IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received September 5, 2019, accepted September 16, 2019, date of publication October 2, 2019, date of current version October 17, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2944982

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

DHANU DWIARDHIKA[1,2] **AND TAKUJI TACHIBANA**[1], **(Member, IEEE)**

[1]Graduate School of Engineering, University of Fukui, Fukui 910-8507, Japan
[2]Center for Informatics and Nuclear Strategic Zone Utilization, National Nuclear Energy Agency of Indonesia, Jakarta 12710, Indonesia

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

This work was supported by 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.

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]–[4]. For example, multiple virtual machines can be implemented in a physical server [5]–[7], and network function such as load-balancing, router, and firewalls can be implemented as a virtual machine in the physical server [8]–[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]–[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]–[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]–[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

The associate editor coordinating the review of this manuscript and approving it for publication was Jiafeng Xie.

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]–[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]. Reference [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. Reference [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.

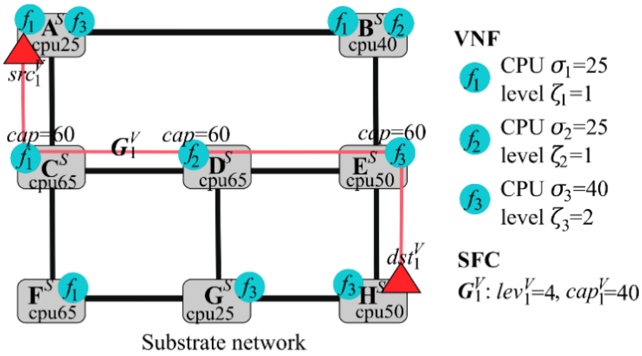


FIGURE 1. System model.

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.

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 $G^S = (N^S, L^S, A_N^S, A_L^S)$. 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:

$$A_N^S = \left\{\left\{cpu\left(n^s\right), F\left(n^s\right)\right\} \mid n^s \in N^S\right\},$$
$$A_L^S = \left\{\left\{bw\left(l^s\right)\right\} \mid l^s \in L^S\right\}.$$

The jth security VNF is denoted as f_j, and the performance of f_j is denoted as

$$f_j = \left\{\sigma_j, \zeta_j\right\}.$$

Here, σ_j is the amount of CPU capacity that is needed for utilizing f_j and ζ_j is the security level of f_j. When f_j is placed in n^s, the amount of available CPU in n^s decreases by σ_j. Moreover, when ζ_j for security VNF f_j is larger than ζ_k for security

VNF f_k, σ_j is larger than σ_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 ith request of service function chain is denoted as

$$G_i^V = \left\{dur_i^V, src_i^V, dst_i^V, L_i^V, lev_i^V, cap_i^V\right\},$$

where dur_i^V is the utilization time of this service function chain, src_i^V and dst_i^V 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, lev_i^V is the security level that is requested by the service chain, and cap_i^V 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_3 in substrate nodes C^S, D^S, and E^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_3.

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 ith 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:

$$Cost\,(M_i) = dur_i^V \cdot \left[cap_i^V \cdot lev_i^V + \sum_{\{(F_i^V, N^S), (L_i^V, L^S)\} \in \boldsymbol{M}_i} \sigma_j\left(n^s\right) \cdot \zeta_j\left(n^s\right)\right]. \tag{1}$$

In (1), the cost for service function chain G_i^V is proportional to the utilization time dur_i^V, and the cost becomes larger for a service function chain that needs a higher CPU capacity cap_i^V and a higher security level lev_i^V. 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

$$Rev\,(M_i) = \alpha \cdot dur_i^V \cdot lev_i^V \cdot cap_i^V \cdot len_i^V\left(L_i^V\right). \tag{2}$$

In (2), $Rev\,(M_i)$ for a service function chain is large when the utilization time dur_i^V, the security level lev_i^V, and the CPU capacity cap_i^V for service function chain G_i^V are large. Moreover, $len_i^v\left(L_i^V\right)$ represents the number of substrate links that are used in the ith service function chain between src_i^V and dst_i^V. Setting parameter α 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 α 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 lev_i^V, 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:

$$\max_{\boldsymbol{M}_i} \sum_{i=1}^{|M|} \left[Rev\,(M_i) - Cost\,(M_i)\right], (3) \tag{3}$$

subject to:

