Virtual Network Function Placement and Routing for Multicast Service Chaining using Merged Paths☆

Narumi Kiji[a], Takehiro Sato[a,*], Ryoichi Shinkuma[a], Eiji Oki[a]

[a] *Graduate School of Informatics, Kyoto University, Yoshida-honmachi, Sakyo-ku, Kyoto 606-8501, Japan*

Abstract

This paper proposes a virtual network function placement and routing model for multicast service chaining based on merging multiple service paths (MSC-M). The multicast service chaining (MSC) is used for providing a network-virtualization based multicast service. The MSC sets up a multicast path, which connects a source node and multiple destination nodes. Virtual network functions (VNFs) are placed on the path so that users on the destination nodes receive their desired services. The conventional MSC model configures multicast paths for services, each of which has the same source data and the same set of VNFs in a predefined order. In the MSC-M model, if paths of different services carry the same data on the same link, these paths are allowed to be merged into one path at that link, which improves the utilization of network resources. The MSC-M model determines the placement of VNFs and the route of paths so that the total cost associated with VNF placement and link usage is minimized. The MSC-M model is formulated as an integer linear programming (ILP) problem. We prove that the decision version of VNF placement and routing problem based on the MSC-M model is NP-complete. A heuristic algorithm is introduced for the case that the ILP problem is intractable. Numerical results

☆This work was supported in part by JSPS KAKENHI, Japan, under Grant Numbers 18H03230 and 19K14980.

*Corresponding author. Tel.: +81 757534808.

Email addresses: nkiji_contact@icn.cce.i.kyoto-u.ac.jp (Narumi Kiji), takehiro.sato@i.kyoto-u.ac.jp (Takehiro Sato), shinkuma@i.kyoto-u.ac.jp (Ryoichi Shinkuma), oki@i.kyoto-u.ac.jp (Eiji Oki)

show that the MSC-M model reduces the total cost required to accommodate service chaining requests compared to the conventional MSC model. We discuss directions for extending the MSC-M model to an optical domain.

Keywords: Virtual network function, Service chaining, Multicast

1. Introduction

Recently, a technology called network function virtualization (NFV), which separates a network function from dedicated hardware and implements the function as a software instance operated on general purpose hardware, has been developed [1]. The function implemented as a software instance is called virtual network function (VNF). Network functions such as firewalls, deep packet inspections and network address translations are conventionally implemented by using dedicated hardware, which causes problems in the arrangement and maintenance of network devices and costs. NFV has several major advantages to solve these problems. First, by virtualizing network functions on a relatively inexpensive server compared to dedicated hardware, the equipment cost is reduced. Second, when a network operator introduces new services, time and effort can be saved. In the NFV environment, a network operator only installs and configures software to provide a new service. Finally, since VNFs are implemented as software, VNFs can be easily added, updated, and deleted. It is also easy to adjust the performance of VNFs according to customers' demands.

Service chaining (SC) relies on NFV technologies for the virtualization of network functions [2]. SC delivers the traffic of a specific service along a predefined ordered list of VNFs. Developments in network control and transmission technologies have increased the potential for providing SC in practical network systems. Software-defined networking (SDN) enables dynamic and flexible control of network traffic, which facilitates network operators to steer the traffic so as to create a chain of VNFs [3]. Optical networks have expanded the range of services that can be provided by using SC thanks to its high transmission capacity. A number of literatures worked on provisioning SC in optical networks, including intra- and inter-datacenter optical networks [4, 5, 6].

In order to provide SC in the network, it is necessary to determine the placement of required VNFs and the route of SC path from a source node to a destination node. The SC path must pass through all the required VNFs in order and meet the bandwidth requirement of the service and the allowable delay of user. This problem is called VNF placement and routing (VNF-PR) problem for SC [7]. The objective of VNF-PR problem varies depending on problem settings: minimizing the cost related to VNF placement and link utilization, maximizing the number of services that can be provided in the network, and so on.

SC is also studied in the multicast communication [8, 9, 10, 11, 12, 13]. The multicast communication saves bandwidth consumption compared to the unicast communication. In order to send the same data to multiple nodes, the multicast builds a branching path from a source node to destination nodes, while the unicast builds multiple paths from a source node to every destination node. In the multicast, packets are replicated at every branch point on the path and finally received by all destination nodes. The multicast communication is used for applications such as video conferences, software updates, and internet protocol televisions. Some of these multicast services are bandwidth-intensive, and can be supported by high-capacity networks such as optical networks [14, 15].

The multicast service chaining (MSC) provides a branching SC path which connects a source node and multiple destination nodes. VNFs are placed in a predefined order on the SC path so that users on the destination nodes receive their desired services. MSC reduces the cost for VNF placement and link usage compared to the traditional (i.e., unicast) SC.

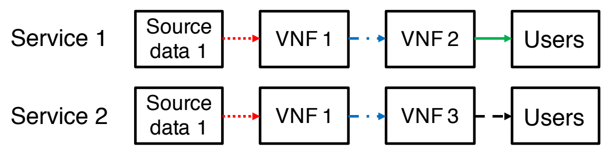
When a network operator provides a service to multiple users by using MSC, it is necessary to consider the VNF-PR problem for MSC. The conventional MSC model [8] configures SC paths for services, each of which has the same source data and the same set of VNFs in a predefined order.

In recent years, applications which provide the same type of data to multiple devices with different characteristics such as personal computers, smart phones and televisions, have been emerged. It is conceivable that there are multiple SC requests that require the same source data but different sets of VNFs in their service chains. When the conventional MSC model is applied to such SC requests, a waste of bandwidth utilization occurs since the conventional MSC model configures SC paths for each service separately. If a part of SC paths of different services where the same data are carried can be merged into one path, the bandwidth utilization efficiency further increases. However, such a model of VNF-PR has not existed.

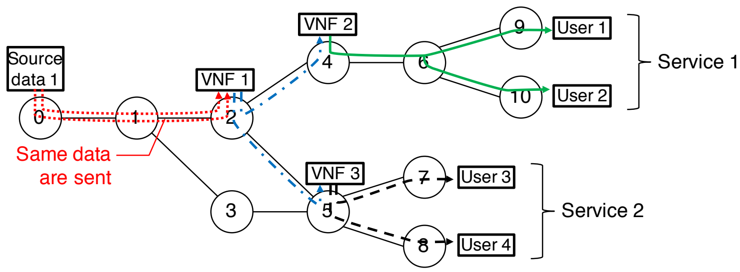
This paper proposes a VNF-PR model for multicast service chaining based on merging multiple service paths (MSC-M). The proposed model allows merging a part of SC paths into a single path if the paths carry the same data even if the paths provide different services. We formulate the MSC-M model as an integer linear programming (ILP) problem which minimizes the cost associated with VNF placement and link usage in providing multicast service chains. We develop a heuristic algorithm for MSC-M for the case that a feasible solution cannot be obtained by solving the ILP problem within practical time. We evaluate the performances of the MSC-M model in terms of the total cost for VNF placement and link utilization to accommodate SC requests. Numerical results show that the MSC-M model reduces the total cost required to accommodate service chaining requests compared to the conventional MSC model.

This paper is an extended version of [16], which mainly focused on modeling of the MSC-M model. The extensions to the work in [16] are mainly described as follows. We prove that the decision version of the VNF-PR problem based on the MSC-M model is NP-complete. We introduce a heuristic algorithm for the MSC-M model. We evaluate the MSC-M model with the heuristic algorithm in terms of total cost and computation time. We discuss directions for extending the MSC-M model to an optical domain. We extensively survey existing researches related to our work.

The rest of the paper is organized as follows. Section 2 presents the related works. Section 3 presents the conventional model. Section 4 describes the



(a) VNF order of services.



(b) VNF placement and routing.

Figure 1: Example of conventional multicast service chaining (MSC).

MSC-M model including the ILP formulation and the proof of NP-completeness. Section 5 introduces the heuristic algorithm for the MSC-M model. Section 6 shows numerical results. Section 7 presents directions for extending the MSC-M model to an optical domain. Finally, Section 8 concludes this paper.

