Optimal Construction of Service Function Chains Based on Security Level for Improving Network Security

Dhanu Dwiardhika1,2, Takuji Tachibana1
1Graduate School of Engineering, University of Fukui, Fukui, Japan.
2Center for Informatics and Nuclear Strategic Zone Utilization, National Nuclear Energy Agency of Indonesia, Jakarta, Indonesia.

Corresponding author: Dhanu Dwiardhika (e-mail: dhanud@u-fukui.ac.jp).

ABSTRACT In this paper, in order to improve network security, we propose an optimal placement of security virtual network function (security VNF) for service function chains based on the security level. In this method, an optimization problem is formulated for constructing service function chains based on the security level, and its near-optimal solution is derived with genetic algorithm. From the derived solution, security VNFs are placed to satisfy the security level of each service function chaining and many service function chains can be constructed to obtain higher revenue and less cost while considering the security level. We evaluate the performance of our proposed method with simulation, and the performance of this method is compared with the performance of other methods where security VNFs are selected based on those CPU resources. Numerical examples show that the proposed method is effective to construct service function chains by using security VNFs in order to improve network security.

INDEX TERMS network security, service function chain, virtual network function

I. INTRODUCTION Nowadays, by using virtualization technology, new network architectures, protocols, and applications can be utilized more feasibly in a substrate network [1, 2, 3, 4]. For example, multiple virtual machines can be implemented in a physical server [5, 6, 7], and network function such as load-balancing, router, and firewalls can be implemented as a virtual machine in the physical server [8, 9, 10, 11]. Such a virtual machine is called a virtual network function (VNF). Utilization of VNFs in the substrate network can deploy several kinds of services flexibly [4, 12, 13], and network resources can be utilized effectively [14]. Therefore, virtualization technology can decrease capital expenditure (CAPEX) and operating expenditure (OPEX) significantly [2, 12, 15, 16, 17].

Moreover, recently, cyber-attack such as Stuxnet, Havex, and Dragonfly results in huge financial losses and reputational losses [18]. Network security devices, such as firewall and deep packet inspector are used widely to avoid the security risk [19]. Here, each security device is developed to resolve a specific security problem [20], and hence multiple security devices have to be utilized simultaneously for improving security on the substrate network [20, 21, 22]. In order to process network traffic in multiple security devices, multiple security functions can be expected to be implemented as VNF in a physical server. Such security VNFs can also reduce both CAPEX and OPEX while improving security in the substrate network.

Here, service function chaining is expected to be used for steering network traffic to VNFs in an appropriate order [11, 12, 23, 24, 25]. This allows the substrate network to accommodate several kinds of applications where network functions are used in different combinations. As a result, for security VNFs, the service function is also indispensable in the future. However, as far as the authors know, the utilization of service function chaining with security VNFs for improving network security has not been considered in detail.

For utilizing service function chaining appropriately, mathematical approach such as optimization problem should be considered and studied. In [5], an indicator of standard protection, which is called security level, has been considered. The security level is effective to embed virtual nodes into substrate nodes, and it can be used in the mathematical approach for security VNFs. On the other hand, mathematical approaches for service function chaining with
security VNFs have not been fully considered to improve network security.

In this paper, we propose an optimal construction of service function chains based on the security level to improve network security. In this method, for each service function chain, security VNFs are selected to satisfy the security level of each service function chain, and the selected security VNFs are placed appropriately in substrate nodes. Here, the selection and placement of security VNFs are performed with our formulated optimization problem. By using our proposed method, many service function chains can be constructed in the substrate network by considering security level. We evaluate the performance of our proposed method with simulation, and we investigate the effectiveness of this method by comparing with other methods.

Our proposed method considers the security level of each security VNF according to [26], where virtual networks are constructed based on [5]. However, [26] has not focused on service function chaining. Therefore, we believe that this paper is the first research on service function chaining with security VNFs by considering security level. The contributions of this paper are in the following:

1. This paper extends the concept in [5] of security level to be used for constructing service function chains,
2. This paper introduces the collaboration of security level of VNF and service function chaining to improve network security.

The rest of this paper is organized as follows. Section II introduces related works about the consideration of security level in virtual networks. Section III explains our system model. Section IV explains our proposed service function chain construction with an optimization problem. Section V shows numerical examples, and finally, Section VI denotes the conclusion.