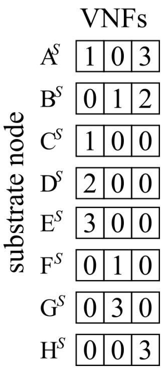
$$\sum_{i=1}^{|M|} \sum_{f_j \in F_i^V, \boldsymbol{M}_i} \sigma_j \leq cpu\left(n^s\right), \quad \forall n^s \in N^S, \tag{4}$$

$$\sum_{i=1}^{|M|} \sum_{\mathbf{Mi}} cap_i^V \leq cpu\left(n^s\right), \quad \forall n^s \in N^S, \tag{5}$$

$$\sum_{n^s \in N^S} \sum_{f_j \in F_i^V, \boldsymbol{M}_i} \zeta_j\left(n^s\right) \geq lev_i^V, \quad \forall G_i^V. \tag{6}$$

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.

In this paper, we solve this optimization problem with a genetic algorithm. In our genetic algorithm,



substrate node	VNFs		
A^S	1	0	3
B^S	0	1	2
C^S	1	0	0
D^S	2	0	0
E^S	3	0	0
F^S	0	1	0
G^S	0	3	0
H^S	0	0	3

FIGURE 2. Chromosome for our problem in Figure 1.

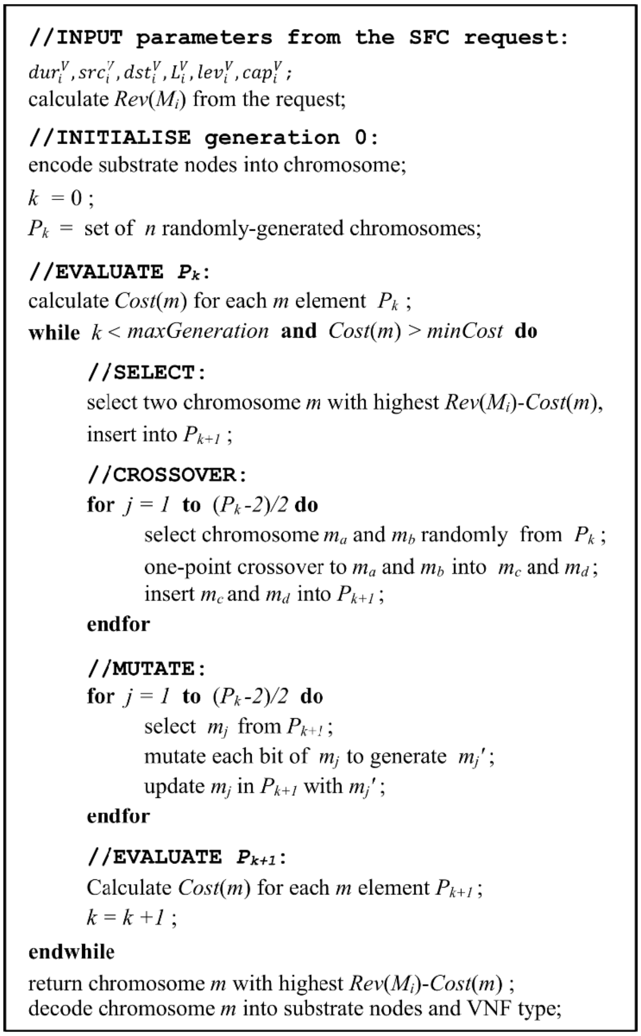
a two-dimensional chromosome is utilized to represent a combination of used security VNFs and placement of security VNFs. Figure 2 shows a chromosome for our optimization problem in Figure 1. Here, for each chromosome, the number of rows is equal to the number of nodes in the substrate network and the number of columns is given by the maximum number of security VNFs that can be placed in a node, which is derived from $\max_{n^s} cpu\,(n^s)\,/\sum \sigma_j$. 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.

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. 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.

Figure 4 shows how two service function chains are constructed. In this figure, G_1^V requires that the security level is equal to or larger than four and G_2^V requires that the security level is equal to or larger than two. Therefore, G_1^V utilizes f_1(security level is 1), f_2 (security level is 1), and f_3(security level is 2), and G_2^V utilizes f_1 and f_2 so as to satisfy the security level. Moreover, in order to minimize (3), VNFs f_1 and f_2 are shared by G_1^V and G_2^V.