2. Related works

The resource allocation problem for SC has attracted attention along with the emergence and development of network virtualization technologies including SDN and NFV. According to the classification provided in [17], the resource allocation in the NFV environment consists of three steps: the VNF chain composition, the VNF forwarding graph embedding, and the VNF scheduling. The VNF-PR problem studied in this paper corresponds to the VNF forwarding graph embedding step, which determines the mapping of virtual resources to substrate resources.

There have been few works on the VNF-PR problem for MSC [8, 9, 10, 11, 12, 13]. Yi et al. [9] presented a multi-stage algorithm for solving the VNF-PR problem, which first constructs a traffic forwarding graph and a function delivery graph and then determines the traffic steering for each multicast session. The algorithm in [9] can be applied to a hybrid network infrastructure where traditional purpose-built hardware for specific network functions and general-purpose hardware for VNFs coexist. Cheng et al. [10] developed a heuristic algorithm for the VNF-PR problem based on constructing a Steiner tree. Alhussein et al. [11] studied the VNF-PR problem for a multicast service in 5G core networks. The problem formulated in [11] allows multipath traffic routing between embedded VNFs to improve the flexibility of VNF placement and routing. Ren et al. [12] formulated the service function tree embedding problem by ILP and presented a two-stage algorithm which has the approximation ratio of $1 + \rho$, where ρ is the best approximation ratio of the Steiner tree problem. Guler et al. [13] designed a heuristic algorithm which maps a multicast SC request onto a substrate network based on the technique named level-based path splitting.

All of the above existing works considered the merging of data flows forwarded to destination nodes of the same service to construct a multicast tree. In this paper, in addition to the merging of data flows of the same service, we consider that of different services to further improve the bandwidth utilization efficiency of the physical network.

3. Conventional model

Zhang et al. [8] studied the MSC problem, which is referred to as the service function chain enabled multicast routing problem (SMRP) in their paper. The SMRP is a problem to find the VNF placement and routing for a multicast service chain. The goal of the SMRP is minimizing the total cost to provide the multicast service chain, which consists of the costs of link utilization and running VNF instances. A heuristic algorithm to solve the SMRP is presented in [8]. The heuristic algorithm first finds a path from the source node to one of the destination users via the chain of VNFs, and then the route to other destination users is determined based on the total cost of the path. The SMRP is a problem which determines the VNF placement and routing only for a single service chain. To configure SC paths for multiple services, the SMRP needs to be iteratively solved for each service.

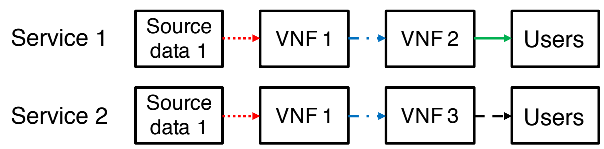
Figure 1 shows an example of VNF placement and routing by using the conventional MSC model including the SMRP. We consider a network whose nodes have a capability to handle data flow transfer and deploy VNF instances [7]. Two SC paths are mapped on the network to provide services 1 and 2. Both services use the same source data provided by the same source node 0. Each service accommodates two users; users 1 and 2 receive service 1 and users 3 and 4 receive service 2. In each SC path, VNFs are deployed and sequentially chained from a source node to each destination node. It can be seen that two SC paths carry the same data in link (0,1) and (1,2) in Fig. 1.

4. Proposed model

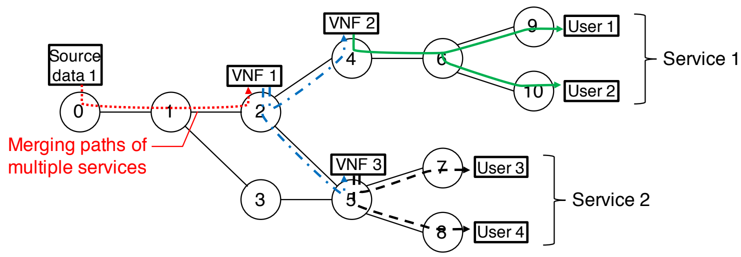
4.1. Model description

This section describes the MSC-M model. In this model, when SC paths of different services which have the same type of source data pass through the same link, the SC paths can be merged into a single path in that link as long as the set and order of VNFs already passed are the same. In Fig. 1, SC paths of services 1 and 2, which have the same source data, pass through links (0, 1) and (1, 2) separately. Figure 2 shows an example of VNF placement and routing based on the MSC-M model. In contrast to Fig. 1, since the two SC paths carry the same data at links (0, 1) and (1, 2), the SC paths are merged into a single path in these links. The MSC-M model saves the total amount of link cost for provisioning SC paths compared with the conventional MSC model.

Since the amount of computation resources of servers on the network is limited, we also consider the reduction of the number of VNF instances deployed on the network. We assume that each node has a limited number of CPU cores, each of which can deploy a VNF instance. Each VNF instance can handle data



(a) VNF order of services.



(b) VNF placement and routing.

Figure 2: Example of multicast service chaining based on merging multiple service paths (MSC-M).

flows of multiple SC paths up to a certain maximum rate. If needed, a VNF instance can be replicated on the same node at the cost of an additional core.

4.2. ILP formulation

We formulate the MSC-M model described in Section 4.1 as an ILP problem.

Model parameters

Table 1 gives the summary of sets and parameters used in the ILP formulation. The network is modeled as a bidirected graph $G = (V, A)$. V is a set of nodes. A is a set of links. Each node $u \in V$ has c_u of CPU cores for deploying VNF instances. Each link $(u, v) \in A, u \in V, v \in V$ has bandwidth capacity b_{uv} and delay amount l_{uv}. We refer to F as the set of types of VNFs. The maximum processing capacity m_f is defined for each VNF $f \in F$, which limits the number of data flows of SC paths that can be processed by one VNF instance. The set of users is denoted by K and each user $k \in K$ requests service $r \in R$ with the allowable delay amount l_k. The destination node of user $k \in K$ is represented as $d_k \in V$. R is a set of services. Service $r \in R$ is characterized by set of VNFs

F_r and passing order of VNFs $p_r(f, g)$. The source node of service $r \in R$ is represented as $s_r \in V$. $p_r(f, g)$ is a binary operator which is inspired by [4]. $p_r(f, g)$ maps a tuple of VNFs (f, g) to the value -1 (resp. 1) if VNF f appears after (resp. before) VNF g along an SC path of service $r \in R$. The number of VNFs required by service r is $|F_r|$, and the chain of service r is divided into $|F_r| + 1$ sections. A set of the sections is denoted by $N_r = \{1, 2, \cdots, |F_r| + 1\}$. We introduce a parameter called *group*, $\beta \in D$, to distinguish types of data flows transferred on SC paths. D is a set of groups. Data flows whose source data is the same and which already passed the same subset of VNFs belong to the same group. A part of SC paths which carry data flows belonging to the same group can be merged into a single path. Figure 3 shows the relationship between the section number and the group number. $\theta(r, n)$ is a parameter which determines the group number, $\beta \in D$, from section number $n \in N_r$ and service r. Bandwidth b_β is required to send data of group β.

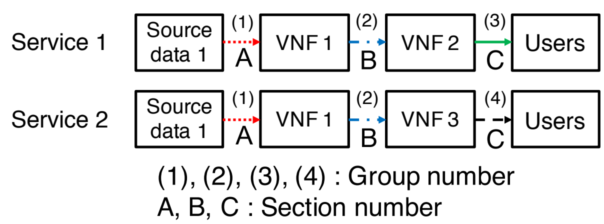


Figure 3: Relationship between section number and group number.