II. RELATED WORK

Optimal construction of service function chain has been considered in the literature. Each construction method has considered one or more of the following objectives: throughput [27, 28], rejection rate [29], latency [30, 31, 32], and resources utilization [4, 7, 24]. These methods have formulated optimization problems and those optimal solutions are derived with a meta-heuristic algorithm and/or heuristic algorithm.

In terms of the security requirements, virtual network embedding has been considered in the literature [5, 26]. [5] has introduced a security level to indicate standard protection. The security level is assigned for each substrate node and virtual node, and those values can be determined by the network operator and the user. A substrate node with higher security level has higher level protection mechanism for embedding virtual nodes. For example, the security level of substrate node with data encryption is higher than that of substrate node without data encryption. A virtual network can be embedded in the substrate network if the security level of the substrate network is equal to or higher than that of the virtual network. [26] has also considered the construction of virtual networks based on the security level for network security. In this construction, security VNFs are added in some substrate nodes for increasing those security level to satisfy the security level of each virtual network.

As far as the authors know, there is no research about the optimal construction of service function chain based on network security. The security level has not been utilized for the construction of service function chains yet. As a result, our paper is different from other methods for service function chaining.

![System model](image)

**Figure 1. System model.**

III. SYSTEM MODEL

In this section, we explain a system model that has extended the model in [26]. Note that in the model, only security VNFs are utilized and other types of VNFs such as routers and load balancing are not utilized. This is because the performance of our proposed method on network security is not changed significantly even if other types of VNFs are considered.

In our system model, a substrate network is modeled as an undirected weighted graph $G^S = (N^S, L^S, A^S_W, A^S_E)$. Figure 1 shows our system model where $N^S$ is a set of all substrate nodes and $L^S$ is a set of all substrate links. For the substrate node $n^s$, $cpu(n^s)$ is the maximum capacity of available CPU and $F(n^s)$ is a set of available security VNFs. On the other hand, $bw(l^S)$ denotes the bandwidth of substrate link $l^S$. From this information, the following sets denote attributes of all substrate nodes and all substrate links:

\[ A^S_W = \{cpu(n^s), F(n^s)|n^s \in N^S\}, \]
\[ A^S_E = \{bw(l^S)|l^S \in L^S\} . \]

The $j$th security VNF is denoted as $f_j$, and the performance of $f_j$ is denoted as $f_j = \{\sigma_j, \zeta_j\}$.

Here, $\sigma_j$ is the amount of CPU capacity that is needed for utilizing $f_j$ and $\zeta_j$ is the security level of $f_j$. When $f_j$ is
placed in \( n^2 \), the amount of available CPU in \( n^2 \) decreases by \( \sigma_j \). Moreover, when \( \zeta_j \) for security VNF \( f_j \) is larger than \( \zeta_k \) for security VNF \( f_k \), \( \sigma_j \) is larger than \( \sigma_k \). This is because a larger amount of CPUs are needed for providing higher security. Note that each VNF can be shared by multiple service function chains, and note that the amount of CPU resources for a VNF does not change even if multiple service chains share the VNF.

On such a substrate network, service function chains are constructed according to each user’s request. Here, the \( i \)th request of service function chain is denoted as

\[
G_i^V = \{\text{dur}^V_i, \text{src}^V_i, \text{dst}^V_i, L_i^V, \text{lev}^V_i, \text{cap}^V_i\},
\]

where \( \text{dur}^V_i \) is the utilization time of this service function chain, \( \text{src}^V_i \) and \( \text{dst}^V_i \) are its source node and the destination node, respectively. \( L_i^V \) is a set of substrate links that are used for the service function chain, \( \text{lev}^V_i \) is the security level that is requested by the service chain, and \( \text{cap}^V_i \) denotes CPU capacity of service function chain.

The security level of a service function chain is given by the sum of security levels of security VNFs that are used in the service function chain. In Fig. 1, the service function chain \( G_i^V \) is constructed by using security VNFs \( f_1, f_2, \) and \( f_i \) in substrate nodes \( C_i^S, D_i^S, \) and \( E_i^S \), respectively. Therefore, the security level of \( G_i^V \), which is four, is equal to the sum of security levels of \( f_1, f_2, \) and \( f_i \).

