IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 1 March 2024, accepted 6 April 2024, date of publication 10 April 2024, date of current version 22 April 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3387083

RESEARCH ARTICLE

Security-Aware Service Function Chaining and Embedding With Asymmetric Dedicated Protection

BEN WANG[1], JUN LI[1], SHAOHUA CAO[2], (Member, IEEE), EVRIM GULER[3], AND DANYANG ZHENG[4], (Member, IEEE)

[1]School of Electronic and Information Engineering, Soochow University, Suzhou 215021, China
[2]College of Computer Science and Technology, China University of Petroleum (East China), Qingdao 266580, China
[3]Department of Computer Engineering, Bartin University, 74100 Bartın, Turkey
[4]School of Computing and Artificial Intelligence, Southwest Jiaotong University, Chengdu 611756, China

Corresponding author: Jun Li (ljun@suda.edu.cn)

This work was supported in part by the National Natural Science Founding of China under Grant 62101372, in part by the Leading Youth Talents of Innovation and Entrepreneurship of Gusu under Grant ZXL2023162, in part by the Open Fund of State Key Laboratory of Information Photonics and Optical Communications (IPOC) [Beijing University of Post Technology (BUPT)] under Grant IPOC2022A07, and in part by the State Key Laboratory of Advanced Optical Communication Systems and Networks under Grant 2023GZKF11.

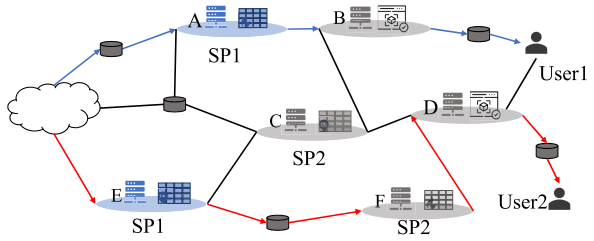
ABSTRACT In the 5G and beyond 5G networks, achieving security-aware data transmission needs to convert clients' requests into a service function chain (SFC), each service function (SF) providing a certain security guarantee. With diverse configuration techniques, an SF may own multiple versions, each version providing various security guarantees with diverse costs. It should be notice that, as the recent software failures have caused severe financial loss, great attentions from both academia and industry have been put onto the SFC reliability. In the literature, existing works have solely investigated the following two fields: 1) how to deploy a security-aware SFC, and 2) how to protect a traditional SFC. Simply applying these techniques to dealing with the problem of security-aware SFC protection might not be efficient as the backup and primary SFCs may not be identical for security-aware SFCs. Therefore, how to jointly take these fields into account is challenging and remains open. To tackle the above problem, this paper studies how to construct and embed a security-aware SFC with asymmetric dedicated protection. We mathematically define this problem and name it security-aware service function chaining, embedding, and protection with multi-versioned SFs (SFCEP-MF) with the objective of cost optimization. Next, to optimize the SFCEP-MF problem, we construct an efficient algorithm, called augmenting-path with primary-first disjoint SFP identifier (APPF-DSI). Extensive simulation results show that the APPF-DSI algorithm outperforms the benchmark approaches that are directly extended from the state-of-the-art.

INDEX TERMS Network function virtualization, service function with multi-versions, service function chaining and embedding, security, reliability.

I. INTRODUCTION

The increasing number of 5G infrastructures and Internet of Things (IoT) devices demand a significant number of network services [1]. Traditionally, these services employ network functions that are hosted by middleboxes to enhance the network service performance [2]. As these middleboxes are generally implemented by hardware-based, it is costly and inefficient for maintenance and management [3]. To reduce the capital expenditure (CAPEX) and operation expense (OPEX), network function virtualization (NFV) is introduced to implement network functions by software-based modules, called service functions (SFs) [2], [4]. In the NFV paradigm, a network service generally includes a set of SFs and the corresponding networking resource demands for initializing the SFs [5]. To accommodate an NFV-based network service,

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



SP \ SF	Firewall		DPI		Rendering
SP1	Security:6	Cost:11	Security:8	Cost:18	Cost:15
SP2	Security:7	Cost:15	Security:10	Cost:20	

FIGURE 1. Sec-SFC with multi-versioned SFs.

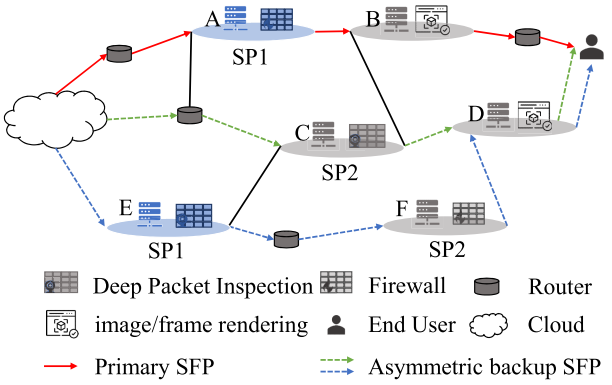


FIGURE 2. Sec-SFC with asymmetric dedicated protection.

the service provider (SP) concatenates the required SFs into a chain structure called service function chain (SFC) and deploys it onto a shared physical network (PN) [6], [7]. The process of composing and embedding an SFC over a shared PN is SF chaining and embedding (SFCE) [8], [9]. After the SFCE process, the SF is initialized as an SF instance (SFI) and hosted by a physical node (e.g., server), and the SFC is hosted by a physical path called service function path (SFP).

Recently, due to the frequent cyber-attack issues [10], the SFC security attracts great attentions from both industry and academia. To defend against various network attacks security-aware SFCs (Sec-SFC) are employed [11], [12]. To enhance the SFC security, diverse security-aware SFs are employed [13], [14]. For example, firewalls are utilized to control network access, network intrusion detection systems (NIDS) are employed to monitor exploitation attacks in network loads, and network anomaly detection systems are used to detect distributed denial of service (DDoS) attacks [14]. For an SF, different service providers implement it in different ways, resulting in different cost and security performances of the SF, one security-aware SF may own multiple versions [13]. At the meantime, these security-aware network services are delivered via the "pay-as-you-go" pattern, and the clients can customize how much safety they need during their data transmissions [15]. Here, the "safety they need" is referred to as the **security requirement** [16]. In this paper, we use a certain number to represent the user's security requirements. The larger the number, the higher the user's security requirements. Due to the various configuration techniques, one SF can be initialized in various versions, each of which will provide dissimilar securities and need diverse implementation costs [17], [18]. To flexibly meet the security requirement of the client, the orchestrator can employ security-aware SFs from different SPs to formulate the SFC, and such an SFC is referred to as the security-aware SFC (Sec-SFC). For example, Fig. 1 shows a cloud virtual reality (VR) service that is satisfied by security-aware SFs with multiple versions. Here, *User1* and *User2* need the security requirements of 5 and 15, respectively. Meanwhile, the security and the cost of implementing firewall and DPI from SP1 and SP2 are respectively shown in the table of Fig. 1. To reduce the cost and satisfy the security requirement, the service of *User1* could employ the functions provided by SP1 via the path of $cloud \rightarrow A(firewall) \rightarrow B(image/framerendering) \rightarrow User1$ with the cost of 26, and *User*2 could employ the functions provided by SP2 via the path of $cloud \rightarrow E(DPI) \rightarrow F(firewall) \rightarrow D(image/framerendering) \rightarrow User2$ with the cost of 53.

In NFV-based networks, it is very important to consider reliability because the failure of any SF in the SFC, whether due to the hardware issue (e.g., a failure of the physical machine hosting the virtual machine that is executing that SF) or the software issue (e.g., the SF itself or the virtual machine executing it is incorrectly configured), will break the entire chain and will result in a suspension of service. Great attention has been put on investigating the SFC reliability issue, and an effective way to improve SFC reliability is through the adoption of redundancy-based SF placement policies. Here, we protect the working SFC by placing a backup SFC in advance to improve the reliability of the service, thus preventing not only the failure of the SF but also the link failure in the network service chain. Even though extensive efforts have been put into SFC protection, no existing work takes the protection of the Sec-SFC into account, and it is non-trivial to directly employ the existing work on handling the Sec-SFC protection. Different from the traditional SFC protection, the purpose of the Sec-SFC protection is to satisfy the security requirement from clients [19]; thus, the backup SFC does not necessarily require a symmetric structure as the primary one does. We employ Fig. 2 to address this. As shown in Fig. 2, a VR service routing the forward path $cloud \rightarrow A(firewall) \rightarrow B(image/framerendering) \rightarrow EndUser$ that use firewall to enhance security, and an asymmetric dedicated protection approach is employed to protect the service. There exists more than one available asymmetric backup SFCs. The green forward path $cloud \rightarrow C(DPI') \rightarrow D(image/framerendering') \rightarrow EndUser$ is an alternative backup SFP, which employs DPI to replace the firewall in the primary SFC. Meanwhile, the blue forward path $cloud \rightarrow$

$E(DPI') \rightarrow F(firewall) \rightarrow D(image/framerendering') \rightarrow EndUser$ shows another backup SFP, which uses firewall and DPI to protects.