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 $cpu\,(n^s)$ of CPU for n^s is set at random among [160, 190]. Here, three types of VNF can be used in each node, i.e., $F\,(n^s) = 3$ for all nodes, and those are $f_1(\sigma_1 = 25, \zeta_1 = 1)$, $f_2\,(\sigma_2 = 40, \zeta_2 = 2)$, and $f_3\,(\sigma_3 = 50, \zeta_3 = 3)$.



```
//INPUT parameters from the SFC request:
$dur_i^V, src_i^V, dst_i^V, L_i^V, lev_i^V, cap_i^V$;
calculate $Rev(M_i)$ from the request;
//INITIALISE generation 0:
encode substrate nodes into chromosome;
$k = 0$;
$P_k$ = set of $n$ randomly-generated chromosomes;
//EVALUATE $P_k$:
calculate $Cost(m)$ for each $m$ element $P_k$;
while $k < maxGeneration$ and $Cost(m) > minCost$ do
    //SELECT:
    select two chromosome $m$ with highest $Rev(M_i)$-$Cost(m)$,
    insert into $P_{k+1}$;
    //CROSSOVER:
    for $j = 1$ to $(P_k$-$2)/2$ do
        select chromosome $m_a$ and $m_b$ randomly from $P_k$;
        one-point crossover to $m_a$ and $m_b$ into $m_c$ and $m_d$;
        insert $m_c$ and $m_d$ into $P_{k+1}$;
    endfor
    //MUTATE:
    for $j = 1$ to $(P_k$-$2)/2$ do
        select $m_j$ from $P_{k+1}$;
        mutate each bit of $m_j$ to generate $m_j'$;
        update $m_j$ in $P_{k+1}$ with $m_j'$;
    endfor
    //EVALUATE $P_{k+1}$:
    Calculate $Cost(m)$ for each $m$ element $P_{k+1}$;
    $k = k + 1$;
endwhile
return chromosome $m$ with highest $Rev(M_i)$-$Cost(m)$;
decode chromosome $m$ into substrate nodes and VNF type;
```

FIGURE 3. Pseudocode of genetic algorithm for our optimization problem.

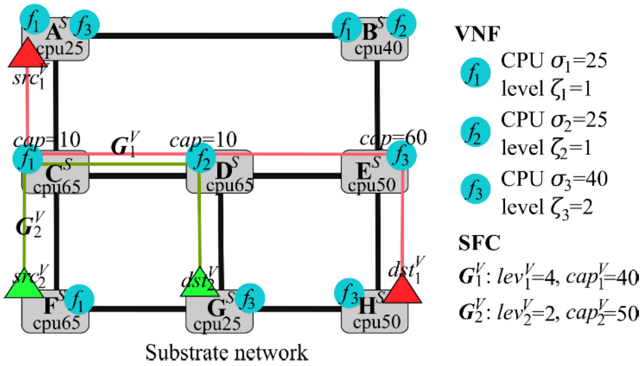


FIGURE 4. Example of construction of two service function chains with our proposed method.

For these networks, user's request G_i^V for a service function chain arrive at a substrate network according to a Poisson process with rate λ. The utilization time dur_i^V of service function chain 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

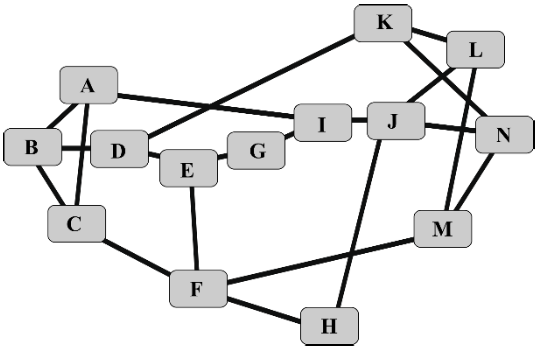


FIGURE 5. NSFnet topology for performance evaluation of our proposed method.

network. Moreover, in each request, the source node src_i^V and destination node dst_i^V are selected randomly. A set L_i^V of links in the transmission route has been determined randomly in advance. Security level lev_i^V for G_i^V is selected at random among [1], [6], and the amount cap_i^V 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 $\sigma_{j.}$ These methods are called *Descending method* and *Ascending method*, respectively. In the two methods, VNFs are placed in a node with larger amount $cpu\,(n^s)$ of CPU capacity. That is, a VNF whose CPU capacity is the ith largest (smallest) is placed in a node whose amount of CPU capacity is the ith 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:

