, (i, j) \in E_{\mathrm{S}}, \tag{2j}$$

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

$$y_i \in \{0, 1\}, \forall i \in V_{\mathrm{S}}. \tag{2l}$$

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 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 $D_{\mathrm{U}}^{\max}$. Equation (2f) indicates that the maximum value of the delay of the selected server-server link is $D_{\mathrm{S}}^{\max}$. Equation (2g) indicates that if the user-server link $(p, i) \in E_{\mathrm{U}}$ is selected, then the server $i \in V_{\mathrm{S}}$ 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.

F. 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}, 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

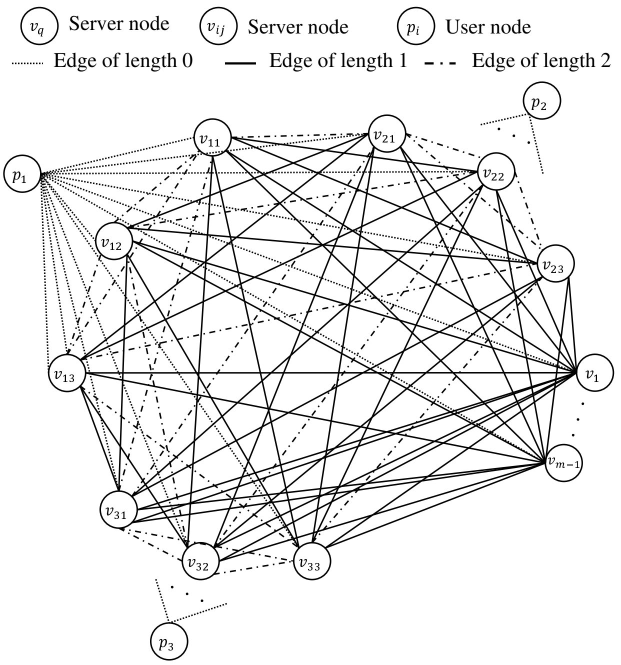


Fig. 4. Graph G corresponding to 3-SAT problem with three clauses.

to each server in V_{S} and compute the maximum delay between users and servers, D_{U}, in $O(|V_{\mathrm{U}}|)$. We compute the maximum delay between servers, D_{S}, in $O(|V_{\mathrm{S}}|^2)$. Then, we compute w, and verify if w is at most h in $O(1)$. Therefore, the overall time complexity is $O(|V_{\mathrm{S}}|^2 + |V_{\mathrm{U}}|))$.

We show that the 3-SAT problem, which is known as an NP-complete problem [9], 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 [6]. 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, $\cdots$, k and j = 1, 2, 3, and $m-1$ server nodes v_q, where $q = 1, 2, \cdots, m-1$, i.e., $|V_{\mathrm{U}}| = k$ and $|V_{\mathrm{S}}| = 3k+m-1$.
- We define $V_1 = \{v_{ij} | i = 1, 2, \cdots, k\}$ and $V_2 = \{v_q | q = 1, 2, \cdots, m-1\}$, where $V_{\mathrm{S}} = V_1 + V_2$.
- 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 $|V_{\mathrm{U}}|$, 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.

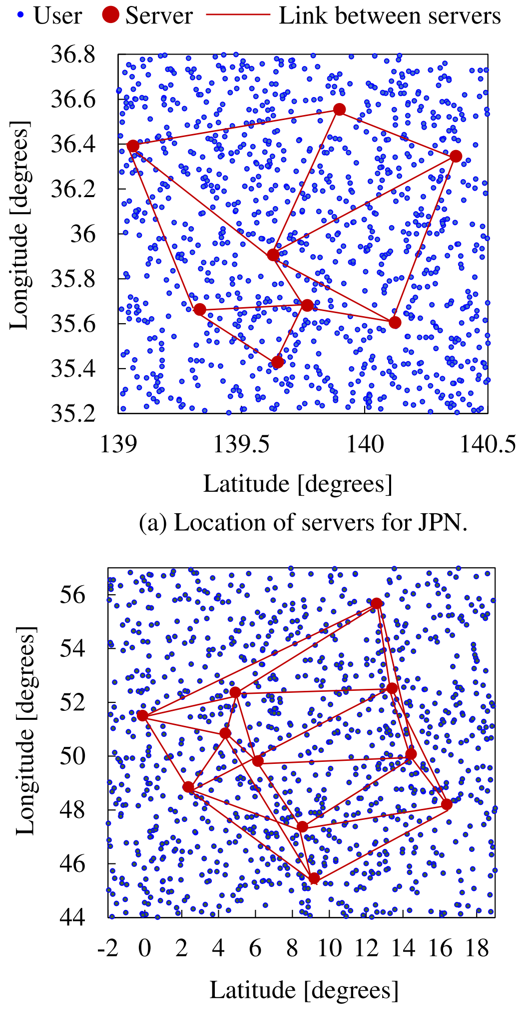
Since the DSA-MHND problem is NP and the 3-SAT problem is polynomial-time reducible to the DSA-MHND problem, the DSA-MHND problem is NP-complete.