Based upon, Fig. 1 and Fig. 2 have introduced the concept of security requirement and multi-versioned SF. As one can see, the joint process of security-aware SF chaining, embedding and protection is distinguished from the traditional SFC embedding process and is even more challenging since one must select the proper set of security-aware SFs to meet the security guarantee requirement. As proved by [20], the cost optimization problem of SFC embedding is already NP-hard. How to resolve the optimization problem of security-aware SF chaining, embedding, and protection remains open and challenging. To tackle the above problem, this work comprehensively studies the problem of jointly chaining, embedding and protecting a Sec-SFC onto a shared PN. First, we mathematically define this problem and name it security-aware service function chaining, embedding, and protecting with multi-versioned SFs (SFCEP-MF) and we prove this problem is NP-hard. Next, to solve this problem, we propose an asymmetric dedicated protection method. Based on this method, we propose a novel technique called augmenting-path-based disjoint SFP identifier (AP-DSI). We analyze the feasibility of AP-DSI and identify the circumstances that AP-DSI does not work well. For improvement, we propose another technique called primary-first disjoint SFP identifier (PF-DSI) to enhance the performance in these circumstances. Then, we combined these two techniques to form our algorithm called Augmenting-Path with Primary-First Disjoint SFP Identifier (APPF-DSI). Last but not least, compared to the schemes directly extended from the existing works [21], [22], extensive simulations show APPF-DSI can significantly reduce the implementation cost.

The rest of this paper is organized as follows. Section II introduces the asymmetric dedicated protection concept. Section III summarizes the related work. We formulate the security-aware service function chaining, embedding, and protection with multi-versioned SFs (SFCEP-MF) problem in Section IV. Section V presents the service function chaining, embedding and protecting algorithm, and Section VI analyzes the experimental results. At last, Section VII concludes the paper.

II. ASYMMETRIC DEDICATED PROTECTION

In this section, we summarize the traditional symmetric SFC dedicated protection concept and propose our novel asymmetric SFC dedicated protection concept.

A. SYMMETRIC SFC DEDICATED PROTECTION

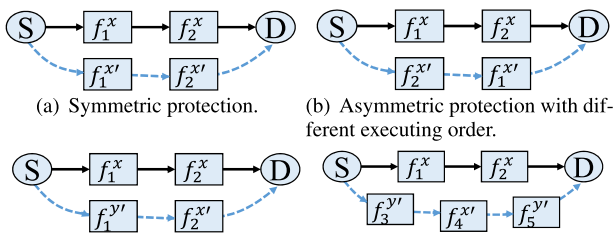
In the traditional symmetric SFC dedicated protection, the backup SFC employs an identical structure as the primary SFC does. Specifically, the backup and primary SFCs have the same source, destination, SF types, and SF executing orders. For example, Fig. 3(a) shows a pair of primary and backup SFs with the symmetric SFC dedicated protection. It is worth noting that, to ensure that the backup SFP could function properly when failures occur at the primary SFP, the embedding process of the primary SFC and the backup SFC must be "physically-disjoint". Here, the "physically-disjointedness" refers to that the primary and backup SFPs should employ different physical nodes for hosting SFIs and diverse physical links for data transmission.

Different from the traditional SFC delivery, the client in the scenario with Sec-SFCs does not directly require a set of SFs but a specified security requirement. In this case, employing the SFC dedicated protection will largely decrease the number of possible primary and backup SFC pairs. This is because the primary and backup SFC pair only needs to satisfy the client's security requirement and they can employ asymmetric SFC structure. When an SFI fails, the service flow will be switched from the the failure-impacted SFC to the backup one, and the destination will notify the source which data are lost due to the failed SFI, rather than directly grabbing the data from the faulty SF. Therefore, even if the primary and backup SFC employed different SFs, the security requirement of the client is guaranteed. Due to this, directly employing the symmetric SFC dedicated protection for Sec-SFCs may exclude the best primary and backup Sec-SFC pair and potentially increase the implementation cost. To further tackle this point, we formally define the concept of asymmetric SFC dedicated protection.

B. ASYMMETRIC SFC DEDICATED PROTECTION

To facilitate the protection of the Sec-SFC, we formally define the concept of asymmetric SFC dedicated protection. Asymmetric SFC dedicated protection means that the pair of primary and backup SFCs does not need to (i) follow the same SF executing order, (ii) employ the same set of SFs, but must (iii) satisfy the client's security requirement. We use Figs. 3 for further elaboration.

Fig. 3(a) shows the symmetric SFC dedicated protection scheme, where the primary and backup SFCs have identical structures. Fig. 3b shows an example of asymmetric SFC dedicated protection, where the executing order of the backup SFC is different from the primary one. In specific, the primary SFC employs $f_1^{x'}$ and $f_2^{x'}$ in the order of $f_1^x \rightarrow f_2^x$ but they are in the order of $f_2^{x'} \rightarrow f_1^{x'}$ in the backup SFC. Fig. 3c shows another example of asymmetric SFC dedicated protection, where different versions are taken into account.



(a) Symmetric protection. (b) Asymmetric protection with different executing order.

(c) Asymmetric protection with different SF version. (d) Asymmetric protection with different SF types.

FIGURE 3. Examples of asymmetric SFC protection.

TABLE 1. Summary of related work.

Paper	SFC embedding	SF chaining	SFC protection	security
[23]–[25]	✓			
[5], [7], [9], [20]	✓	✓		
[21], [26], [27]	✓		✓	
[28]	✓	✓	✓	
[16], [22], [29]–[31]	✓			✓
This work	✓	✓	✓	✓

In particular, the backup SFC employs $f_1^{y'}$ and $f_2^{x'}$ to protect the primary SFC that employs $f_1^{x'}$ and $f_2^{x'}$. Last but not least, Fig. 3d demonstrates an example of asymmetric SFC dedicated protection, where the primary and backup SFCs employ different types of SFs to meet the client's security requirements. That is, the backup SFC use $f_3^{y'}$, $f_4^{x'}$, and $f_5^{y'}$ instead of $f_1^{x'}$ and $f_2^{x'}$ in the primary SFC for protection.

As one can see from the above examples, the symmetric SFC protection is a sub-case of the asymmetric SFC protection, where the primary and backup SFCs employ the same set of SFs with the same executing order. Since the symmetric protection is a sub-case of the asymmetric one, directly employing existing symmetric SFC protection approaches may not be efficient for the asymmetric case. To find an efficient way for tackling the asymmetric SFC dedicated protection, in the following contents, we summarize the related work, formulate a novel problem called security-aware service function chaining, embedding, and protection with multi-versioned SFs (SFCEP-MF), and propose efficient techniques and heuristic algorithms.

III. RELATED WORK

In this section, we summarize the existing work related to the SFCE process. We classify these works according to whether they have taken (i) SF chaining, (ii) SFC embedding, (iii) SFC protection, and (iv) security into account, which is shown in table 1.

A. SFC EMBEDDING

When given an NFV service request, the embedding of SFC received a lot of attention in the literature [23], [24], [25]. The author of [23] have investigated the problem of dependence-aware service function chain embedding in Optical networks, and they proposed an efficient algorithm, namely, dependence-aware service function chain embedding with least-used consecutive subcarriers algorithm to solve this problem. The author of [24] have defined a new problem called multicast services embedding in optical networks with fanout limitation, and they proposed an efficient algorithm, namely, centrality-based degree bounded shortest path tree (C-DB-SPT) algorithm, to take advantage of the centrality technique for multicasting node and link mapping while minimizing the needed resource and satisfying the fanout limitation. The authors of [25] have studied the problem of dependence-aware SFC embedding, and they have proposed an efficient algorithm, namely, dependence-aware SFC embedding with group embedding, to efficiently map clients' service requests over the physical network while taking into consideration the computing resource demand, function dependence of the VNFs, and the bandwidth demand for the service request.