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

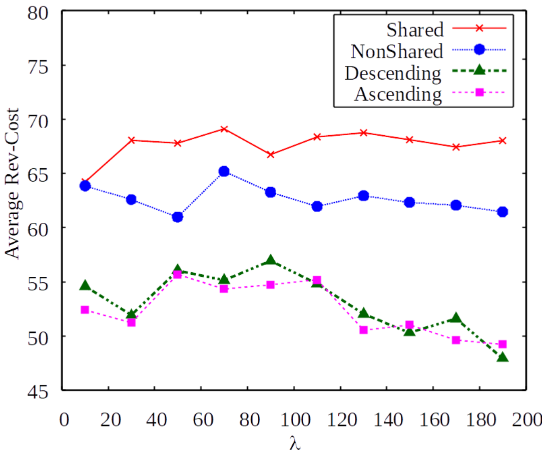


FIGURE 6. Revenue - Cost against arrival rate in Subs8 topology.

In this equation, $Accept\,(M_i)$ is one if the request M_i is accepted and zero if M_i is rejected. Therefore, $\sum_{i=1}^{|M|} Accept\,(M_i)$ 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:

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

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

$$Avg\ Lev = \frac{\sum_{i=1}^{|M|} Lev(Accept(M_i))}{\sum_{i-1}^{|M|} Accept(M_i)}. \tag{9}$$

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 λ. From this figure, we find that Rev-Cost for our proposed Shared method is higher than those for other methods regardless of λ. 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 λ 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 λ. 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 λ. 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

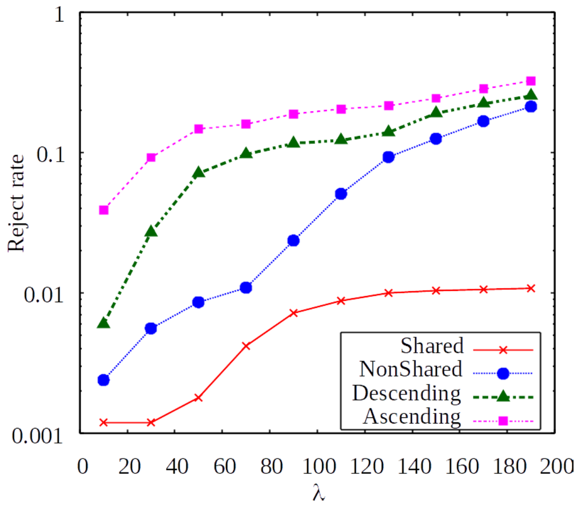


FIGURE 7. Rejection rate against arrival rate in Subs8 topology.

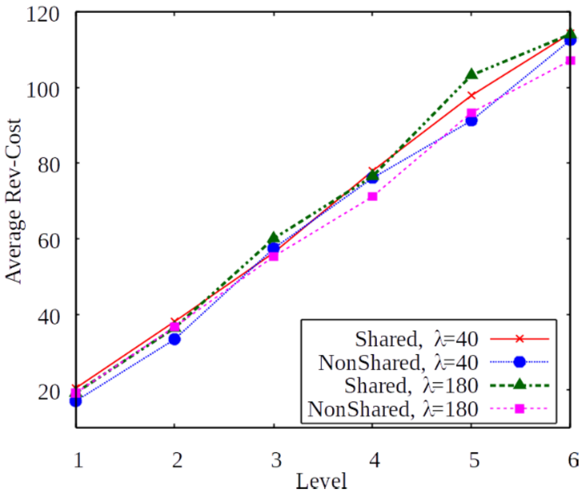


FIGURE 9. Revenue - Cost against security level in Subs8 topology.