In the model mentioned above, the cost of constructing a service function chain is defined as the amount of resources that are needed for constructing the service function chain. Now, we define \( F_i^V \) as a set of security VNFs that are used for \( G_i^V \). Here, we consider mapping \( M_i \) between the \( i \)th request \( G_i^V \) of service function chain and the substrate network as

\[
M_i = \{(F_i^V, N^S), (L_i^V, L^S)\}.
\]

By \( M_i \), security VNF \( f_j \) in \( F_i^V \) is selected and it is placed in \( n^S \in N^S \) and virtual link \( l^V \in L_i^V \) is embedded in substrate link \( l^S \in L^S \). The cost of service function chain construction is calculated with the following equation:

\[
\begin{align*}
\text{Cost}(M_i) & = \text{dur}^V_i \\
& \cdot \left[\text{cap}^V_i \cdot \text{lev}^V_i + \sum_{(F_i^V, N^S), (L_i^V, L^S) \in M_i} \sigma_j(n^S) \cdot \zeta_j(n^S)\right].
\end{align*}
\]

In (1), the cost for service function chain \( G_i^V \) is proportional to the utilization time \( \text{dur}^V_i \), and the cost becomes larger for a service function chain that needs a higher CPU capacity \( \text{cap}^V_i \) and a higher security level \( \text{lev}^V_i \). Moreover, \( \sigma_j(n^S) \) and \( \zeta_j(n^S) \) are the amount of CPU capacity and the security level of security VNF \( f_j \) that is placed in \( n^S \). The last part is the total cost of adding a new VNF \( f_j \) into a substrate node \( n^S \). Therefore, the addition of many VNFs increases the cost. On the other hand, the revenue is defined as the amount of resources that are utilized in service function chains, and it is denoted as

\[
\text{Rev}(M_i) = \alpha \cdot \text{dur}^V_i \cdot \text{lev}^V_i \cdot \text{cap}^V_i \cdot \text{len}^V(L_i^V).
\]

In (2), \( \text{Rev}(M_i) \) for a service function chain is large when the utilization time \( \text{dur}^V_i \), the security level \( \text{lev}^V_i \), and the CPU capacity \( \text{cap}^V_i \) for service function chain \( G_i^V \) are large. Moreover, \( \text{len}^V(L_i^V) \) represents the number of substrate links that are used in the \( i \)th service function chain between \( \text{src}^V_i \) and \( \text{dst}^V_i \). Setting parameter \( \alpha \) is decided by the provider so that he can obtain a benefit. The addition of VNFs increases the cost in (1), and hence the provider has to determine \( \alpha \) carefully. Moreover, in this model, the cost is decreased, but the reward is increased by sharing VNFs among multiple service function chains.

In this model, we assume that a network provider determines the security level of each security VNF and the security level that is needed in each network service. On the other hand, each user requests a service function chain for a specific network service and selects only a simple security quality such as high, middle, and low. Then, the selected simple quality is changed into the actual security level \( \text{lev}^V_i \), which is determined by the network operator.

**IV. OPTIMAL CONSTRUCTION OF SERVICE FUNCTION CHAINS BASED ON SECURITY LEVEL**

In this section, we propose an optimal construction of service function chains based on the security level for improving network security. Our proposed method is used to construct service function chains to decrease cost (1) and increase revenue (2) while satisfying the requirement of network security.

Now, let \(|M|\) denote the number of requests. In our proposed method, a lot of service function chains are expected to be constructed by using security VNFs so that revenue (2) minus cost (1) can be maximized. To achieve this end, we formulate the following optimization problem:

\[
\max_{M_i} \sum_{i=1}^{M} [\text{Rev}(M_i) - \text{Cost}(M_i)],
\]

subject to:
insists of eight rows that \( N \) apacity for \( n^s \) in Fi. Moreover, constraint (6) satisfies that the security level that is requested for a service function chain is equal to or smaller than the total security level of security VNFs that are used in the service function chain.

In this paper, we solve this optimization problem with a genetic algorithm. In our genetic algorithm, a two-dimensional chromosome is utilized to represent a combination of used security VNFs and placement of security VNFs. Figure 2 shows a chromosome for our optimization problem in Figure 1. Here, for each chromosome, the number of rows is equal to the number of nodes in the substrate network and the number of columns is given by the maximum number of security VNFs that can be placed in a node, which is derived from \( \min n^s \sigma_j \). In Figure 2, the chromosome consists of eight rows that represent all substrate nodes and three columns that represent three VNFs that can be placed in each node.