B. SF CHAINING AND EMBEDDING

Great efforts have been put into investigating how to efficiently compose and embed an SFC over a shared PN [5], [7], [9], [20]. The work of [9] defined the concept of hybrid SFC (h-SFC), whereas some SF nodes are required to process bidirectional traffic while others only handle unidirectional traffic, and proposed efficient approximate algorithms to solve the h-SFC composing and embedding problem. Furthermore, the authors of [5] comprehensively studied how to optimize the latency of hybrid SFC composing and embedding process, and proposed the first 2-approximation algorithm by utilizing the feature of Eulerian circuit. When an h-SFC is given a priori, the authors of [7] studied the problem of optimally embedding a hybrid SFC over a shared multi-access edge computing system, and the proposed Opt-HSFCE algorithm needs much less runtime compared with the brutal force algorithm, and significantly outperforms the schemes that are directly extended from the existing techniques. The authors in [20] comprehensively investigate how to minimize the cost when delivering network services as SFCs with provable bounds and fewer assumptions and formally define the problem of minimum cost service function chaining and embedding (MC-SFCE) and propose an algorithm, namely, cost factor-based SFCE optimization with shortcut (COFO-SC), for MC-SFCE. they provided novel mathematical analysis to demonstrate the correctness of the approaches and related bounds.

C. SFC PROTECTION

In order to ensure the reliability of SFC, the protection of SFC has also been studied extensively [21], [26], [27], [28]. The authors of [21] present novel survivability provisioning heuristics for SFC embedding in multi-layer fog networks, they use pre-provisioned backup node resources and link bandwidth to achieve rapid switchovers for single node/link failures. The authors of [26] proposed for the first time a multi-path based disaster protection scheme for SFC embedding, and they formulated this problem by a layered-flow based ILP so as to find the protection solution with the minimum network cost in terms of bandwidth and processing resource. The authors of [27] jointly considered physical/virtual network failures and hardware/software

failures, and they first introduce an augmented SF protection graph, called k-connected service function slicing (KC-SFS), which can facilitate the SF protection against multiple concurrent physical/virtual node and physical link failures. Based on the KC-SFS, they propose an efficient heuristic algorithm, called service function slice embedding (SFSE), which employs the k-connected network slicing technique (k-NST) to solve the deterministic fault-tolerant service function slicing (DFT-SFC) problem. Via thorough mathematical analysis, they prove that k-NST achieves 2-approximation. The authors of [28] proposed an instance-sharing and reliable construction algorithm (ISRCA) to aggregate multiple SFCs into a service function graph (SFG), and perform reliability screening for the SFG set. After mapping the SFG to the physical network, a node-ranking algorithm with centrality and reliability (NRCR) is proposed for backup node selection and backup instance implementation to improve the reliability of SFCs that have not met the requirements.

D. SECURITY

Recently, the security-aware SFC has attracted great attention from academic groups [16], [22], [29], [30], [31]. The authors of [29] mainly studied the implementation of SFC and proposed two algorithms that can guarantee privacy and security, and optimize the deployment time, respectively. The authors of [16] have introduced a security level to indicate standard protection, whereas 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, thereby, a substrate node with a higher security level has a higher level protection mechanism for embedding virtual nodes. The authors of [30] have also considered the construction of virtual networks based on the security level for network security, whereas security VNFs are added to some substrate nodes for increasing those security levels such that the security level of each virtual network is satisfied. Meanwhile, the authors of [31] proposed a flexible and configurable dynamic composition mechanism of the security service function chain, and this mechanism establishes a combined model based on vector space and integer programming, which reduces the transmission delay but increases the new reconstruction operation time overhead. The authors of [22] first proposed security constraints based on physical isolation and formulated the problem of reasonably placing multi-tenant SFCs subject to multiple constraints to minimize resource consumption, and they proposed a hypergraph matching algorithm to find the maximum weight subset of hypergraphs whose vertices do not intersect to give a high degree of physical isolation by dealing with the conflict graphs.

As one can see, no existing work jointly took the above four fields into account. That is, one cannot simply employ the existing approaches to solve the SFCEP-MF problem. To begin, we formulate the mathematical model for the SFCEP-MF problem as follows.

IV. PROBLEM STATEMENT

In this section, we formulate the problem of security-aware service function chaining, embedding, and protection with multi-versioned SFs (SFCEP-MF).

A. PHYSICAL/SUBSTRATE NETWORK MODEL

We use an undirected graph $G = (N, L, F)$ to represent the physical/substrate network, where N, L and F are the set of physical nodes, physical links and service functions, respectively. An SF has multiple SPs, and for different SPs, it requires different costs and provides different security performance. For each physical node $n \in N$, it has a total amount of computing resources (C_n) and can host a specific set of SFs. For each link $l \in L$, it has a certain amount of bandwidth resource (BW_l). To facilitate presentation, we use $l_{a,b}$, where a and b ($\forall a, b \in N$) are endpoints of this link.

B. NETWORK SERVICE REQUEST MODEL

A network service request (NSR) is denoted by a 4-tuple $NSR =< s', d', bw, \Theta >$, where s' and d' respectively represents the source and destination, bw represents bandwidth requirements, and Θ is client's security requirements. We use $Sec_{f_i^x}$ to represent the security provided by f_i^x.

C. SECURITY-AWARE SERVICE FUNCTION CHAINING, EMBEDDING, AND PROTECTION WITH MULTI-VERSIONED SFS (SFCEP-MF)

Given an NSR and a physical network, we investigate the problem of how to compose, embed and protect a security-aware SFC that meets the security requirement and has the minimum overall cost. Table 2 lists the notation used in the problem formulation. The objective of the proposed problem is to minimize the primary and backup SFC cost (i.e., $\mathbb{C}^P$, $\mathbb{C}^B$):

$$min\ \mathbb{C}^P + \mathbb{C}^B \tag{1}$$

The sum of the computation and bandwidth cost is denoted by $\mathbb{C}_{CPU}$ and $\mathbb{C}_{BW}$, respectively. Eq. (2) and Eq. (3) elaborate on the calculation of the cost for creating the primary/backup SFP, respectively.

$$\mathbb{C}^P = \mathbb{C}^P_{CPU} + \mathbb{C}^P_{BW} \tag{2}$$

$$\mathbb{C}^B = \mathbb{C}^B_{CPU} + \mathbb{C}^B_{BW} \tag{3}$$

1) SF NODE EMBEDDING CONSTRAINT

In Eq. (4), we use $M_n^{i/x}(M_n^{i/x'})$ to represent whether an SF (i.e., f_i^x or $f_i^{x'}$) is mapped onto physical node n or not, where i and x respectively represent the type and SP of an SF f. Meanwhile, i/x (i/x') represents an SF f_i^x ($f_i^{x'}$) in primary SFC (pSFC) or backup SFC (bSFC). Eq. (5) ensures that every required SF must be hosted by a physical node. Eq. (6) shows that SFs in $\mathbb{R}_f$ cannot be mapped onto the same node. Eq. (7) requires that a node n cannot be simultaneously employed in both primary and backup SFCs.

TABLE 2. Notation Table for Problem Formulation.

Notation	Definition
N	Set of physical nodes.
L	Set of physical links.
F	Set of service functions in the network.
S	Set of service providers in the network.
f_i^x	The i^{th} SF in the SF set that provided by SP x.
$\mathbb{R}_f$	Service functions that cannot be placed on the same node.
D_{f_i,f_j}	Service functional dependency matrix (element = 0, f_i cannot be placed before f_j, = 1 can be placed before f_j).
c_f	Computing resources required by service functions.
C_n	Total computing resources on a node $n, \forall n \in N$.
bw	Bandwidth demand.
Θ	Security requirement.
$Sec_{f_i^x}$	The security that provide by f_i^x.
$BW_{l_{a,b}}$	Bandwidth capacity of link $l_{a,b}$.
$M_n^{i/x}$	binary variable, represent whether an SF is mapped onto physical node n or not.
$p_{m,n}^{i/x,j/y}$	binary variable, represent whether or not the physical path from m to n is used to complete the SFP.
$l_{a,b}$	binary variable, represent whether or not the physical link from a to b is used to complete the SFP.
δ	The cost required to consume a unit of computing resources.
γ	The cost required to consume a unit of bandwidth resources.
α_n	Cost scale factor of node n.