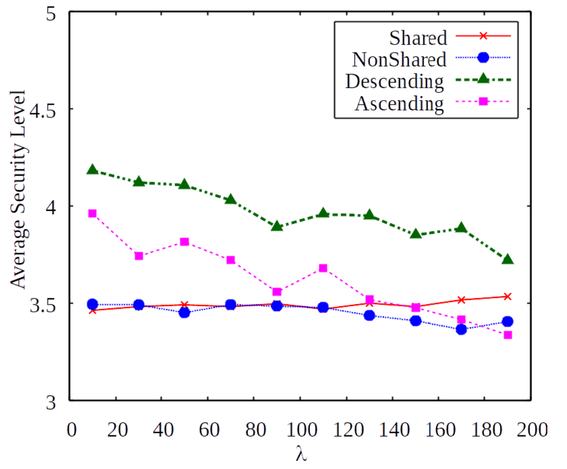


FIGURE 8. Average Security Level against arrival rate in Subs8 topology.

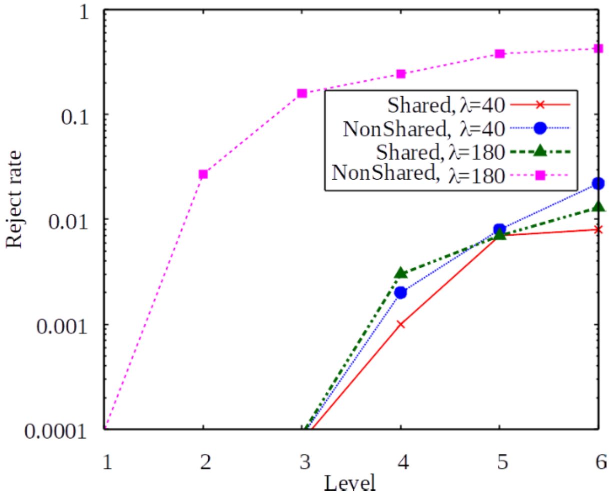


FIGURE 10. Rejection rate against security level in Subs8 topology.

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.

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 λ. 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 VNFs against security level of service function chain. 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 ser-

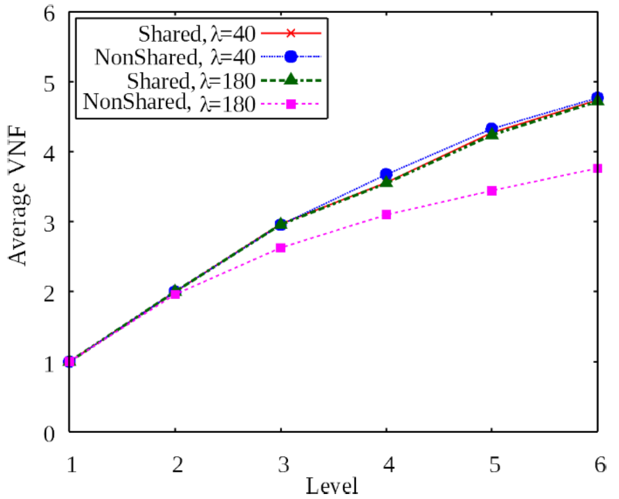


FIGURE 11. Average number of used security VNFs against security level in Subs8 topology.

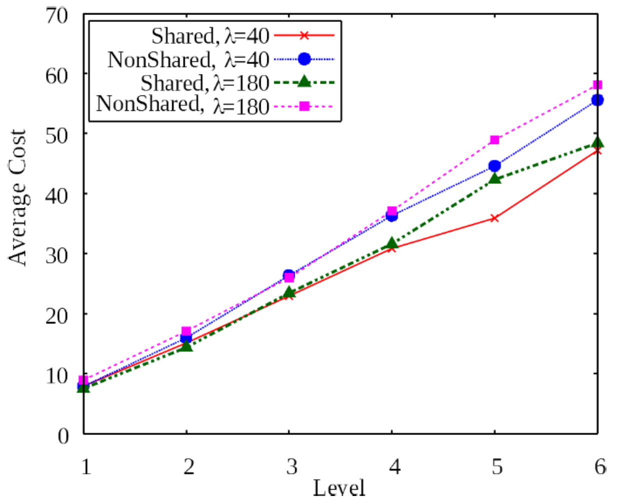


FIGURE 12. Average Cost against security level in Subs8 topology.

vice 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 λ. This figure shows that Rev-Cost for the proposed Shared method is higher than those of other methods regardless of λ even in