Decision variables

Table 2 gives the summary of decision variables used in the ILP formulation. We denote by q_{uv}^{krn} the routing binary variables that state whether link $(u, v) \in A$ is used for the path for section $n \in N_r$ of service $r \in R$ that user $k \in K$ requests or not. We denote by a_{uv}^{β} the routing binary variables that state whether link $(u, v) \in A$ is used for the path for group $\beta \in D$ or not. The route of data flow from the source node of service $r \in R$, s_r, to the destination node of user $k \in K$, d_k, can be derived from q_{uv}^{krn}. The number of required instances of VNF $f \in F$ on node $u \in V$ is denoted by x_u^f. We denote by $y_u^{\beta f}$ the binary variable stating

Table 1: Sets and parameters

Sets and parameters	**Definitions**
V	Set of nodes
A	Set of links
K	Set of users
R	Set of services
F	Set of VNFs
F_r	Set of VNFs requested by service $r \in R$
N_r	Set of nonnegative integers less than or equal to the number of sections of service $r \in R$
D	Set of groups
s_r	Source node of service $r \in R$
d_k	Destination node of user $k \in K$
b_β	Bandwidth demand of group $\beta \in D$
c_u	Number of CPU cores of node $u \in V$
m_f	Maximum rate for VNF $f \in F$
$p_r(f, g)$	1 (resp. -1) if VNF $f \in F$ appears before (resp. after) VNF $g \in F$ on SC path of service $r \in R$
b_{uv}	Bandwidth of link $(u, v) \in A$
l_{uv}	Latency of link $(u, v) \in A$
l_k	Maximum tolerated latency for user $k \in K$
r_k	Services requested by user $k \in K$
$\theta(r, n)$	Parameter for determining group number $\beta \in D$ from section number $n \in N_r$ of service $r \in R$

whether VNF $f \in F$ handles data of group $\beta \in D$ on node $u \in V$ or not. We denote by z_u^{krnf} the binary variable stating whether VNF $f \in F$ handles data of section $n \in N_r$ of service $r \in R$ that user $k \in K$ requests on node $u \in V$ or not. Y_u^{krf} accounts for the state of data flow on node $u \in V$; $Y_u^{krf} = 1$ if the data flow toward user $k \in K$ has already been assigned to VNF $f \in F$ at any one of the nodes along its route from s_r to node u, including u itself.

Table 2: Decision variables

Decision variables	**Descriptions**	**Types**
Y_u^{krf}	1 if data flow destined to user $k \in K$ requiring service $r \in R$ meets its assigned VNF $f \in F$ before or on node $u \in V$	Binary
$y_u^{\beta f}$	1 if VNF $f \in F$ handles data of group $\beta \in D$ on node $u \in V$	Binary
z_u^{krnf}	1 if VNF $f \in F$ handles data of section $n \in N_r$ of service $r \in R$ that user $k \in K$ requests on node $u \in V$	Binary
x_u^f	Number of instances of VNF $f \in F$ located on node $u \in V$	Integer
q_{uv}^{krn}	1 if SC path for section $n \in N_r$ of service $r \in R$ that user $k \in K$ requests contains link $(u, v) \in A$	Binary
a_{uv}^{β}	1 if link $(u, v) \in A$ is used for path for group $\beta \in D$	Binary
p_{uv}^{kr}	Section number in link $(u, v) \in A$ of the data of service $r \in R$ requested by user $k \in K$	Integer

Objective function

The objective function is to minimize the total cost for link utilization and VNF placement while accommodating all service chain requests, as

$$\min \sum_{(u,v)\in A} \sum_{\beta\in D} \delta_{uv} a_{uv}^{\beta} + \sum_{u\in V} \sum_{f\in F} \varepsilon_u x_u^f. \tag{1}$$

The first term except δ_{uv} represents the sum of the number of links passed by SC paths. The second term except ε_u represents the total number of placed VNF instances. δ_{uv} and ε_u are the link cost of $(u,v) \in A$ and the cost for placing a VNF instance on node $u \in V$, respectively. δ_{uv} and ε_u should be set based on the actual operation policy of network operators.

Constraints

We introduce constraints in the ILP formulation of the MSC-M model.

Equation (2) enforces the conservation of data flow between the source node and each destination node of service r. This constraint treats each data flow from a source node to a destination node as if it is a unicast flow.

$$\sum_{n\in N_r} \sum_{(u,v)\in A} q_{uv}^{krn} - \sum_{n\in N_r} \sum_{(v,u)\in A} q_{vu}^{krn} = \begin{cases} 1 & \text{if } u = s_r \\ -1 & \text{if } u = d_k \\ 0 & \text{otherwise,} \end{cases} \tag{2}$$

$$\forall k \in K, \quad \forall r \in R : r = r_k, \quad \forall u \in V$$

Equation (3) means that, if a data flow of service r destined to user k passes through link (u,v) and the flow belongs to group β in this link, a part of SC path which carries data of group β passes through that link.

$$a_{uv}^{\beta} \geq q_{uv}^{krn}, \quad \forall k \in K, \quad \forall r \in R : r = r_k, \quad \forall (u,v) \in A, \quad \forall n \in N_r, \quad \forall \beta \in D : \beta = \theta(r,n) \tag{3}$$

We denote by (4), (5) and (6) constraints on SC paths that must not go through the same link twice or more. Equation (4) is a constraint prohibiting a data flow of group β from passing twice through the same link. Equation (5) is a constraint prohibiting a data flow which is carried by section n of SC path of service r toward user k from passing twice through the same link. Equation (6) means that at most one section of SC path of service r toward user k is mapped on link (u, v).

$$\sum_{(v,u)\in A} a_{vu}^{\beta} \leq 1, \quad \forall \beta \in D, \quad \forall u \in V \tag{4}$$

$$\begin{gathered} \sum_{v\in V:(u,v)\in A} q_{uv}^{krn} \leq 1, \\ \forall k \in K, \quad \forall r \in R : r = r_k, \quad \forall n \in N_r, \quad \forall u \in V \end{gathered} \tag{5}$$

$$\begin{gathered} \sum_{n\in N_r} q_{uv}^{krn} \leq 1, \\ \forall k \in K, \quad \forall r \in R : r = r_k, \quad \forall (u, v) \in A \end{gathered} \tag{6}$$

Equation (7) is a constraint that the bandwidth limitation of each link is not violated. Equation (8) is a constraint of the maximum delay tolerance of user k.

$$\sum_{\beta\in D} a_{uv}^{\beta} b_{\beta} \leq b_{uv}, \quad \forall (u, v) \in A \tag{7}$$

$$\sum_{(u,v)\in A} \sum_{n\in N_r} q_{uv}^{krn} l_{uv} \leq l_k, \quad \forall k \in K, \quad \forall r \in R : r = r_k \tag{8}$$

Equation (9) states that a node cannot support more VNF instances than its number of available CPU cores. Equation (10) states that each VNF instance has a limited capacity and needs to be duplicated enough to handle all data flows which request the VNF on the corresponding node [7].

$$\sum_{f\in F} x_u^f \leq c_u, \quad \forall u \in V \tag{9}$$

$$\sum_{\beta\in D} y_u^{\beta f} b_{\beta} \leq m_f x_u^f, \quad \forall u \in V, \quad \forall f \in F \tag{10}$$

Equation (11) states that VNF $f \in F$ on node $u \in V$ is used by data flow of group $\beta \in D$ if at least one user whose data flow belongs to group $\beta \in D$ uses VNF f on node u.

$$\begin{gathered} z_u^{krnf} \leq y_u^{\beta f}, \\ \forall k \in K, \quad \forall r \in R : r = r_k, \quad \forall n \in N_r, \\ \forall \beta \in D : \beta = \theta(r, n), \quad \forall f \in F, \quad \forall u \in V \end{gathered} \tag{11}$$

Equation (12) states that if a service requests VNF f, VNF f is always placed on the source node or the route to each destination node.

$$\begin{gathered} \sum_{u \in V} \sum_{n \in N_r} z_u^{krnf} = \begin{cases} 1 & \text{if } f \in F_r \\ 0 & \text{otherwise,} \end{cases} \\ \forall k \in K, \quad \forall r \in R : r = r_k, \quad \forall f \in F \end{gathered} \tag{12}$$

Equation (13) states that at the source node of service r, a data flow destined to each user has met none of the requested VNFs. Equation (14) states that at the destination node of each user receiving service r, a data flow has met all the VNFs requested by service r.