Moreover, the pseudocode for this algorithm is shown in Figure 3. This pseudocode is almost the same as the general genetic algorithm. Note that some software such as CPLEX and other meta-heuristic algorithms can be used for solving the above optimization problem.

\[
\sum_{i=1}^{M} \sum_{f \in F^i_M} \sigma_f \leq \text{cpu}(n^s), \quad \forall n^s \in N^S, \quad (4)
\]

\[
\sum_{i=1}^{M} \sum_{f \in F^i_M} a_p^i \leq \text{cpu}(n^s), \quad \forall n^s \in N^S, \quad (5)
\]

\[
\sum_{n \in N^S} \sum_{f \in F^i_M} \zeta_f(n^s) \geq \text{lev}_f^v, \quad \forall G^v_f. \quad (6)
\]

Constraint (4) ensures that the amount of CPU resources in \( n^s \) is sufficient to place security VNF \( f_j \). Constraint (5) ensures that security VNF in substrate nodes have enough processing capacity for each service function chain. Moreover, constraint (6) satisfies that the security level that is requested for a service function chain is equal to or smaller than the total security level of security VNFs that are used in the service function chain.

Here, the genetic algorithm solves our optimization problem by generating a group of random chromosomes and evaluate each chromosome by equation (3)-(6). Two best chromosomes whose (3) is higher are selected to create another group of random chromosomes by mutation and recombination. This procedure continues until the obtained result reaches convergence or until the number of generations reaches the maximum number of generations.

Figure 2. Chromosome for our problem in Figure 1.

Here, the genetic algorithm solves our optimization problem by generating a group of random chromosomes and evaluate each chromosome by equation (3)-(6). Two best chromosomes whose (3) is higher are selected to create another group of random chromosomes by mutation and recombination. This procedure continues until the obtained result reaches convergence or until the number of generations reaches the maximum number of generations.

Figure 3. Pseudocode of genetic algorithm for our optimization problem.
Figure 4. Example of construction of two service function chains with our proposed method.

V. PERFORMANCE EVALUATION

In this section, we evaluate the performance of our proposed method in substrate networks shown in Figure 1 and Figure 5. For both topologies, the amount $cpu(n^4)$ of CPU for $n^4$ is set at random among [160, 190]. Here, three types of VNF can be used in each node, i.e., $F(n^4) = 3$ for all nodes, and those are $f_1 (\sigma_1 = 25, \zeta_1 = 1)$, $f_2 (\sigma_2 = 40, \zeta_2 = 2)$, and $f_3 (\sigma_3 = 50, \zeta_3 = 3)$.

For these networks, user’s request $G^i_f$ for a service function chain arrive at a substrate network according to a Poisson process with rate $\lambda$. The utilization time $dur^i_f$ of service function chains follows an exponential distribution with rate 10. Just after the utilization of the service function chain finishes, the used resources are returned to each substrate network. Moreover, in each request, the source node $src^i_f$ and destination node $dst^i_f$ are selected randomly. A set $L^i_f$ of links in the transmission route has been determined randomly in advance. Security level $lev^i_f$ for $G^i_f$ is selected at random among [1, 6], and the amount $cap^i_f$ of CPU capacity is decided randomly among [40, 80].

For each request, the optimization problem (3)-(6) is solved by using a genetic algorithm. Note that the selection of security VNFs and the placement of selected VNFs are determined from the optimization problem. The genetic algorithm utilizes 100 two-dimensional chromosomes in each generation, and the maximum number of generations is 5,000.

For the performance comparison, we evaluate the performance of our method in two cases: one is a case where security VNFs can be shared among multiple service function chains (Shared method) and the other is a case where security VNFs cannot be shared among multiple service function chains (NonShared method). In the NonShared method, a service function chain exclusively uses all of the capacity of the existing VNFs.

Moreover, we evaluate the performance of two methods where VNFs are selected and placed according to ascending order or descending order of $\sigma$. These methods are called Descending method and Ascending method, respectively. In the two methods, VNFs are placed in a node with larger amount $cpu(n^4)$ of CPU capacity. That is, a VNF whose CPU capacity is the $i$th largest (smallest) is placed in a node whose amount of CPU capacity is the $i$th largest. Therefore, the ascending and descending methods do not use a genetic algorithm. Note that we have evaluated the performances of methods where a node with a smaller amount of CPU capacity is selected, however, this method is not more effective than both descending method and ascending method.