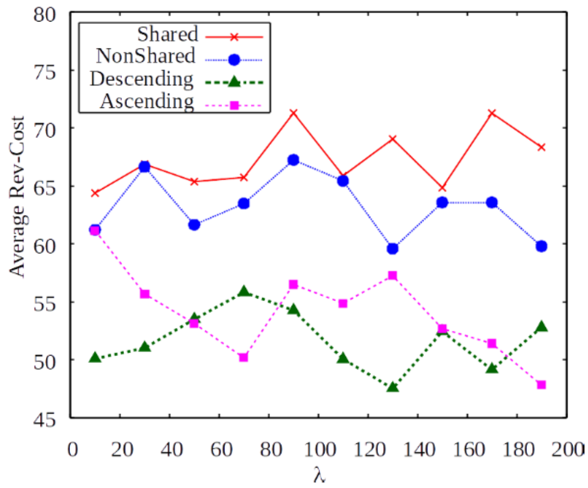


FIGURE 13. Rev-Cost against arrival rate in NSFNet topology.

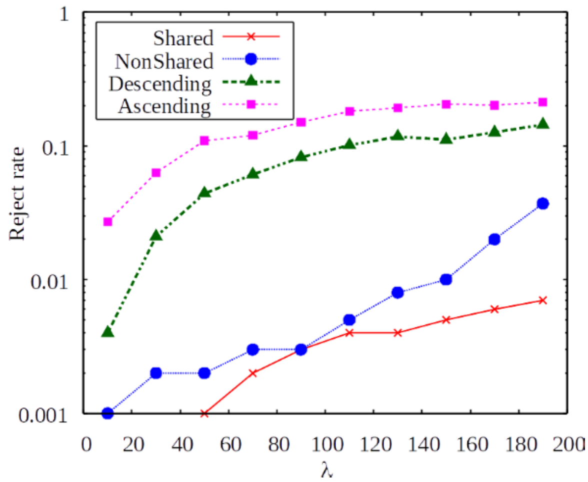


FIGURE 14. Rejection rate against arrival rate in NSFnet topology.

NSFnet topology. This result is similar to the result of Subs8 topology.

Figure 14 shows that the rejection rate against arrival rate λ in NSFnet. From this figure, we find that the rejection rate of our proposed Shared method is the smallest among all methods regardless of λ. 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.

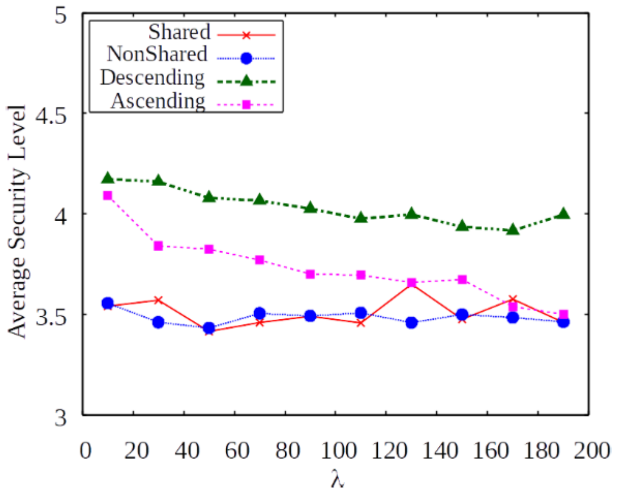


FIGURE 15. Average Security Level against arrival rate in NSFnet topology.

TABLE 1. Calculation time for subs8 topology and NSFnet topology.

Security Level	Subs8 Topology	NSFnet Topology
1	11.6 [sec]	15.2 [sec]
2	12.0 [sec]	15.9 [sec]
3	12.5 [sec]	16.5 [sec]
4	12.7 [sec]	17.0 [sec]
5	12.8 [sec]	17.5 [sec]
6	13.4 [sec]	17.8 [sec]

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. CONCLUSION

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.

REFERENCES

[1] S.-H. Li, D. C. Yen, S.-C. Chen, P. S. Chen, W.-H. Lu, and C.-C. Cho, "Effects of virtualization on information security," *Comput. Standards Interfaces*, vol. 42, pp. 1–8, Nov. 2015.

[2] S. Clayman, E. Maini, A. Galis, A. Manzalini, and N. Mazzocca, "The dynamic placement of virtual network functions," in *Proc. IEEE Netw. Oper. Manage. Symp. (NOMS)*, Poland, May 2014, pp. 1–9.

[3] C. Xing, J. Lan, and Y. Hu, "Virtual network with security guarantee embedding algorithms," *J. Comput.*, vol. 8, no. 11, pp. 2782–2788, Nov. 2013.