Eq. (8) guarantees that node n must have enough computing resources to host the SFs.

$$M_n^{i/x} = \begin{cases} 1, & f_i^x \in \{\text{pSFC, bSFC}\} \text{ is mapped onto } n; \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

$$\sum_{n \in N} M_n^{i/x} = 1, \forall f_i^x \in \{\text{pSFC, bSFC}\} \tag{5}$$

$$M_n^{i/x} + M_n^{j/y} \leq 1, \forall f_i^x, f_j^y \in \mathbb{R}_f \tag{6}$$

$$M_n^{i/x} + M_n^{j/y'} \leq 1, \forall n \in N \tag{7}$$

$$\sum_{f_i \in F} \sum_{x \in S} c_{f_i^x} * M_n^{i/x} \leq C_n, \forall n \in N \tag{8}$$

2) SFP ROUTING CONSTRAINT

In Eq. (9), we use $p_{m,n}^{i/x,j/y}$ ($p_{m,n}^{i/x',j/y'}$) to represent whether or not the physical path from m to n is used to complete the primary SFP (pSFP) and backup SFP (bSFP) or not. Eq. (10) ensures that a type of SF f^x will only be used once in pSFC (bSFC). Eq. (11) represents whether the link $l_{a,b}(l'_{a,b})$ is a sub-path of pSFP (bSFP) or not. Eq. (12) ensures that a link $l_{a,b}$ cannot be employed in both pSFP and bSFP. Eq. (13) shows the bandwidth constraint.

$$p_{m,n}^{i/x,j/y} = \begin{cases} 1, & \text{path from } m \text{ to } n \text{ that host} \\ & f_i^x, f_j^y \in \{\text{pSFC, bSFC}\} \\ 0, & \text{otherwise} \end{cases} \tag{9}$$

$$p_{m,n}^{i/x,j/x} = 0, \forall i, j \in S \tag{10}$$

$$l_{a,b} = \begin{cases} 1, & \text{is a sub-path of } p_{m,n} \\ 0, & \text{otherwise} \end{cases}, \forall a, b \in N, a \neq b \tag{11}$$

$$l_{a,b} + l'_{a,b} \leq 1, \forall a, b \in N \tag{12}$$

$$\sum_{f_i, f_j \in F} \sum_{x,y \in S} l_{a,b} * p^{i/x,j/y} * bw \leq BW_{l_{a,b}}, \forall a, b \in N \tag{13}$$

3) SEC-SFC CHAINING CONSTRAINT:

Eq. (14) ensures that the pSFC (bSFC) satisfies the client's security requirement. Eq. (15) guarantees that the SFs f_i and f_j meet the placement sequence requirements, where D_{f_i,f_j} is a binary value (=1, f_i should be placed before f_j; =0, otherwise). Eqs. (16) and (17) demonstrate the computation and bandwidth cost.

$$\sum_{f_i \in F} \sum_{x \in S} \sum_{n \in N} Sec_{f_i^x} * M_n^{i/x} \geq \Theta \tag{14}$$

$$p_{m,n}^{i,j} * \frac{M_m^i + M_n^j}{2} \leq D_{f_i,f_j} \tag{15}$$

$$\mathbb{C}_{CPU} = \sum_{f_i \in F} \sum_{x \in S} \sum_{n \in N} c_{f_i^x} * M_n^{i/x} * \delta * \alpha \tag{16}$$

$$\mathbb{C}_{BW} = \sum_{f_i, f_j \in F} \sum_{x,y \in S} \sum_{a,b \in N} l_{a,b} * p^{i/x,j/y} * bw * \gamma \tag{17}$$

With limited networking resources, the SF chaining and embedding problem is proved to be NP-hard [20]. In our problem, we not only need compose and embed a security-aware SFC, but also need to protect it in networks with multi-versioned SFs. Accordingly, the problem of SFCEP-MF is NP-hard as well.

V. SECURITY-AWARE SERVICE FUNCTION CHAIN PROTECTION AND DEPLOYMENT

To tackle the SFCEP-MF problem, we propose novel techniques of individual SFP security maintenance (ISSM), augmenting-path-based disjoint SFP identifier (AP-DSI), and primary-first disjoint SFP identifier (PF-DSI). By combing the above techniques, we then propose the augmenting-path with primary-first disjoint SFP identifier (APPF-DSI) algorithm.

A. INDIVIDUAL SFP SECURITY MAINTENANCE (ISSM)

Constructing security-aware SFCs needs to take into account all types SFs in all versions. In practice, all physical nodes cannot be the candidates of a specified SF. Meanwhile, the candidate nodes of an NSR varies during the SFC chaining process as one SFC can at most includes one specified type of SF regardless of which version it is.

Algorithm 1 Individual SFP Security Maintenance

1: **Input:** G, SFP_b, $SFC_p(SFP_p, SFC_b)$;
2: **Output:** Primary/Backup SFI candidate graph;
3: remove the node used in SFP_b/SFP_p from the PN;
4: **if** $SFI \cap SFC_p/SFC_b \neq \emptyset$ **then**
5: Remove this SFI from the PN;
6: **end if**
7: **Return** Primary/Backup SFI candidate graph;

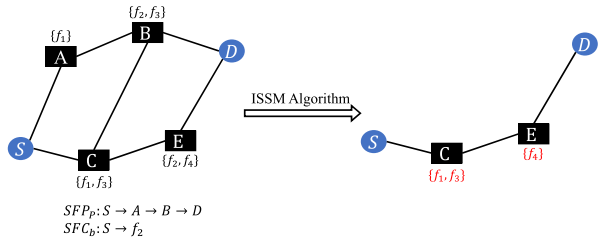


FIGURE 4. Generating a backup SFI candidate graph.

Notably, the selection of candidate nodes will be affected by the disjointedness constraint as well. Based on the above discussion, to efficiently identify the set of candidate nodes, we propose the individual SFP security maintenance technique as shown in Algorithm 1. The input of the algorithm is the network G, an SFC, and an SFP. Notably, to achieve the SF non-repetition and disjointedness, when SFC is the primary one, then the SFP is the backup one, vice versa. According to SFC (either the SFC_p or SFC_b), and the given SFP (either the SFP_p or SFP_b), the algorithm removes all physical nodes employed by the SFP and all SFIs that are already included by the SFC. At last, the algorithm returns a network within the available SF candidate for constructing either the primary flow or the backup flow. We call such a network primary/backup SFI candidate graph in terms of which flow (primary or backup) it supports. For example, we use Fig. 4 to show how to generate a backup SFI candidate graph. At first, according to the primary SFP ($S \to A \to B \to D$), we remove the nodes A and B used in the primary SFP from the network graph. Then, we remove SFs (f_2) that have been used by the backup Sec-SFC from the network graph. The remaining network graph is the backup SFI candidate graph. Here we assume the links have enough bandwidth that satisfy the bandwidth demand in NSR. Otherwise, there may no possible SFI candidate graph.

B. AUGMENTING-PATH-BASED DISJOINT SFP IDENTIFIER (AP-DSI)

Before introducing the augmented path technique, we first introduce Suurballe's method of finding the shortest disjoint path pairs [32]. The algorithm searched a pair of shortest disjoint paths through the following steps: 1) find a shortest path in the topology through Dijkstra's algorithm; 2) change the weight of each edge; then reverse the shortest path obtained in step 1 in the topology to obtain a new topology; 3) use Dijkstra's algorithm in the new topology, obtain a new path; 4) discard every edge in one path whose reversal appears in the other, and the remaining edges into two paths. Inspired by Suurballe's algorithm, we propose the technique of augmenting-path in SFP construction. This technique constructs the primary SFP and backup SFP through the following steps: 1) find the shortest path in the topology through Dijkstra's algorithm; 2) reverse the shortest path obtained in step 1 in the topology diagram, and change the edge weight after reverse to zero to obtain a new topology diagram; 3) apply Dijkstra's algorithm to obtain a path in the new topology; and 4) remove every edge in one path whose reversal appears in the other, and re-formulate the remaining edges into two disjoint paths. Steps 2 and 4 ensure that primary SFP and backup SFP are disjoint.