For each method, we evaluate the average value of Rev-Cost, the rejection rate, and the average level of security. Here, $|M|$ denotes the number of requests. In this case, the average value of Rev-Cost is calculated based on (3) as follows.

$$\text{Rev} - \text{Cost} = \frac{\sum_{i=1}^{M} (\text{Rev}(M_i) - \text{Cost}(M_i))}{\sum_{i=1}^{M} \text{Accept}(M_i)}. \quad (7)$$

In this equation, $\text{Accept}(M_i)$ is one if the request $M_i$ is accepted and zero if $M_i$ is rejected. Therefore, $\sum_{i=1}^{M} \text{Accept}(M_i)$ is the number of accepted requests. Moreover, the rejection rate is given by the number of rejected requests divided by the total number of requests as follows:

$$\text{Reject rate} = \frac{|M| - \sum_{i=1}^{M} \text{Accept}(M_i)}{|M|}. \quad (8)$$

Finally, the average value of the security level for the constructed service function chains is derived from

$$\text{Avg Lev} = \frac{\sum_{i=1}^{M} \text{Lev}(\text{Accept}(M_i))}{\sum_{i=1}^{M} \text{Accept}(M_i)}. \quad (9)$$

A. Impact of Arrival Rate

In this section, we investigate the impact of the arrival rate
of user’s requests on the performance of each method in Subs8 Topology shown in Figure 1.

Figure 6 shows how Rev-Cost changes against arrival rate $\lambda$. From this figure, we find that Rev-Cost for our proposed Shared method is higher than those for other methods regardless of $\lambda$. Here, higher Rev-Cost means that the efficient construction of service function chains is performed in the substrate node. Therefore, from this figure, we find that our proposed Shared method is the most effective for service function chains to improve network security.

Figure 7 shows that the impact of arrival rate $\lambda$ on the rejection rate for each method. This figure shows that the rejection rate for the proposed shared method is smaller than those for other methods regardless of $\lambda$. Therefore, our proposed method can construct many service function chains by using a limited amount of resources effectively.

Moreover, Figure 8 shows the average value of the security level for constructed service function chains. From this figure, we find that the descending method has the highest security level against $\lambda$. Note that each method utilizes three types of security VNFs $f_1$ ($\sigma_1 = 25, \zeta_1 = 1$), $f_2$ ($\sigma_2 = 40, \zeta_2 = 2$), and $f_3$ ($\sigma_3 = 50, \zeta_3 = 3$), and security VNF with higher security level needs a larger amount of CPU resources. Because many security VNFs whose security level is higher are used in the descending method, the average security level becomes the highest for the descending method. On the other hand, the average security level of our proposed Shared method is small because CPU resources are utilized effectively for maximizing the objective function (3).

**B. Impact of Security Level**

In this section, we investigate the impact of security level of service function chain on the performance of each method in Subs8 Topology. In the following, the security level of service function chain is selected among $[1, 6]$. Moreover, the arrival rate is set to $\lambda=40$ or $\lambda=180$. In this section, we evaluate only the performance of our proposed Shared method and NonShared method. This is because Descending and Ascending methods are not effective as shown in the previous subsection.
number of VNFs for NonShared method in $\lambda=180$ is the smallest. Figure 12 shows the average cost for constructing service function chains against the security level of the service function chain. From this figure, we find that the average cost increases as the security level of service function chain increases. Moreover, the average cost for the Shared method is smaller than the NonShared method regardless of security level.

From these results, we find that our proposed Shared method is effective for constructing service function chains regardless of the requested security level.

Figure 9 shows Rev-Cost against security level for each service function chain. From this figure, we find that Rev-Cost for the proposed Shard method is somewhat higher than that for the NonShared method regardless of the security level and $\lambda$. Therefore, from this figure, we find that our proposed Shared method is the most efficient for the construction of service function chains.

Figure 10 shows that the rejection rate of the proposed Shared method is somewhat lower than that of NonShared method regardless of security level in the case of $\lambda=40$. On the other hand, in the case of $\lambda=180$, the rejection rate for NonShared method is much higher than that for the Shared method. This is because a higher amount of CPU resources is utilized in the NonShared method and it is hard to place new security VNFs.