$$Y_{s_r}^{krf} = 0, \ \forall r \in R, \ k \in K : r = r_k, \ \forall f \in F \tag{13}$$

$$Y_{d_k}^{krf} = 1, \ \forall r \in R, \ k \in K : r = r_k, \ \forall f \in F_r \tag{14}$$

Equation (15) means that, when link (u, v) is used for data flow towards user k and the data flow meets its assigned instance of VNF f on node v, the value of Y_v^{krf} increases from 0 to 1.

$$\begin{gathered} (\sum_{n \in N_r} q_{uv}^{krn} - 1) + (Y_v^{krf} - Y_u^{krf}) \leq \sum_{n \in N_r} z_v^{krnf}, \\ \forall k \in K, \quad \forall r \in R : r = r_k, \quad \forall f \in F_r, \quad \forall (u, v) \in A \end{gathered} \tag{15}$$

Equation (16) accounts for the order of VNFs on the SC path.

$$\begin{gathered} (Y_u^{krf} - Y_u^{krg}) p_r(f, g) \geq 0, \\ \forall k \in K, \quad \forall r \in R : r = r_k, \quad \forall u \in V, \quad \forall f, g \in F_r \end{gathered} \tag{16}$$

Equation (17) states that a requested VNF can be deployed on a node only if the data flow passes through this node.

$$z_u^{krnf} \leq \sum_{(v,u)\in A} q_{vu}^{krn}, \quad \forall k \in K,\ \forall r \in R : r = r_k,\ \forall n \in N_r,\ \forall u \in V,\ \ \forall f \in F \tag{17}$$

Equations (18) and (19) ensure that section number n increases by the number of VNFs used at node u on the data flow. Equation (20) is a constraint for setting p_{uv}^{kr} of any egress link of the destination node to zero, since there is no data flow from the destination node when considering the flow for each user. Equation (21) is a constraint for setting p_{uv}^{kr} of any egress link of source node of service $r \in R$; p_{uv}^{kr} is set to one if data flow to user k passes link (u, v).

$$\sum_{(v,u)\in A} \sum_{n\in N_r} q_{vu}^{krn} + \sum_{f\in F_r} Y_u^{krf} = \sum_{(u,w)\in A} p_{uw}^{kr}, \quad \forall k \in K,\ \ \forall r \in R : r = r_k,\ \ \forall u \in V : u \neq s_r, d_k \tag{18}$$

$$p_{uv}^{kr} = \sum_{n\in N_r} n q_{uv}^{krn}, \quad \forall k \in K,\ \ \forall r \in R : r = r_k,\ \ \forall (u, v) \in A \tag{19}$$

$$p_{uv}^{kr} = 0, \quad \forall k \in K,\ \ \forall r \in R : r = r_k,\ \ \forall (u, v) \in A : u = d_k \tag{20}$$

$$p_{uv}^{kr} \leq 1, \quad \forall k \in K,\ \ \forall r \in R : r = r_k,\ \ \forall (u, v) \in A : u = s_r \tag{21}$$

4.3. NP-completeness

We prove that the VNF-PR decision problem based on the MSC-M model is NP-complete. We define the VNF-PR decision problem based on the MSC-M model as follows:

Problem. Graph $G = (V, A)$, the number of CPU cores c_u and VNF placement cost ε_u for node $u \in V$, the bandwidth capacity b_{uv}, delay l_{uv}, and link utilization cost δ_{uv} for link $(u, v) \in A$, a set of VNF types F, the maximum processing capacity m_f for VNF $f \in F$, a set of users K, a set of services R, the source node s_r and a set of VNFs F_r with the passing order for service $r \in R$, the destination node d_k and the allowable delay amount l_k for user $k \in K$, a set of data groups D, and the required bandwidth for sending data b_β for group $\beta \in D$ are given. Is there any set of VNF placement and routing for services in R such that the total required cost is at most C?

Theorem. The VNF-PR decision problem based on the MSC-M model is NP-complete.

Proof. If any instance of the problem is given, we can verify whether the instance provides the total cost at most C in polynomial time of $O(|A||D| + |V||F|)$. Therefore, the VNF-PR decision problem based on the MSC-M model is in NP.

Let T be a subset of node V. Let $|R| = 1$, $|F_r| = 1$, and T consist of one source node and $(|T| - 1)$ destination nodes of service $r \in R$. When $c_u = 1$ for all destination nodes and $c_u = 0$ for other nodes, $\varepsilon_u = 0$, $b_{uv} = 1$, $l_{uv} = 1$, $m_f = 1$, $l_k = |A|$, and $b_\beta = 1$, the VNF-PR decision problem based on the MSC-M model is equivalent to the minimum Steiner tree problem, which is a well-known NP-complete problem [18]. Given a graph, a subset of nodes, and the cost of links, the minimum Steiner tree problem finds a tree which provides the total cost at most C. In other words, the minimum Steiner tree problem is a subset of the VNF-PR decision problem based on the MSC-M model. Therefore, the VNF-PR decision problem based on the MSC-M model is NP-complete.

5. Heuristic algorithm

As the scale of the problem grows, the ILP problem for the MSC-M model introduced in Section 4.2 may not be able to be solved within practical time. In this section, we present a heuristic algorithm, named the combining path and tree (CPT) algorithm, to solve the VNF-PR problem based on the MSC-M model.

Algorithm 1 Combining path and tree (CPT) for MSC-M model

Decide the processing order of $r \in R$.
Define o_j = Index of service which is processed in jth order.
Define δ_{uv} = Cost of link $(u, v) \in A$.
Define ε_u = Cost of placing VNF on node $u \in V$.
for $j = 1, \cdots, |R|$ **do**
    If the same type of VNF requested first in service o_j is already placed in node $v, \varepsilon_v^{\mathrm{P}} = 0$; otherwise $\varepsilon_v^{\mathrm{P}} = \varepsilon_v$.
    If the data of the group that the first path section belongs to already exist in link $(u, v), \delta_{uv}^{\mathrm{P}} = 0$; otherwise $\delta_{uv}^{\mathrm{P}} = \delta_{uv}$.
    Define $h_1^{o_j} = \arg\min_{v \in H_1^{o_j} : c_v^{\mathrm{used}} < c_v} (\lambda(s_{o_j}, v) + \varepsilon_v^{\mathrm{P}})$.
    If a new VNF is deployed $c_{h_1^{o_j}}^{\mathrm{used}} = c_{h_1^{o_j}}^{\mathrm{used}} + 1$
    for $i = 2, \cdots, |F_{o_j}|$ **do**
        If the same type of VNF requested in i th in service o_j is already placed in node $v, \varepsilon_v^{\mathrm{P}} = 0$; otherwise $\varepsilon_v^{\mathrm{P}} = \varepsilon_v$.
        If the data of the group that the i th path section belongs to already exist in link (u, v), $\delta_{uv}^{\mathrm{P}} = 0$; otherwise $\delta_{uv}^{\mathrm{P}} = \delta_{uv}$.
        Define $h_i^{o_j} = \arg\min_{v \in H_i^{o_j} : c_v^{\mathrm{used}} < c_v} (\lambda(h_{i-1}^{o_j}, v) + \varepsilon_v^{\mathrm{P}})$.
        If a new VNF is deployed $c_{h_i^{o_j}}^{\mathrm{used}} = c_{h_i^{o_j}}^{\mathrm{used}} + 1$
    end for
    If the data of the group that the last path section belongs to already exist in link $(u, v), \delta_{uv}^{\mathrm{P}} = 0$; otherwise $\delta_{uv}^{\mathrm{P}} = \delta_{uv}$.
    Define $h_{\mathrm{dst}}^{o_j} = \arg\min_{d_k : r_k = o_j, k \in K} (\lambda(h_{|F_{o_j}|}^{o_j}, d_k))$.
    Construct path P_{o_j} by connecting $s_{o_j}, h_1^{o_j}, \ldots, h_{\mathrm{dst}}^{o_j}$.
end for
for $j = 1, \cdots, |R|$ **do**
    If the data of the group that the last path section belongs to already exist in link $(u, v), \delta_{uv}^{\mathrm{P}} = 0$; otherwise $\delta_{uv}^{\mathrm{P}} = \delta_{uv}$.
    Construct a complete graph C_{o_j} whose nodes correspond to users of service o_j. The link cost between any two users in C_{o_j} is the sum of the link costs of the shortest path between the two users in G, which is computed by using the amended link cost δ_{uv}^{P}.
    Find the minimum spanning tree T_{o_j} from C_{o_j}.
    Connect P_{o_j} with T_{o_j}, and call the combined graph G_{o_j}.
    If there is a user who cannot satisfy the delay constraint in service o_j, a path from s_{o_j} to the user with the smallest latency on G is computed and added to G_{o_j}.
end for
Replace each link in $G_1, \cdots, G_{|R|}$ with the shortest path in G. Call the result multicast topology $T_{\min}$.
Remove any unnecessary edges in $T_{\min}$ to get final output.