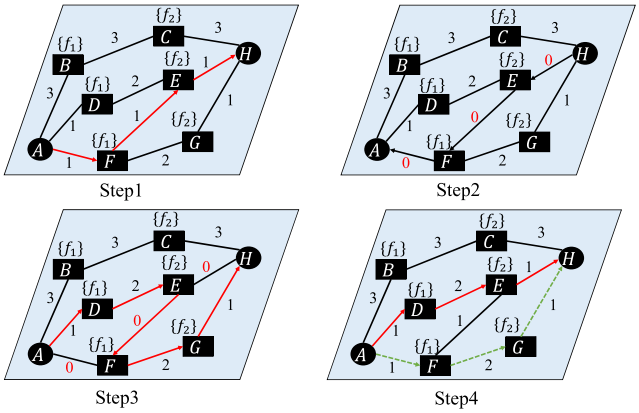


FIGURE 5. The steps in the technique of augmenting-path technique in SFP construction.

We use Fig. 5 to illustrate how the augmenting-path in SFP construction technique works. In Fig. 5, each node can host one SF instance, and the number beside a link represents the link weight or cost. We assume that the client's SFC request is $A \to f_1 \to f_2 \to H$. To begin, find a shortest path ($A \to F(f_1) \to E(f_2) \to H$) using dijkstra's algorithm. Then reverse the path ($A \leftarrow F(f_1) \leftarrow E(f_2) \leftarrow H$) and change the weight of these edges to zero. Next, use Dijkstra's algorithm to find a shortest path ($A \to D(f_1) \to E \to F \to G(f_2) \to H$) in the changed topology. Finally, remove edge ($E - F$) in one path whose reversal appears in the other, and grouping the remaining edges into two disjoint paths (primary SFP: $A \to D(f_1) \to E(f_2) \to H$, backup SFP: $A \to F(f_1) \to G(f_2) \to H$). Compared with deleting path ($A \to F(f_1) \to E(f_2) \to H$) directly from the topology and finding a new disjoint path ($A \to B(f_1) \to C(f_2) \to H$), the augmenting-path technique has better network availability.

Based on the above augmenting-path in SFP construction technique, we propose an augmenting-path-based disjoint SFP identifier (AP-DSI) algorithm. Since there probably exist multiple SFs in an SFC, the SFIs in these two paths formed after the path reversal and removing process may change. This may cause an SFC might fail the security requirement. To avoid this circumstance, we keep updating the starting point so that only one service function is included each time during the SFP creation process. The input s and s^* indicate the start point of the updated primary SFP and backup SFP respectively. Here, n indicates the node on primary SFP that hosts the required SFI. Line 3 removes the path SFP_p from the network graph and line 4 reverses the paths of s to n and n to d. To decrease the security-aware SFC implementation cost by jointly taking both the security and the embedding cost into account, we propose a factor called security-cost

Algorithm 2 Augmenting-Path-Based Disjoint SFP Identifier

1: **Input:** G, SFP_p, SFC_p, s, s^*, d, n;
2: **Output:** SFP_b, SFC_b;
3: remove the path SFP_p from the network graph;
4: reverse the paths of s to n and n to d;
5: **for** SFI in backup SFI candidate graph **do**
6: Find the proper SFI with the highest R_{sc};
7: **end for**
8: Remove the path that reversed in SFP_p and SFP_b;
9: Update SFP_p, SFC_p, SFP_b, SFC_b;
10: **Return** SFP_b, SFC_b;

Algorithm 3 Primary-First Disjoint SFP Identifier

1: **Input:** G, SFP_p, SFC_p, s, s^*, d, n;
2: **Output:** SFP_b, SFC_b;
3: remove the paths SFP_p, s to n and n to $d$from the network graph;
4: **for** SFI in backup SFI candidate graph **do**
5: Find the proper SFI with the highest R_{sc};
6: **end for**
7: Update SFP_b, SFC_b;
8: **Return** SFP_b, SFC_b;

center ratio (R_{sc}) as shown in Eq. (18). This factor represents the ratio of the security provided by an SFI on a node to the routing cost and the implementation cost. The higher the ratio is, the higher the security and the lower the routing and embedding cost can be provided by initializing an SFI on this node (a node has a high security-cost center ratio when it can provide SFI with high security and is close to both source and destination).

$$R_{sc} = \frac{Sec_{sf}}{\mathbb{C}_{sf} + \mathbb{C}_{BW}(s, n) + \mathbb{C}_{BW}(n, d)} \tag{18}$$

Then, lines 5-7 find the SFI with the least security-cost center ratio (R_{sc}). Next, the algorithm needs to remove the paths that are reversed in SFP_p and SFP_b and swap the sub-paths before and after the reversed path to ensure that the two paths are disjoint (lines 8-9). Finally, we add the new security SF to the SFC_b and get the updated SFP_b.

C. PRIMARY-FIRST DISJOINT SFP IDENTIFIER (PF-DSI)

Even though the AP-DSI technique can solve the problem of SFI position changing problems, it does not take into account the duplication of SFI in the already constructed primary security-aware SFC and the newly created SFP may violate the SF no-repetition constraint. Due to this, when paths need to be switched in the AP-DSI algorithm if the newly found SFI is repeated in the SFC_p, the algorithm may fail. To avoid this circumstance, we proposed primary-first disjoint SFP identifier (PF-DSI) algorithm. PF-DSI prevents reversing paths from the primary and backup paths by directly deleting paths SFP, s to n and n to d (line 3). Then, lines 4-7

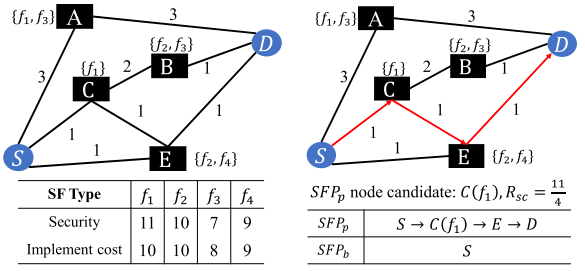
Algorithm 4 Augmenting-Path With Primary-First Disjoint SFP Identifier

1: **Input:** G, NSR;
2: **Output:** SFP_p, SFP_b, SFC_p, SFC_b;
3: Initialization;
4: **while** $Sec_{SFC_p} < \theta || Sec_{SFC_b} < \theta$ **do**
5: **if** $Sec_{SFC_p} < \theta$ **then**
6: Call ISSM get primary SFI candidate graph;
7: **for** SFI in primary SFI candidate graph **do**
8: Find the proper SFI with the highest R_{sc};
9: **end for**
10: **end if**
11: **if** $Sec_{SFC_b} < \theta$ **then**
12: Call ISSM get backup SFI candidate graph;
13: **for** SFI* in backup SFI candidate graph **do**
14: **if** SFI* $\cap SFC_p = \emptyset$ **then**
15: Call AP-DSI update SFP_b, SFC_b;
16: **else**
17: Call PF-DSI update SFP_b, SFC_b;
18: **end if**
19: **end for**
20: **end if**
21: Update SFP_p, SFC_p;
22: **end while**
23: **Return** SFP_p, SFP_b, SFC_p, SFC_b;

find the SFI with the least security-cost center ratio (R_{sc}) from the backup SFI candidate graph. Line 7 adds the found service function instance to the backup SFC SFC_b and updates the backup path SFP_b.

D. AUGMENTING-PATH WITH PRIMARY-FIRST DISJOINT SFP IDENTIFIER (APPF-DSI)

Last but not the least, the augmenting-path with primary-first disjoint SFP identifier (APPF-DSI) combines the above three algorithms to solve the SFCEP-MF problem. This algorithm simultaneously realizes the construction of the security-aware SFC, as well as the mapping and protection. Based on the input network topology G and NSR, APPF-DSI first initializes some variables, i.e., $SFP_p = \emptyset$, $SFP_b = \emptyset$, $SFC_p = \emptyset$, $SFC_b = \emptyset$, $s = s'$, $s^* = s'$ and $d = d'$. Then, APPF-DSI starts with a while loop, which does not end until the security of both the primary security-aware SFC (e.g. Sec_{SFC_p}) and the backup security-aware SFC (e.g. Sec_{SFC_b}) meets the user's security requirements. In this while loop, APPF-DSI first determines whether the security of SFC_p meets the requirements or not. If the requirements are not met, APPF-DSI will call ISSM algorithm to generate the primary SFI candidate graph, then, APPF-DSI traverses all instances of SFIs in the primary SFI candidate graph, selects the instance with the lowest R_{sc}, and records this SFI and the node location of this SFI. Next, APPF-DSI determines whether the security of SFC_b meets the requirements or not. If the security requirements are not met, APPF-DSI calls



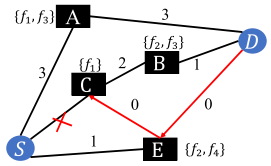
SF Type	f_1	f_2	f_3	f_4
Security	11	10	7	9
Implement cost	10	10	8	9

(a) An example of a PN.