■

III. Numerical evaluation and discussion

The following are the prerequisites for the evaluation of MHND. We assume that a network delay of a link is proportional to the length of a link. The delay is determined by the total length of the links, assuming no server processing delay. We note that, since the proceeding delay at two nodes is always included regardless of network design, we focus on the delay of links in the network design for simplicity. In addition, the reason for focusing only on link transmission delay is to determine optimal servers for network design regardless of traffic congestion and server processing load. Moreover, traffic congestion and server processing performance vary with user usage and can be resolved by increasing the service provider's equipment resources, especially using virtualization technologies. On the other hand, the selected servers are determined based on where the servers are deployed, and the location of servers cannot be easily changed by the service provider.

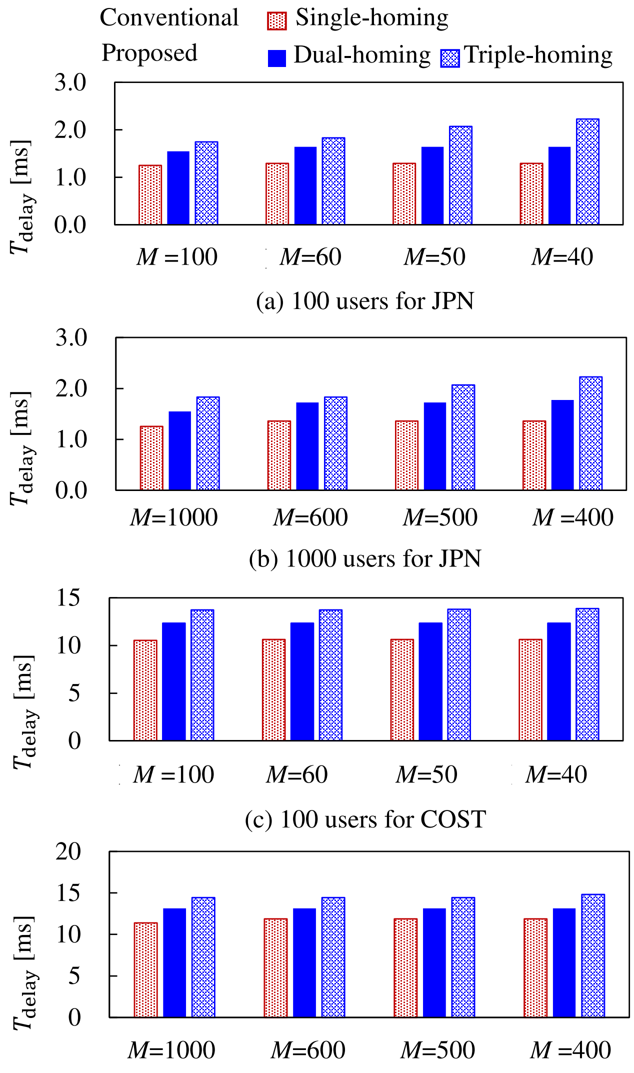
Figures 5(a) and (b) show the network for evaluation. Servers are assumed to be located in the nodes of Kanto area of Japan Photonic Network (JPN) [10] and nodes of COST239 (COST) [11] node, respectively. We assume that all servers are logically connected in a full-mesh, e.g., servers in Yokohama and Omiya are connected via a Tokyo node. In the JPN network 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 the COST network shown in Fig. 5 (b), we assume



(a) Location of servers for JPN.

(b) Location of servers for COST.

Fig. 5. User and server location for JPN Kanto region and COST.



(a) 100 users for JPN

(b) 1000 users for JPN

(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.

that users are randomly distributed in an area with a longitude of -2.0 degrees to 19.0 degrees and a 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 consider 1000 users and 100 users. Our evaluations are performed on an Intel(R) Core(TM) i5-10210U CPU 1.60GHz, 16 GB memory environment using SCIP [12].

Figures 6(a) and (b) show the delay of T_{delay} for single, double and triple-homing for 100 and 1000 users in JPN, respectively. Figures 6(c) and (d) show T_{delay} for single, double and triple-homing for 100 and 1000 users in COST, respectively. In this evaluation, we assume the same values for all servers; $M_i = M, \forall i \in V_{\text{S}}$, 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 (JPN 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.