[4] A. Gupta, M. F. Habib, U. Mandal, P. Chowdhury, M. Tornatore, and B. Mukherjee, "On service-chaining strategies using virtual network functions in operator networks," *Comput. Netw.*, vol. 133, pp. 1–16, Mar. 2018.

[5] S. Liu, Z. Cai, H. Xu, and M. Xu, "Security-aware virtual network embedding," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Sydney, NSW, Australia, Jun. 2014, pp. 834–840.

[6] C. Beşiktaş, D. Gözüpek, A. Ulaş, and E. Lokman, "Secure virtual network embedding with flexible bandwidth-based revenue maximization," *Comput. Netw.*, vol. 121, pp. 89–99, Jul. 2017.

[7] B. Addis, D. Belabed, M. Bouet, and S. Secci, "Virtual network functions placement and routing optimization," in *Proc. IEEE 4th Int. Conf. Cloud Netw. (CloudNet)*, Niagara Falls, ON, Canada, 2015, pp. 171–177.

[8] H. Chen, X. Wang, Y. Zhao, T. Song, Y. Wang, S. Xu, and L. Li, "Mosc: A method to assign the outsourcing of service function chain across multiple clouds," *Comput. Netw.*, vol. 133, pp. 166–182, Mar. 2018.

[9] G. Cheng, H. Chen, H. Hu, Z. Wang, and J. Lan, "Enabling network function combination via service chain instantiation," *Comput. Netw.*, vol. 92, pp. 396–407, Dec. 2015.

[10] B. Yi, X. Wang, S. K. Das, K. Li, and M. Huang, "A comprehensive survey of network function virtualization," *Comput. Netw.*, vol. 133, pp. 212–262, Mar. 2018.

[11] B. Addis, M. Gao, and G. Carello, "On the complexity of a virtual network function placement and routing problem," *Electron. Notes Discrete Math.*, vol. 69, pp. 197–204, Aug. 2018.

[12] T. Li, H. Zhou, and H. Luo, "A new method for providing network services: Service function chain," *Opt. Switching Netw.*, vol. 26, pp. 60–68, Nov. 2017.

[13] H. Jang, J. Jeong, H. Kim, and J. Park, "A survey on interfaces to network security functions in network virtualization," in *Proc. IEEE 29th Int. Conf. Adv. Inf. Netw. Appl. Workshops*, Seoul South Korea, Mar. 2015, pp. 160–163.

[14] J. Guan, Z. Wei, and I. You, "GRBC-based Network Security Functions placement scheme in SDS for 5G security," *J. Netw. Comput. Appl.*, vol. 114, pp. 48–56, Jul. 2018.

[15] M. D. Firoozjaei, J. Jeong, H. Ko, and H. Kim, "Security challenges with network functions virtualization," *Future Gener. Comput. Syst.*, vol. 67, pp. 315–324, Feb. 2017.

[16] A. Aljuhani and T. Alharbi, "Virtualized network functions security attacks and vulnerabilities," in *Proc. IEEE 7th Annu. Comput. Commun. Workshop Conf. (CCWC)*, Las Vegas, NV, USA, Jan. 2017, pp. 1–4.

[17] A. M. Alwakeel, A. K. Alnaim, and E. B. Fernandez, "A survey of network function virtualization security," in *Proc. SoutheastCon*, St. Petersburg, FL, USA, Apr. 2018, pp. 1–8.

[18] M. Ehrlich, L. Wisniewski, H. Trsek, D. Mahrenholz, and J. Jasperneite, "Automatic mapping of cyber security requirements to support network slicing in software-defined networks," in *Proc. 22nd IEEE Int. Conf. Emerg. Technol. Factory Automat. (ETFA)*, Cyprus, Sep. 2017, pp. 1–4.

[19] A. F. Ocampo, J. Gil-Herrera, P. H. Isolani, M. C. Neves, J. F. Botero, S. Latré, L. Zambenedetti, M. P. Barcellos, and L. P. Gaspary, "Optimal service function chain composition in network functions virtualization," in *Proc. AIMS*, Switzerland, 2017, p. 62.

[20] S. Shin, H. Wang, and G. Gu, "A first step toward network security virtualization: From concept to prototype," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 10, pp. 2236–2249, Oct. 2015.