5.1. Algorithm description

The CPT algorithm for the MSC-M model is illustrated in Algorithm 1. This algorithm is an extension of the heuristic algorithm for the conventional MSC model, which is presented in [8], to the MSC-M model. We introduce the amended costs of link and VNF placement in this algorithm. The VNF placement and the route of each SC path are computed based on these amended costs. The values of amended costs are adjusted in the process of the algorithm so that as many SC paths which carry the same data as possible are merged. The amended link cost of link $(u, v) \in A$ and the amended VNF placement cost of $u \in V$ are denoted as δ^{P}_{uv} and $\varepsilon^{\mathrm{P}}_{u}$, respectively. The procedure for computing the VNF placement and the route of SC paths in the CPT algorithm is as follows.

First, the processing order of services is determined. For example, the processing order can be determined based on the number of requested VNFs of each service and the similarity of requested VNFs between services. The service which is processed in the jth order is denoted by o_j. A service which requests the largest number of VNFs is selected as a service to be processed first, o_1. Then a service which has the same source as o_1 and has the longest common VNF sequence from the source with o_1 is selected among the unselected services; the newly selected service becomes o_2. In a similar way, a service which is processed in the jth order is iteratively selected based on the length of common VNF sequence from the source with o_{j-1}. If all of the services which have the same source as o_1 have been selected, a service which requests the largest number of VNFs among the unselected services is selected as the service to be processed next, and this service is followed by services which have the same source and the common VNF sequence in the same way as o_1.

For each service, an SC path from the source node to one of the users requesting the service is computed. This path is denoted by P_{o_j}. In Algorithm 1, $\lambda(u, v)$ is a function that returns the sum of the link costs of the shortest path between nodes u and v, which is computed by using the amended link cost δ^{P}_{uv}. The route of P_{o_j} is determined step-by-step, i.e., for each section between the source node to the first VNF, between consecutive VNFs, and the last VNF to the user node. Each of these sections composing P_{o_j} is called a path section hereafter. A link that does not satisfy the bandwidth constraint is not selected as a route of P_{o_j}. Also, a node that does not satisfy the constraint of available CPU cores is not considered as a candidate node for installing a VNF. The number of CPU cores on node v which are already used for placing VNFs is denoted as c_v^{used} in Algorithm 1. When computing the path section from the source node to the first VNF, the node that can be reached with the lowest cost, $h_1^{o_j}$, is selected from a set of nodes in which the first VNF can be installed, $H_1^{o_j}$. The cost of path section is calculated by using the amended cost of link (u, v), δ_{uv}^{P}, and the amended VNF placement cost of node v, ε_v^{P}. δ_{uv}^{P} is set to zero if the data of the same group already exists in the link. ε_v^{P} is set to zero if the same type of VNF instance is already placed in node v where the VNF can be placed. The path section between the first VNF and the second VNF is computed in the same way as the first path section. Similarly, the path sections are computed until the path reaches the node of one user, $h_{\text{dst}}^{o_j}$.

After determining P_{o_j} for all $o_j, j = 1, \cdots, |R|$, a complete graph C_{o_j} whose nodes correspond to users is constructed for each service o_j. The link cost between any two users in C_{o_j} is the sum of the link costs of the shortest path between the two users in G, which is computed by using the amended link cost δ_{uv}^{P}. The minimum spanning tree (MST) among users, T_{o_j}, is computed on C_{o_j}. T_{o_j} and P_{o_j} are combined as a tree, which is denoted as G_{o_j}. If there is a user who cannot satisfy the delay constraint in service o_j, a path which connects s_{o_j} and the user with the smallest latency on G is computed and added to G_{o_j}. VNFs required to provide o_j are additionally placed on nodes on the path so that the data flow towards the user passes through them in the order on the path. G_{o_j}, $j = 1, \cdots, |R|$, are mapped to the physical network, G, so that each link in G_{o_j} corresponds to the shortest path in G which connects the two end point nodes of the link. The mapped graph is denoted as $T_{\min}$. Finally, a tree for providing all services is obtained by removing redundant parts of $T_{\min}$.

5.2. Computational time complexity

The computational time complexity of CPT algorithm is analyzed as follows.

First, the CPT algorithm determines the processing order of services. Let us consider that the processing order is determined based on the number of requested VNFs of each service and the similarity of requested VNFs between services, as described in Section 5.1. A service which requests the largest number of VNFs is found in $O(|R|)$. A service which has the same source and has the longest common VNF sequence from the source with the previously selected service is found in $O(|R||F|)$. As a result, the processing order of $|R|$ services is determined in $O(|R|^2|F|)$.

In the computation of P_{o_j}, routes of $|F_{o_j}|+1$ path sections are determined. Each service requests up to $|F|$ VNFs, so the number of path sections can be $|F|+1$ at a maximum. Amended costs $\varepsilon_u^{\mathrm{P}}$ and δ_{uv}^{P} are computed in $O(|V||F|)$ and $O(|A||D|)$, respectively. The route of each path section is selected based on the shortest path between two nodes, which can be obtained in $O(|A|\log|V|)$ by using Dijkstra's algorithm. As a result, $O(|R||F|(|V||F|+|A||D|+|A|\log|V|))$ is required to determine P_{o_j} for $|R|$ services.

In the computation of G_{o_j}, amended cost δ_{uv}^{P} is computed in $O(|A||D|)$ first. The cost of each link in complete graph C_{o_j} is determined based on the shortest path between two user nodes in G. Each service contains up to $|K|$ users, so $O(|K||A|\log|V|)$ is required to determine the link costs in C_{o_j}, where at most $|K|$ shortest path trees are computed. The number of links in complete graph C_{o_j} is $\frac{|K|(|K|-1)}{2}$. Therefore, minimum spanning tree of C_{o_j} is computed in $O(|K|^2\log|K|)$ by using Kruskal's algorithm [19]. At most $|K|$ users do not satisfy the delay constraint. A path connecting s_{o_j} and each user of o_j with the smallest latency on G is computed in $O(|A|\log|V|)$. For each user who does not satisfy the delay constraint, VNFs required to provide service o_j are additionally placed on nodes on the path, which requires $O(|V||F|)$. As a result, $O(|R||A||D|+|R||K|(|A|\log|V|+|K|\log|K|)+|R||K||V||F|)$ is required to determine G_{o_j} for $|R|$ services.

Finally, G_{o_j} is mapped to G so that each link in G_{o_j} corresponds to the shortest path in G. For each service, we need to compute the shortest path on G from the node where the last VNF is placed to each user node. Therefore, $O(|R||A|\log|V|)$ is required to map G_{o_j} of $|R|$ services to G.

The overall time complexity of CPT algorithm is $O(|R|^2|F|+|R||F|(|V||F|+|A||D|+|A|\log|V|)+|R||K|(|A|\log|V|+|K|\log|K|)+|R||K||V||F|)$.