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 $|V_{\text{S}}|$ servers. A_m can be expressed by:

$$A_m = \sum_{k=0}^{m-1} \binom{|V_{\text{S}}|}{k} p_{\text{s}}^k (1-p_{\text{s}})^{|V_{\text{S}}|-k}. \tag{3a}$$

In this evaluation, we assume that all servers have the same failure rate of each server, λ, and the same mean time to

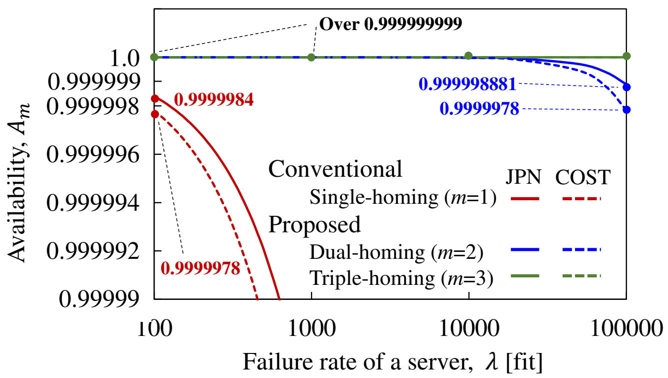


Fig. 7. Server failure rate dependence of service availability in single, dual, and triple-homing at MTTR = 2 [hour].

repair (MTTR) of two hours; the unavailability of a server is expressed as $p_s=\frac{2\lambda}{1+2\lambda}$. Figure 7 shows the dependency of server failure rate, λ, on the availability of service, 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, 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 fit. 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.

Table I shows the computation time in 1000 users to determine the delay for Figs. 6(b) and (d). The computation time is the average value over five trials. The maximum computation time is 4.4 [min] for JPN and 40.9 [min] 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.

TABLE I
COMPUTATION TIMES OF EACH SCHEME IN 1000 USERS.

		Conventional	Proposed	
		(Single) [sec]	(Dual) [sec]	(Triple) [sec]
JPN	M=100	103.3	76.3	157.2
	M=60	84.2	148.8	111.7
	M=50	98.7	183.6	264.8
	M=40	171.0	232.8	180.9
COST	M=1000	84.7	247.7	187.1
	M=600	126.4	1317.0	2452.4
	M=500	153.1	494.2	698.4
	M=400	275.5	615.6	1134.3

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 the 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.

IV. 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 using multiple servers. We formulated MHND as an integer linear programming problem. We proved the NP-completeness for the considered problem. Numerical results, using two types of networks, indicate that, with the increase of multi-homing number, the availability is improved at the cost of delay degradation. Two or more multi-homing networks designed with MHND can achieve more than an order of magnitude higher availability compared to that of the conventional single-homing network. In case that each server has no capacity restriction on the number of accommodated users, the delay of MHND is increased by 1.25 times compared to that of the single-homing network. Therefore, MHND can effectively design networks based on the acceptable delay and required availability in service.

References

[1] "NTT DOCOMO to Launch 5G Service in Japan on March 25," https://www.nttdocomo.co.jp/english/info/media_center/pr/2020/0318_00.html, online; accessed 10 April 2022.

[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. New York, NY, USA: Wiley, 2000.

[4] D. R. Jefferson, "Virtual time," *ACM Transactions on Programming Languages and Systems (TOPLAS)*, vol. 7, no. 3, pp. 404–425, 1985.

[5] 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.

[6] 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 Transactions on Communications*, vol. E104-B, no. 8, pp. 942–950, 2021.

[7] S. Yang, F. Li, R. Yahyapour, and X. Fu, "Delay-sensitive and availability-aware virtual network function scheduling for NFV," *IEEE Transactions on Services Computing*, vol. 15, pp. 188–201, 2019.

[8] Technical Committee, "IEEE standard for a precision clock synchronization protocol for networked measurement and control systems," *IEEE Std*, pp. 1588–2008, 2008.

[9] R. M. Karp, Reducibility among Combinatorial Problems. Springer, 1972.

[10] "Japan Photonic Network Model," http://www.ieice.org/cs/pn/jpn/jpnm.html, online; accessed 10 April 2022.

[11] M. O'MAHONY, "Ultrahigh capacity optical transmission network: European research project cost 239," *Information, Telecommunications, Automata Journal*, vol. 12, pp. 33–45, 1993.

[12] "Solving Constraint Integer Programs (SCIP)," http://scip.zib.de, online; accessed 10 April 2022.