Figure 11 shows the average number of used security VNFs against security level in Subs topology. From this figure, we find that higher requested security level needs a larger number of security VNFs. On the other hand, the average number of VNFs for NonShared method in $\lambda=180$ is the smallest. Figure 12 shows the average cost for constructing service function chains against the security level of the service function chain. From this figure, we find that the average cost increases as the security level of service function chain increases. Moreover, the average cost for the Shared method is smaller than the NonShared method regardless of security level.

From these results, we find that our proposed Shared method is effective for constructing service function chains regardless of the requested security level.

**C. Impact of Network Topology**

In this section, we evaluate the performance of each method in NSFnet topology shown in Figure 5. NSFnet topology is an actual network topology and it has been used in many papers. Therefore, we also evaluate the performance of our method in NSFnet.
Figure 13 shows Rev-Cost against arrival rate $\lambda$. This figure shows that Rev-Cost for the proposed Shared method is higher than those of other methods regardless of $\lambda$, even in NSFnet topology. This result is similar to the result of Subs8 topology.

Figure 14 shows that the rejection rate against arrival rate $\lambda$ in NSFnet. From this figure, we find that the rejection rate of our proposed Shared method is the smallest among all methods regardless of $\lambda$. Here, the rejection rate is decreased by placing security VNFs effectively. Moreover, Figure 15 shows the average value of the security level for service function chains constructed in NSFnet. Here, the Descending method has a higher security level. This result is also similar to the result for Subs8 topology as shown in Figure 8. On the other hand, the average security level for our proposed Shared method is small.

From these results, we find that our proposed Shared method is the most effective for service function chain construction regardless of network topology.

D. Calculation Time
Finally, for both topologies, we evaluate the calculation time for solving the optimization problem for one user’s request. For both topologies, the security level in the user’s request is set to 1, 2, 3, 4, 5, or 6. We derive the calculation time that is taken from the arrival of a request to solve the optimization problem.

Table 1 shows that the calculation time for solving the optimization problem increases as the security level increases. This is because many security VNFs are used when the security level is large. Moreover, the calculation time for NSFnet topology is larger than that for Subs8 topology. This means that the calculation time becomes large when the number of nodes in the substrate network is large. However, even if the security level and the number of nodes in the topology increase, our proposed method can calculate the optimization problem and construct a service function chain within 1.0 min in this case.

VI. CONCLUSIONS
In this paper, we proposed the optimal construction for
service function chaining based on the security level for improving network security. In this method, some security VNFs are placed in order to satisfy the security level of the service function chaining and a lot of service function chaining can be constructed in a substrate network. We evaluated the performance of our proposed method. From numerical examples, we found that a larger number of service function chains can be constructed based on the security level by using the proposed method. Moreover, by using the proposed method service function chains can be constructed with high revenue and less cost.

ACKNOWLEDGMENT
The authors would like to thank the University of Fukui, National Nuclear Energy Agency of Indonesia (BATAN) and Research and Innovation in Science and Technology Project (RISET-PRO), Ministry of Research, Technology, and Higher Education of Indonesia as a supporter for this research.

REFERENCES
Dhanu Dwiardhika received the B. S. degree in physics science from the Universitas Gadjah Mada, Indonesia, in 2006, and M. S. degree in computer science from the Universitas Indonesia, Indonesia, in 2014.

From 2008 to 2011 he was a network administrator at National Nuclear Energy Agency of Indonesia. From 2014 to 2017 he was a data center engineer at National Nuclear Energy Agency of Indonesia.

Since 2017, he has been a doctor course student at Graduate School of Engineering, University of Fukui, Japan.

Takuji Tachibana received the B. Eng. degree from the Department of Systems Engineering, Nagoya Institute of Technology, Japan, in 2000. He received the M. Eng. and Dr. Eng. degrees from the Department of Information Systems, Graduate School of Information Science, Nara Institute of Science and Technology, Japan, in 2001 and 2004, respectively.

From 2004 to 2006, he was an expert researcher in the Information and Network Systems Department, National Institute of Information and Communications Technology, Japan. From 2006 to 2011, he was an assistant professor at the Department of Information Systems, Graduate School of Information Science, Nara Institute of Science and Technology, Japan. From 2011 to 2018, he was an associate professor at Graduate School of Engineering, University of Fukui, Japan. Since 2019, he has been a professor at Graduate School of Engineering, University of Fukui, Japan.

His research interests include network architectures in optical networking and performance analysis of computer and communication systems. He is a member of the IEEE, The Institute of Electronics, Information and Communication Engineers (IEICE), and the Operations Research Society of Japan.