6. Numerical results

We investigate the MSC-M model in terms of total cost (i.e., the objective value obtained by (1)) required to accommodate SC requests and the computation time. We compare the total cost and the computation time of MSC-M model with those of three models; the unicast service chaining (USC) model, the conventional MSC (MSC-C) model, and the MSC with incidental merging (MSC-I) model. The USC model configures a unicast SC path for every destination node. The MSC-C model configures a branching SC path for every service. Note that the USC model and the MSC-C model do not merge multiple SC paths even if the paths carry the same data in the same link. The MSC-I model computes the route of SC paths in the same way as the MSC-C model, and merges SC paths if the paths carry the same data in the same link. Each of the three models compared to the MSC-M model is formulated as an ILP problem that minimizes the total cost associated with VNF placement and link usage in the same way as (1). The CPT algorithm is applied for the MSC-C model and the MSC-I model with a modification. The modification from the CPT algorithm for the MSC-M model (i.e., Algorithm 1) is that the original link cost, δ_{uv}, is always used in the computation of SC paths instead of the amended link cost, δ^{P}_{uv}.

6.1. Evaluation on small number of services

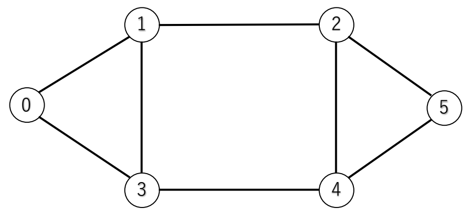
6.1.1. Evaluation setup

First, we begin with the results of total cost and computation time under the condition that the number of services, $|R|$, is small.

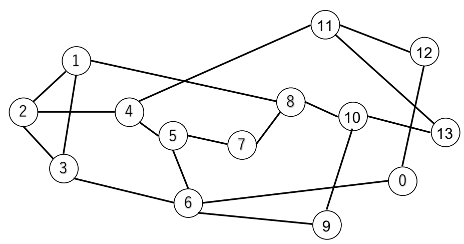
We evaluate the MSC-M model over two networks shown in Fig. 4, including the six-node network (6 nodes and 8 links) and NSFNET (14 nodes and 19 links). We set $c_u = 2$, $b_{uv} = 100$, $l_{uv} = 1$, $l_k = 100$, $b_\beta = 1$, $b_r = 1$ and $m_f = 50$. b_r is the bandwidth required to send data of service $r \in R$ in the USC, MSC-C, and MSC-I models. Figure 5 shows services considered in this evaluation. Each service requests three VNFs in the order shown in the figure. All services have the same source node. In this evaluation, we change the number of services mapped on the network in the range of one to four. When the number of services is less than four, the services with smaller index number are mapped. Each service is provided to two users. In the CPT algorithm, the services are processed in the order of their index numbers. We set δ_{uv} in the objective function in (1) to 1. We set ε_u in the objective function in (1) to 1, 0.5, and 0.02. When $\varepsilon_u = 0.02$, the ratio of link cost and VNF placement cost is 50 : 1; in this case, the minimization of link cost is prioritized over that of VNF placement cost. We conduct 100 trials for every network, setting of ε_u, and the number of services. In each trial, the source node and destination nodes of each service are selected randomly so that these nodes are set in different nodes. Note that the destination nodes of different services can be set in the same node. We obtain the total cost and computation time of each model for each trial, and take the averages over 100 trials for each model. We use a server with Intel Xeon Silver 4114 2.20 GHz 10-core CPU and 64 GB memory through the evaluation. The ILP problem of each model is solved by the IBM(R) ILOG(R) CPLEX(R) Interactive Optimizer with version 12.8 [20].

6.1.2. Total cost

Table 3 compares the total cost for the MSC-M, MSC-C, and MSC-I models with ILP and CPT algorithm and the USC model with ILP when $\varepsilon_u = 1$. In both six-node network and NSFNET, the total cost obtained by solving the ILP of MSC-M model can be lower than those of the USC, MSC-C, and MSC-I models. In the case of six-node network, the total cost for the MSC-M model with ILP is reduced by 12.54% and 0.74% compared to that of the MSC-C and



(a) Six-node



(b) NSFNET

Figure 4: Networks used in evaluation on small number of services.

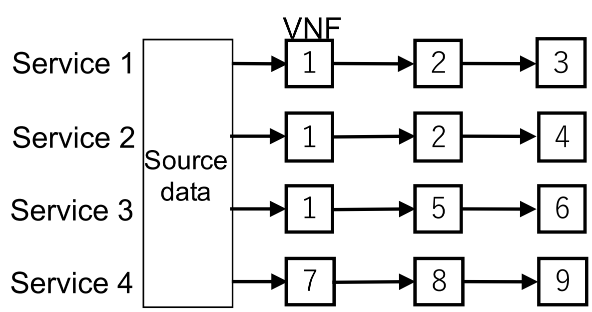


Figure 5: Services used in evaluation on small number of services.

MSC-I models with ILP, respectively, when $|R| = 4$. In the case of NSFNET, the total cost for MSC-M model with ILP is reduced by 14.10% and 6.00% compared to that of MSC-C and MSC-I models with ILP, respectively, when $|R| = 4$. When $|R| \geq 2$, the MSC-M model achieves smaller cost compared to the MSC-C model since the MSC-M model merges SC paths of different services. The MSC-M model also reduces the total cost compared to the MSC-I model. This is because the MSC-M model determines the VNF placement and routing so that SC paths of different services are merged whereas the MSC-I model searches for SC paths to be merged after computing each SC path separately. The cost reduction effect of the MSC-M model with ILP compared to the MSC-C and MSC-I models with ILP in the NSFNET is larger than that in the six-node network. This result indicates that there are more chances to merge multiple SC paths into a single path in the NSFNET compared to the six-node network due to the increase of average length of SC paths.

Table 3 also shows that the MSC-M model achieves smaller total cost than the MSC-C and MSC-I models in some cases of $|R| \geq 2$ when the CPT algorithm is adopted. When $|R| = 4$, the total cost obtained by the MSC-M model with CPT algorithm is reduced by 8.05% and by 7.78% in six-node network and NSFNET, respectively, compared to the MSC-C model with CPT algorithm.

Tables 4 and 5 compare the total cost for the MSC-M, USC, MSC-C, and MSC-I models when $\varepsilon_u = 0.5$ and $\varepsilon_u = 0.02$, respectively. When $\varepsilon_u = 0.02$ and $|R| = 4$, the total cost obtained by the MSC-M model with ILP is reduced by 19.91% and 30.86% in the six-node network and the NSFNET, respectively, compared to the MSC-C model with ILP. As is the case in $\varepsilon_u = 1$, the MSC-M model reduces the total cost compared to the MSC-C and MSC-I models when $|R| \geq 2$. The ratio of the total cost of MSC-M model to that of MSC-C model decreases as the value of ε_u decreases. Since the MSC-M model reduces the link cost by merging SC paths of different services, the cost reduction effect of the MSC-M model appears more clearly when ε_u is set to the smaller value.

Table 3: Total cost ($\varepsilon_u = 1$)

Networks	**Models**	Total cost			
		$\lvert R\rvert = 1$	$\lvert R\rvert = 2$	$\lvert R\rvert = 3$	$\lvert R\rvert = 4$
Six-node	USC (ILP)	8.49	15.62	24.12	35.07
	MSC-C (ILP)	6.64	11.77	18.28	25.93
	MSC-I (ILP)	6.64	10.12	16.15	22.84
	MSC-M (ILP)	6.64	9.91	15.85	22.68
	MSC-C (CPT)	6.69	11.52	18.43	27.33
	MSC-I (CPT)	6.69	10.52	16.43	25.15
	MSC-M (CPT)	6.69	10.52	16.43	25.13
NSFNET	USC (ILP)	9.64	18.67	27.62	38.37
	MSC-C (ILP)	7.54	14.03	21.21	29.20
	MSC-I (ILP)	7.54	12.45	18.99	26.68
	MSC-M (ILP)	7.54	11.83	17.79	25.08
	MSC-C (CPT)	8.54	15.04	23.04	31.60
	MSC-I (CPT)	8.54	13.04	21.04	29.30
	MSC-M (CPT)	8.54	13.04	20.86	29.14