[21] M. Sourour, B. Adel, and A. Tarek, "Collaboration between security devices toward improving network defense," in *Proc. 7th IEEE/ACIS Int. Conf. Comput. Inf. Sci. (IcIs)*, Portland, OR, USA, May 2008, pp. 13–18.

[22] D. Migault, M. A. Simplicio, Jr., B. M. Barros, M. Pourzandi, T. R. Almeid, E. R. Andrade, and T. C. Carvalho, "A framework for enabling security services collaboration across multiple domains," *Comput. Elect. Eng.*, vol. 69, pp. 224–239, Jul. 2018.

[23] T.-W. Kuo, B.-H. Liou, K. C.-J. Lin, and M.-J. Tsai, "Deploying chains of virtual network functions: On the relation between link and server usage," in *Proc. IEEE INFOCOM 35th Annu. IEEE Int. Conf. Comput. Commun.*, San Francisco, CA, USA, Apr. 2016, pp. 1–9.

[24] M. Gao, B. Addis, M. Bouet, and S. Secci, "Optimal orchestration of virtual network functions," *Comput. Netw.*, vol. 142, pp. 108–127, Sep. 2018.

[25] D. Qi, S. Shen, and G. Wang, "Towards an efficient VNF placement in network function virtualization," *Comput. Commun.*, vol. 138, pp. 81–89, Apr. 2019.

[26] D. Dwiardhika and T. Tachibana, "Virtual network embedding based on security level with VNF placement," *Secur. Commun. Netw.*, vol. 2019, Feb. 2019, Art. no. 5640134.

[27] D. Amaya, Y. Sumi, S. Homma, T. Okugawa, and T. Tachibana, "VNF placement with optimization problem based on data throughput for service chaining," in *Proc. IEEE 7th Int. Conf. Cloud Netw. (CloudNet)*, Tokyo, Japan, Oct. 2018, pp. 1–3.

[28] Z. Xu, W. Liang, A. Galis, Y. Ma, Q. Xia, and W. Xu, "Throughput optimization for admitting NFV-enabled requests in cloud networks," *Comput. Netw.*, vol. 143, pp. 15–29, Oct. 2018.

[29] Y.-W. Ma, J.-L. Chen, and J.-Y. Jhou, "Adaptive service function selection for network function Virtualization networking," *Future Gener. Comput. Syst.*, vol. 91, pp. 108–123, Feb. 2019.

[30] A. N. Toosi, J. Son, Q. Chi, and R. Buyya, "ElasticSFC: Auto-scaling techniques for elastic service function chaining in network functions virtualization-based clouds," *J. Syst. Softw.*, vol. 152, pp. 108–119, Jun. 2019.

[31] G. Sun, Y. Li, Y. Li, D. Liao, and V. Chang, "Low-latency orchestration for workflow-oriented service function chain in edge computing," *Future Gener. Comput. Syst.*, vol. 85, pp. 116–128, Aug. 2018.

[32] J. Son and R. Buyya, "Latency-aware virtualized network function provisioning for distributed edge clouds," *J. Syst. Softw.*, vol. 152, pp. 24–31, Jun. 2019.

DHANU DWIARDHIKA received the B.S. degree in physics science from the Universitas Gadjah Mada, Indonesia, in 2006, and the M.S. degree in computer science from the Universitas Indonesia, Indonesia, in 2014. He is currently pursuing the Ph.D. degree with the Graduate School of Engineering, University of Fukui, Japan.

From 2008 to 2011, he was a network administrator with the National Nuclear Energy Agency of Indonesia, where he has been a Data Center Engineer with the Center for Informatics and Nuclear Strategic Zone Utilization, since 2014.

TAKUJI TACHIBANA received the B.Eng. degree from the Department of Systems Engineering, Nagoya Institute of Technology, Japan, in 2000, and 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 with the Department of Information Systems, Graduate School of Information Science, Nara Institute of Science and Technology. From 2011 to 2018, he was an Associate Professor with the Graduate School of Engineering, University of Fukui, Japan, where he has been a Professor, since 2019. His research interests include network architectures in optical networking, and performance analysis of computer and communication systems.

Dr. T. Tachibana is a member of The Institute of Electronics, Information and Communication Engineers (IEICE), and the Operations Research Society of Japan.