SFP_p node candidate: $C(f_1), R_{sc} = \frac{11}{4}$

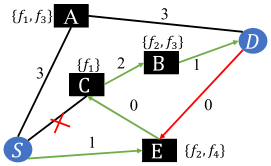
SFP_p	$S \to C(f_1) \to E \to D$
SFP_b	S

(b) Find primary Sec-SFC candidate node and SF.



SFP_p	$S \to C(f_1) \to E \to D$
SFP_b	S

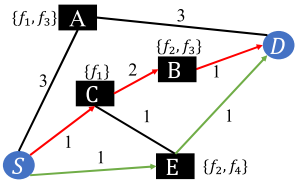
(c) Change the topology based on the SFP_p.



SFP_b node candidate: $E(f_2), R_{sc} = \frac{10}{4}$

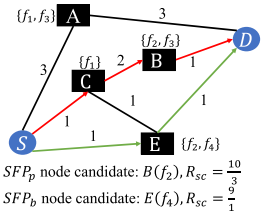
SFP_p	$S \to C(f_1) \to E \to D$
SFP_b	$S \to E(f_2) \to C \to B \to D$

(d) Find backup Sec-SFC candidate node.



SFP_p	$S \to C(f_1) \to B \to D$
SFP_b	$S \to E(f_2) \to D$

(e) Removes edges that appear in the primary and backup SFP.



SFP_p node candidate: $B(f_2), R_{sc} = \frac{10}{3}$

SFP_b node candidate: $E(f_4), R_{sc} = \frac{9}{1}$

SFP_p	$S \to C(f_1) \to B(f_2) \to D$
SFP_b	$S \to E(f_2, f_4) \to D$

(f) Differences between symmetric and asymmetric SFC dedicated protection.

FIGURE 6. Differences between symmetric and asymmetric SFC dedicated protection.

the ISSM algorithm to generate the backup SFI candidate graph, if there is an SFI* that does not overlap with an SFC_p, APPF-DSI will call AP-DSI to update SFP_b, SFC_b, otherwise APPF-DSI will call PF-DSI to update SFP_b, SFC_b. Then, according to the recorded the SFI and the node location of the SFI to update SFP_p, SFC_p. At last, the algorithm returns SFP_p, SFP_b, SFC_p, SFC_b, where SFC_p, SFC_b meet the security requirement, and SFP_p, SFP_b are disjoint. We use Fig. 6 to illustrate how the proposed APPF-DSI algorithm works. Fig. 6(a) shows an example of a PN, where the bracket besides the node represents its installable SFI(s) and the table below shows the information of each security-aware SF. When assuming that an NSR comes with $\Theta = 15$ and $bw = 1$, Fig. 6(b) shows the APPF-DSI algorithm how to find the primary SFI. First, by call ISSM algorithm to get a primary SFI candidate graph, this algorithm can find an SFI (f_1) and corresponding node (C) with the highest R_{sc}. The current $SFP_p(SFC_p)$ is $S \to C(f_1) \to E \to D$ Next, as shown in Fig. 6(c), based on current SFP_p, remove the path $S \to C$; reverse the path $C \to E \to D$ and change the weight to zero. Then, Fig. 6(d) shows this algorithm call ISSM algorithm to find SFI (f_2) and corresponding node (E) of backup Sec-SFC according to modified graph. The current $SFP_b(SFC_b)$ is $S \to E(f_2) \to C \to B \to D$. Due to $f_2 \cap SFC_p = \emptyset$ and a reused edge exists between the SFP_p and SFP_b, we remove this reused edge and reverse the remain path to get disjoint path shown in Fig. 6(e). The current $SFP_p(SFC_p)$ is $S \to C(f_1) \to B \to D$ and $SFP_b(SFC_b)$ is $S \to E(f_2) \to D$. To meet the client's security requirement, APPF-DSI repeatedly selects SFs and nodes to create primary and backup SFPs. If it does not exist reused edge having between the SFP_p and SFP_b, skip the step shown in Fig. 6(f). Finally, we can get a pair of disjoint primary and backup SFPs that satisfy the security requirements. The $SFP_p(SFC_p)$ is $S \to C(f_1) \to B(f_2) \to D$ and $SFP_b(SFC_b)$ is $S \to E(f_2, f_4) \to D$.

VI. NUMERICAL RESULTS AND ANALYSIS

A. SIMULATION SETTINGS

As the edge servers at the metropolitan edge network are generally densely-connected [1], [33], [34], [35], [36], [37], [38], we conduct experiments on a randomly-generated 40-node metropolitan edge network with a minimum node degree of 3. Each physical node has random computing resources in the range of [180, 200] to host multiple SF instances and provides the different types of SFs in a discrete-uniform range [1, 3]. Each physical link has a random bandwidth capacity in the range of [30, 40] to support connections between the primary and backup SF instances. In addition, we did a broad survey on the relationships between the security-aware SFs and their attack-protection abilities, as shown in table 3. Based on the above survey, we assume that 6 types of security-aware SFs are considered in our work, each of which has three versions. The embedding cost of an SF in version low (L), middle (M), and high (H) is randomly set in the range of [12, 16], [15, 18], and [18, 21], respectively; the security of an SF in version L, M, and H is [5, 7], [7, 9], and [9, 11], respectively[1] [39]. The link cost is randomly set in the [1, 10] range. Each NSR demands a security requirement within the range of [10, 30] units and a bandwidth resource within the range of [1, 3].

B. PERFORMANCE EVALUATION FOR DIVERSE PROTECTION MECHANISMS

In this subsection, we compare the performances of two protection mechanisms: (i) symmetric and asymmetric SFC protection, and (ii) cross-version protection and unified-version protection mechanisms. At first, we compare the SFC implementation cost using asymmetric and symmetric dedicated protection schemes. In order to make the convincible experiment, the benchmark algorithm of dedicated protection

[1]Note that, in order to normalize the related costs and make the simulation scenarios more generic, we use "units" instead of specific metrics, e.g., CPU cycles, Mbps, to quantify computing and bandwidth resources.

TABLE 3. Common security SFs and the network attacks they can resist.

Attack	Firewall	IDS/IPS	VPN	SIEM	NIDS	DPI
Network intrusion attack	✓	✓			✓	
Sniffing and eavesdropping	✓		✓			
Viruses and malware		✓				✓
Insider threat				✓		
Application level attack		✓				✓
Data breach			✓			
Average security	7	12	7	5	4	8

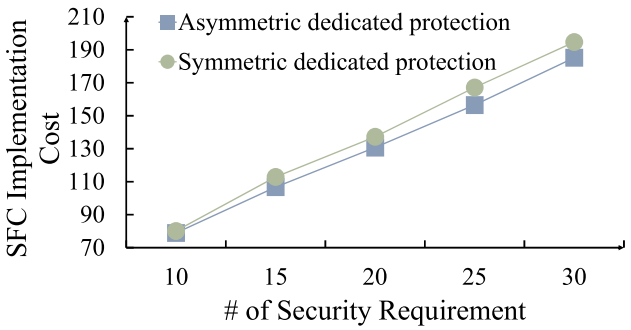


FIGURE 7. Symmetric dedicated protection vs. Asymmetric dedicated protection.