Table 4: Total cost ($\varepsilon_u = 0.5$)

Networks	**Models**	Total cost			
		$\|R\| = 1$	$\|R\| = 2$	$\|R\| = 3$	$\|R\| = 4$
Six-node	USC (ILP)	6.72	12.63	19.96	29.93
	MSC-C (ILP)	4.95	9.20	14.31	20.38
	MSC-I (ILP)	4.95	7.90	12.37	17.59
	MSC-M (ILP)	4.95	7.23	11.51	17.45
	MSC-C (CPT)	5.16	9.32	15.15	22.13
	MSC-I (CPT)	5.16	7.32	13.15	19.88
	MSC-M (CPT)	5.16	7.32	13.15	19.78
NSFNET	USC (ILP)	8.05	15.72	23.48	32.66
	MSC-C (ILP)	6.04	11.64	17.62	24.26
	MSC-I (ILP)	6.04	10.49	15.70	21.59
	MSC-M (ILP)	6.04	9.20	13.62	19.33
	MSC-C (CPT)	6.70	12.77	19.42	27.07
	MSC-I (CPT)	6.70	10.77	17.52	24.55
	MSC-M (CPT)	6.70	10.77	17.52	24.38

Table 5: Total cost ($\varepsilon_u = 0.02$)

Networks	**Models**	Total cost			
		$\lvert R\rvert = 1$	$\lvert R\rvert = 2$	$\lvert R\rvert = 3$	$\lvert R\rvert = 4$
Six-node	USC (ILP)	4.56	9.52	15.89	25.42
	MSC-C (ILP)	3.44	7.00	10.76	16.16
	MSC-I (ILP)	3.44	5.86	8.88	13.16
	MSC-M (ILP)	3.44	4.91	7.67	12.94
	MSC-C (CPT)	3.76	7.39	11.51	17.31
	MSC-I (CPT)	3.76	5.39	9.51	14.90
	MSC-M (CPT)	3.76	5.39	9.51	14.83
NSFNET	USC (ILP)	6.27	12.11	18.96	25.57
	MSC-C (ILP)	4.73	9.23	13.93	18.58
	MSC-I (ILP)	4.73	8.11	12.07	15.97
	MSC-M (ILP)	4.73	6.73	9.01	12.85
	MSC-C (CPT)	5.30	11.21	16.23	22.27
	MSC-I (CPT)	5.30	9.21	14.23	19.80
	MSC-M (CPT)	5.30	9.21	14.08	19.59

6.1.3. Computation time

Table 6 compares the computation times for the MSC-M, USC, MSC-C, and MSC-I models in the NSFNET when $\varepsilon_u = 1$. As mentioned in the beginning part of Section 6, each of the USC, MSC-C, and MSC-I models is formulated as an ILP problem. The computation times of USC, MSC-C, and MSC-M models are the times required to solve the ILP of each model, which determines the VNF placement and routing of SC paths for all services. The computation time of MSC-I model consists of the time requried to solve the ILP problem, which determines the VNF placement and routing of SC paths for all services in the same way as the MSC-C model, and the time requried to compute merged SC paths. The computation time for solving the ILP in each model increases according to the number of services, $|R|$. The increasing rate of computation time of MSC-M model is larger than those of USC, MSC-C, and MSC-I models. This is because the ILP problem of MSC-M model has more decision variables than those of the MSC-C, MSC-I, and USC models in order to determine the VNF placement and link utilization of each data group.

The computation time of the MSC-M model is reduced by adopting the CPT algorithm instead of solving the ILP. In the settings of parameters considered in this evaluation, the computation time of MSC-M model with CPT algorithm is comparable to those of MSC-C and MSC-I models with CPT algorithm.

6.2. Evaluation on larger number of services

6.2.1. Evaluation setup

Next, we introduce the results of total cost and computation time under the condition that the number of services, $|R|$, is relatively large. We evaluate the MSC-C, MSC-I, and MSC-M models with CPT algorithm over the NSFNET. We set $\varepsilon_u = 1$ in this evaluation. Each service considered in this evaluation requests three VNFs out of six VNF types. A set of VNFs that each service requests and the order of them are randomly chosen in each trial. The processing order of services is determined based on the number of requested VNFs of each

Table 6: Computation time at NSFNET ($\varepsilon_u = 1$)

Models	Computation time (s)			
	$\|R\| = 1$	$\|R\| = 2$	$\|R\| = 3$	$\|R\| = 4$
USC (ILP)	0.09	0.37	1.02	2.19
MSC-C (ILP)	0.15	1.03	2.86	6.59
MSC-I (ILP)	0.15	1.04	2.87	6.60
MSC-M (ILP)	0.30	6.11	298.9	53.57×10^2
MSC-C (CPT)	1.29	1.30	1.28	1.31
MSC-I (CPT)	1.29	1.29	1.29	1.29
MSC-M (CPT)	1.29	1.28	1.31	1.27

service and the similarity of requested VNFs between services as described in Section 5.1. Other evaluation conditions are the same as Section 6.1.

6.2.2. Total cost

Figure 6 compares the total cost for the MSC-C, MSC-I, and MSC-M models with CPT algorithm. The MSC-M model achieves smaller total cost compared to the MSC-C and MSC-I models. Especially, the reduction effect of the total cost of MSC-M model compared to the MSC-C model increases as the number of services increases. This is because the chance that services that carries the same data appears increases according to the increase of the number of services.

6.2.3. Computation time

Figure 7 shows the computation times for the MSC-C, MSC-I and MSC-M models with CPT algorithm. The computation time of each model increases according to the number of services, $|R|$. We observe that, in each setting of $|R|$, the computation time of MSC-M model with CPT algorithm is comparable to those of MSC-C and MSC-I models with CPT algorithm.

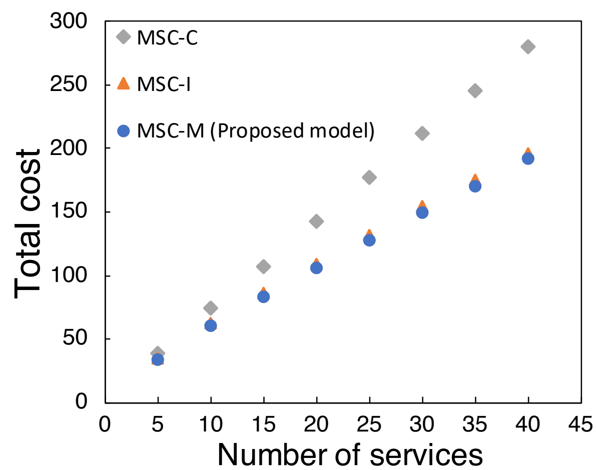


Figure 6: Total cost depending on number of services.

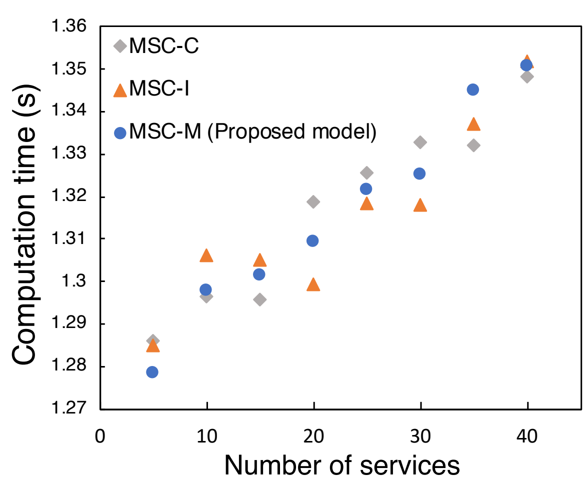


Figure 7: Computation time depending on number of services.

7. Directions for extending MSC-M to optical domain

The proposed model, MSC-M, provides the VNF placement and routes for SC requests which allow merging a part of SC paths. As mentioned in Section 1, there have been a number of studies which consider provisioning SC in optical networks. Optical networks facilitate providing muticast service chains which require high transmission capacity. Since the capacity granularity of optical networks is large [21], like Gbps or Tbps order, we should consider how to map SC paths obtained by solving the MSC-M model into the optical network in order to efficiently use spectrum resources.