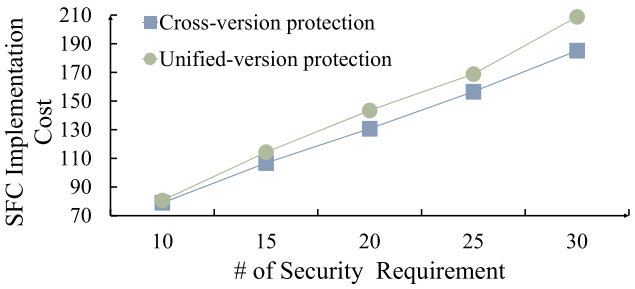


FIGURE 8. Cross-version protection vs. Unified-version protection.

is modified from the APPF-DSI algorithm, where only symmetric backup SFC structure is allowed. The benchmark algorithm first finds a primary SFP and a security-aware SFC according to the APPF-DSI algorithm, then removes the used links in the network, while finding an optimal backup SFC according to the layered graph-based mapping algorithm.

In Fig. 7, the green-circled and blue-squared curves represent the performances of symmetric dedicated protection and asymmetric dedicated protection, respectively. When varying the security requirements from 10 to 30, the SFC implementation cost is linearly increasing with the security requirements. Notably, when the security requirement is 10, the SFC implementation cost of the two schemes is basically the same. When the security requirements increase, the SFC implementation cost gap between these two schemes is gradually increasing. This is because, with an increasing security requirement, the length of the deployed SFC is gradually increasing, dedicated protection limits the possible implementation of the backup SFC, while asymmetric protection can provide a more flexible backup SFC selection; thereby, the cost gap between the two protection schemes gradually increases. Numerically, the asymmetric protection approach outperforms the dedicated protection approach as much as by 5%.

Next, we compare the implementation cost of cross-version protection and unified-version protection under the asymmetric protection scheme. In order to make the more convincible experiment, the benchmark algorithm is also extended from the APPF-DSI algorithm. In specific, the benchmark algorithm finds the primary and backup SFC process with the same processes as the APPF-DSI algorithm does, except that the version of the service function being searched must be in the same version as the first SF. In Fig. 8, the green-circled and blue-squared curves represent the same-version and cross-version protection performances, respectively. Since the same version protection limits the version of the SF in the SFC, the solution space that can be selected will be reduced. We infer that the performance of cross-version protection will be better than that of the same-version protection. This inference has been proved experimentally, as shown in Fig. 8. Numerically, cross-version protection outperforms same-version protection by an average of 7.91%.

C. PERFORMANCE EVALUATION ON DIVERSE PROTECTION APPROACHES

We extend state-of-the-art SFC protection schemes for comparison purposes [21], [22]. We implement [22] by: (i) constructing a hyper-graph to get all SFs cost and security, (ii) according to the ratio of cost and security to construct a pair of Sec-SFC, (iii) use layered graph mapping technique to create a primary SFP, (iv) pruning the created SFP from the PN, and creating the backup SFP via layered graph mapping technique. This method ID is denoted by greedy based on hyper graph (GB-HG). Meanwhile, we implement [21] by (i) random generating five pairs of Sec-SFC, (ii) selecting a pair of Sec-SFC with the least implementation cost to create the primary SFP via layered graph mapping technique, (iii) pruning the created SFP from the PN, and creating the backup SFP via layered graph mapping technique. This method ID is denoted by disjoint SFP based on layered graph mapping (DSB-LG). In the following figures, the blue, green, and red bars represent the performances of APPF-DSI,

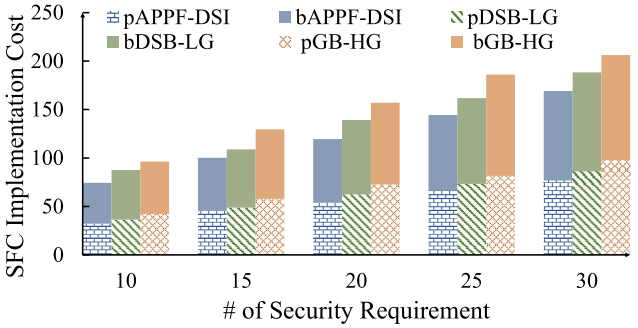


FIGURE 9. Total SFC implementation cost for different security requirements.

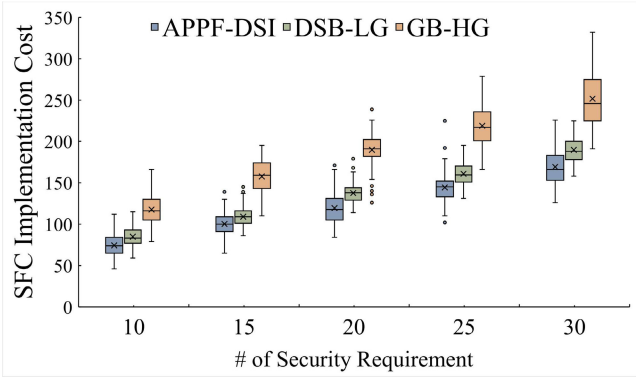


FIGURE 10. SFC implementation cost distribution under different security requirements.

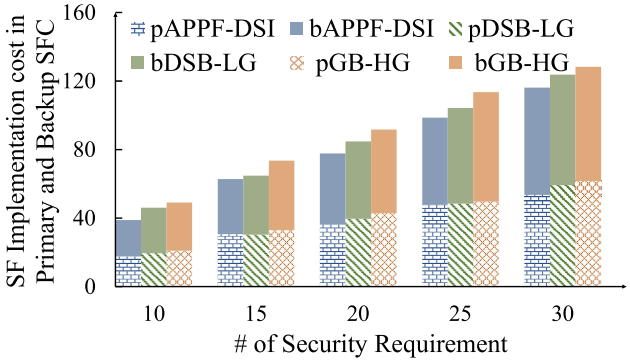


FIGURE 11. SF implementation cost in primary and backup SFC for different security requirements.

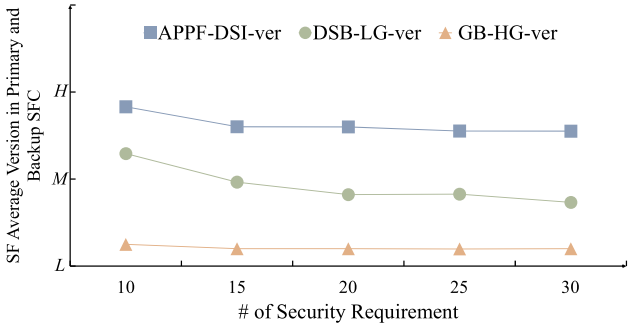


FIGURE 12. SF average version in primary and backup SFC for different security requirements.

DSB-LG, and GB-HG, respectively. Notably, we employ the prefix "p" and "b" to demonstrate the primary and backup costs.

Fig. 9 shows the SFC implementation cost of the above algorithms when varying the security requirements. Fig. 10 shows the SFC implementation cost under different security requirements. The rectangular area represents the middle 50 percent of the data distribution, the horizontal line in the rectangle represents the median, and the cross in the middle represents the average cost. As the security requirement increases, the distribution of data becomes more dispersed. Notably, the APPF-DSI algorithm has the best performance compared to DSB-LG and GB-HG. This is because when the APPF-DSI algorithm constructs the Sec-SFC and SFP, it selects the SF according to the security-cost central ratio, which tries its best to minimize the routing and implementation cost. In contrast, the DSB-LG algorithm uses the layered graph technology to complete the embedding of the Sec-SFC according to the constructed chain. However, the layered graph mapping technique optimizes the forwarding path construction process without taking into account the SF implementation cost. Meanwhile, the GB-HG algorithm creates the pair of disjoint pSFP and bSFP by combining the hyper-graph and greedy algorithm, which does not necessarily reduce the implementation cost of SFs as the selected SFs may provide low security; in turn, more SFs are needed for creating the primary and backup SFCs. Numerically, the APPF-DSI algorithm outperforms DSB-LG and GB-HG algorithms by an average of 12.34% and 28.22%, respectively.

Next, we conduct an in-depth analysis of the total SFC implementation cost. The total cost of SFC implementation consists of four parts, namely, the SF implementation cost of the primary and backup Sec-SFC, and the routing cost of the primary and backup SFP. Fig. 11, 12 and 13 show the SF implementation cost, average SF version, and routing cost on the primary Sec-SFC and the backup Sec-SFC.

It can be seen from Fig. 11 that the SF implementation cost of the three algorithms increases continuously with the increase of security requirements. As one can see from Fig. 11, with an increasing security requirement, APPF-DSI slightly outperforms the other two algorithms in primary SF implementation cost. Even though the APPF-DSI algorithm employs the security-cost center ratio, it jointly optimizes the primary and backup SFPs in the meantime. For the other two schemes, follow a primary-SFP-first fashion. As a result, APPF-DSI only slightly outperforms DSB-LG and GB-HG by an average of 6.06% and 8.47%, respectively.