In the optical network, optical paths are established between network nodes which equip optical transceivers. A certain amount of spectrum resource, such as a wavelength in the wavelength division multiplexing (WDM) channel, is assigned to each optical path. Optical paths are switched by using optical cross connects (OXCs) or reconfigurable optical add/drop multiplexers (ROADMs).

In the following discussions, we assume an optical network which consists of OXCs and ROADMs. Some ROADMs are locally connected with servers that can hold VNFs. When SC is provided in the optical network, traffic of service chains transferred as optical signals needs to be converted into electric signals at ROADMs where the traffic is processed by using VNFs. This means that each SC path can be mapped into the optical network by using several optical paths, each of which is established between ROADMs where consecutive VNFs are placed, between ROADMs of the source node and of the first VNF, or between ROADMs of the last VNF and of the destination node.

In the proposed model, bandwidth demands of each group are given. If the bandwidth demand of a group is close to the capacity of an optical path, the optical path can be directly assigned to each section of an SC path or a merged SC path belonging to the group. If the bandwidth demand of a group is much smaller than the capacity of an optical path, the capacity of optical path cannot be utilized efficiently. The traffic grooming technique [22] which agglomerates multiple small traffic flows into a single optical path can be utilized to improve the utilization efficiency of optical path. If the section of an SC path or a merged SC path includes one or more branches, an optical path can be divided in the optical or the electrical domain at the OXC/ROADM corresponding to the branching node.

8. Conclusion

In this paper, we proposed the MSC-M model, which is the VNF placement and routing model for multicast SC that allows merging SC paths of different services. We formulated the MSC-M model as an ILP problem. We proved that the decision version of VNF placement and routing problem based on the MSC-M model is NP-complete. A heuristic algorithm for the MSC-M model was introduced for the case that the ILP problem is intractable. The numerical results showed that the MSC-M model reduces the total cost to provide multiple service paths compared with the USC, MSC-C, and MSC-I models. The results indicated that determining the VNF placement and routing for all SC requests at a time so that SC paths of different services are merged is efficient to reduce the total cost. We discussed directions for extending the MSC-M model to an optical domain.

References

[1] Y. Li, M. Chen, Software-defined network function virtualization: A survey, IEEE Access 3 (2015) 2542–2553. doi:10.1109/ACCESS.2015.2499271.

[2] T. Lin, Z. Zhou, M. Tornatore, B. Mukherjee, Optimal network function virtualization realizing end-to-end requests, in: 2015 IEEE Global Communications Conference (GLOBECOM), 2015, pp. 1–6. doi:10.1109/GLOCOM.2015.7417447.

[3] V. Nikam, J. Gross, A. Rostami, VNF service chaining in optical data center networks, in: 2017 IEEE Conference on Network Function Virtualization

and Software Defined Networks (NFV-SDN), 2017, pp. 1–7. doi:10.1109/NFV-SDN.2017.8169845.

[4] A. Muhammad, N. Skorin-Kapov, M. Furdek, Manycast, anycast, and replica placement in optical inter-datacenter networks, IEEE/OSA Journal of Optical Communications and Networking 9 (12) (2017) 1161–1171. doi:10.1364/JOCN.9.001161.

[5] M. Xia, M. Shirazipour, Y. Zhang, H. Green, A. Takacs, Network function placement for NFV chaining in packet/optical datacenters, Journal of Lightwave Technology 33 (8) (2015) 1565–1570. doi:10.1109/JLT.2015.2388585.

[6] Q. Chen, B. Yang, D. Zhang, Q. Zhang, J. Chen, Flexible service chain mapping in server-centric optical datacenter networks, in: 2018 Asia Communications and Photonics Conference (ACP), 2018, pp. 1–3. doi:10.1109/ACP.2018.8596251.

[7] Z. Allybokus, N. Perrot, J. Leguay, L. Maggi, E. Gourdin, Virtual function placement for service chaining with partial orders and anti-affinity rules, Networks 71 (2) (2018) 97–106. doi:10.1002/net.21768.
URL https://onlinelibrary.wiley.com/doi/abs/10.1002/net.21768

[8] S. Q. Zhang, A. Tizghadam, B. Park, H. Bannazadeh, A. Leon-Garcia, Joint NFV placement and routing for multicast service on SDN, in: NOMS 2016 - 2016 IEEE/IFIP Network Operations and Management Symposium, 2016, pp. 333–341. doi:10.1109/NOMS.2016.7502829.

[9] B. Yi, X. Wang, M. Huang, A. Dong, A Multi-Stage solution for NFV-Enabled multicast over the hybrid infrastructure, IEEE Communications Letters 21 (9) (2017) 2061–2064. doi:10.1109/LCOMM.2017.2715808.

[10] Y. Cheng, L. Yang, VNF deployment and routing for NFV-enabled multicast: A steiner tree-based approach, in: 2017 9th International Conference

on Wireless Communications and Signal Processing (WCSP), 2017, pp. 1–4. doi:10.1109/WCSP.2017.8170947.

[11] O. Alhussein, P. T. Do, J. Li, Q. Ye, W. Shi, W. Zhuang, X. Shen, X. Li, J. Rao, Joint VNF placement and multicast traffic routing in 5G core networks, in: 2018 IEEE Global Communications Conference (GLOBECOM), 2018, pp. 1–6. doi:10.1109/GLOCOM.2018.8648029.

[12] B. Ren, D. Guo, Y. Shen, G. Tang, X. Lin, Embedding service function tree with minimum cost for NFV-Enabled multicast, IEEE Journal on Selected Areas in Communications 37 (5) (2019) 1085–1097. doi:10.1109/JSAC.2019.2906764.

[13] E. Guler, S. Devaraju, G. Luo, L. Tian, X. Cao, Multicast-aware service function tree embedding, in: 2019 IEEE 20th International Conference on High Performance Switching and Routing (HPSR), 2019, pp. 1–6. doi:10.1109/HPSR.2019.8807990.

[14] Y. Zhou, G.-S. Poo, Optical multicast over wavelength-routed WDM networks: A survey, Optical Switching and Networking 2 (3) (2005) 176 – 197. doi:https://doi.org/10.1016/j.osn.2005.10.001.
URL http://www.sciencedirect.com/science/article/pii/S1573427705000354

[15] X. Jia, D. Du, X. Hu, M. Lee, J. Gu, Optimization of wavelength assignment for QoS multicast in WDM networks, IEEE Transactions on Communications 49 (2) (2001) 341–350. doi:10.1109/26.905896.

[16] N. Kiji, T. Sato, R. Shinkuma, E. Oki, Virtual network function placement and routing model for multicast service chaining based on merging multiple service paths, in: 2019 IEEE 20th International Conference on High Performance Switching and Routing (HPSR), 2019, pp. 1–6. doi:10.1109/HPSR.2019.8807998.

[17] J. G. Herrera, J. F. Botero, Resource allocation in NFV: A comprehensive survey, IEEE Transactions on Network and Service Management 13 (3) (2016) 518–532. doi:10.1109/TNSM.2016.2598420.

[18] M. R. Garey, D. S. Johnson, Computers and Intractability; A Guide to the Theory of NP-Completeness, W. H. Freeman & Co., New York, NY, USA, 1990.

[19] J. Kleinberg, E. Tardos, Algorithm Design, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2005.

[20] IBM ILOG CPLEX optimization studio, https://www.ibm.com/products/ilog-cplex-optimization-studio.

[21] E. Oki, N. Wadaand S. Okamoto, N. Yamanaka, K. Sato, Optical networking paradigm: Past, recent trends and future directions, IEICE Transactions on Communications E100.B (9) (2017) 1564–1580. doi:10.1587/transcom.2016PFI0004.

[22] K. Zhu, B. Mukherjee, Traffic grooming in an optical WDM mesh network, IEEE Journal on Selected Areas in Communications 20 (1) (2002) 122–133. doi:10.1109/49.974667.