In contrast, the cost gap of SF implementation among these three algorithms mainly comes from the SF implementation cost in the backup Sec-SFC. As analyzed above, the primary and backup cost ratio of the APPF-DSI, DSB-LG, and

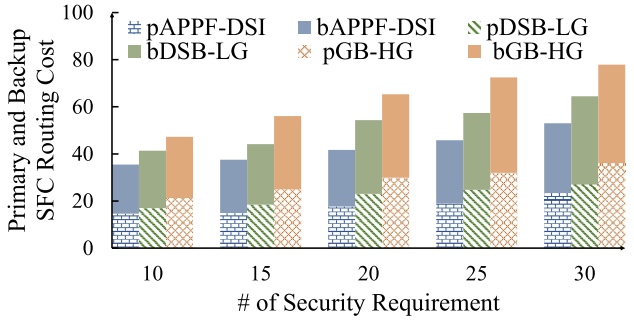


FIGURE 13. Primary and backup SFC router cost for different security requirements.

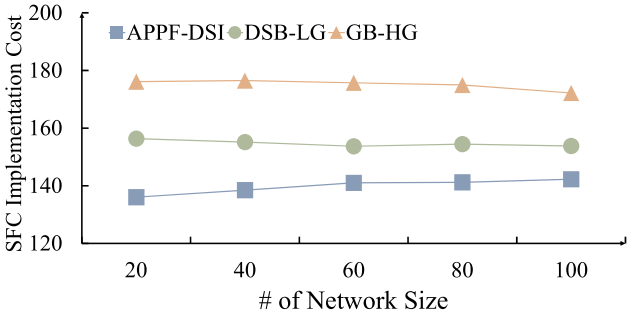


FIGURE 14. SFC implementation cost vs. network sizes.

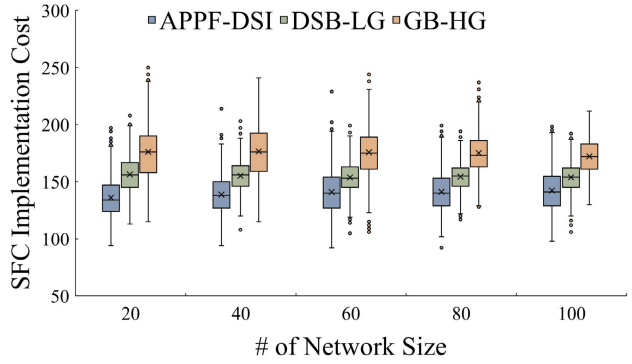


FIGURE 15. SFC implementation cost distribution under different network sizes.

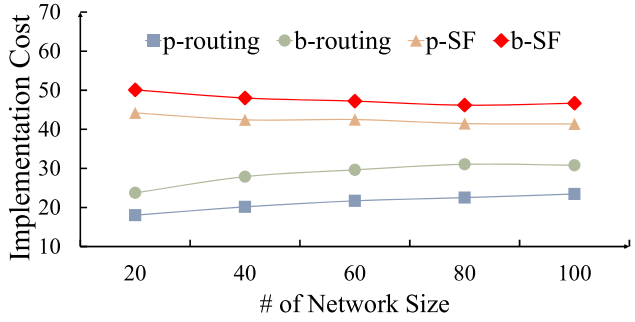


FIGURE 16. SFC implementation cost of APPF-DSI.

GB-HG algorithms are 12.35%, 17.62%, and 21.64%, respectively. Again, this is because the other two algorithms give priority to construct the primary Sec-SFC while considering the construction of the backup Sec-SFC. This is a local optimization of the primary Sec-SFC, without considering the impact on the construction of the backup chain, which leads to a rapid increase in the implementation cost of the backup chain. The APPF-DSI algorithm constructs the primary Sec-SFC and the backup Sec-SFC at the same time, which increases the feasible solution space and effectively reduces the implementation cost of SFs between the primary and backup Sec-SFC. Numerically, regarding the backup SF implementation cost, the APPF-DSI algorithm outperforms the DSB-LG and GB-HG algorithms by an average of 11.11%, and 21.89%, respectively.

Regarding the SF implementation cost, we also evaluate the SF versions taken by each scheme as shown in Fig. 12. As one can see, the APPF-DSI algorithm tends to select the SF of the H version, and the average value of the version is between the H version and M version. The average version of the SF selected by the DSB-HG algorithm fluctuates around the M version. The average version chosen by the GB-HG algorithm is close to the L version as it always selects the SF with low cost. Accordingly, from our experiments, the SF with a higher version can provide a better security guarantee in a cost-efficient fashion.

Fig. 13 shows the routing cost of the above three schemes when increasing. As one can see, the SF implementation cost of the three algorithms increases continuously and the gap between the three algorithms is also widening. This is because increasing the security requirement, the length of Sec-SFC is increase. The APPF-DSI algorithm considers the routing cost and SF cost at the same time, and optimizes the path while constructing the Sec-SFC. Therefore, the routing cost and the gap between the three algorithms are increasing. Numerically, the APPF-DSI algorithm outperforms DSB-LG and GB-HG algorithms by an average of 22.33% and 48.92% in the routing cost, respectively.

Finally, we analyze the SFC implementation cost with different network sizes of the proposed algorithms when the security requirement is 20. It can be seen from Fig. 14, when the network size changes, the SFC implementation cost of DSB-LG and GB-HG algorithms basically does not change, while the SFC implementation cost of APPF-DSI algorithm increases a little with the expansion of the network size. And Fig. 15 shows the SFC implementation cost under different network size. Similarly, the rectangular area represents the middle 50 percent of the data distribution, the horizontal line in the rectangle represents the median, and the cross represents the average cost. As the size of the network increases, the distribution of data becomes more concentrated. In order to better explain why this phenomenon occurs, we show the changing trend of the SFC implementation cost composition in the APPF-DIS algorithm Fig. 16. From Fig. 16, one can see that as the network size increases, the Sec-SFC routing cost of the primary and backup Sec-SFC also increases (as shown by

the orange and red lines). This is because as the size of the network increases, the distance between the randomly generated source and destination will also increase, which may cause more routing costs. It can also be seen from Fig. 16 that with the increasing the number of nodes in the network, the SF implementation cost of primary and backup Sec-SFC is decreasing (as shown by the blue and green dotted lines). This is because, for the same SF, its implementation cost on different nodes is different. The more nodes in the network, the more nodes can be selected. According to the characteristics of the APPF-DSI algorithm, it will select the node with the smallest security-cost center ratio for chain construction, so the implementation cost of SF will slightly decrease with the expansion of network size. On the one hand, the routing cost increases. On the other hand, the SF implementation cost decreases. Therefore, with the increases of the network size, the implementation cost is basically in a stable trend.

VII. CONCLUSION

In this work, we have comprehensively investigated the problem of how to optimize the cost of chaining, embedding, and protecting security-aware service function chains over the shared physical networks. We mathematically formulated the problem of security-aware service function chaining, embedding, and protection with multi-versioned SFs (SFCEP-MF) and proved its NP-hardness. To effectively protect the Sec-SFC, we have proposed a novel protection scheme, namely, asymmetric SFC dedicated protection, which can flexibly and efficiently construct backup SFC. Based on the asymmetric SFC dedicated protection scheme and augmenting path technique, we proposed an efficient heuristic algorithm called augmenting-path with primary-first disjoint SFP identifier (APPF-DSI). Our extensive simulation results have shown that (i) the asymmetric SFC dedicated protection can enhance the variety of creating primary and backup SFCs and outperforms the traditional symmetric SFC dedicated protection by an average of 4.86%, (ii) the cross-version SFC dedicated protection reduces the cost in an average of 7.91% compared to the unified-version SFC dedicated protection, and (iii) the proposed APPF-DSI outperforms the other two benchmark algorithms that are directly extended from the state-of-the-art by an average of 12.34% and 28.22%, respectively. Notably, the APPF-DSI algorithm can make full use of the computing and bandwidth resources and is capable of reducing the deployment cost.

REFERENCES

[1] D. Zheng, G. Shen, X. Cao, and B. Mukherjee, "Towards optimal parallelism-aware service chaining and embedding," *IEEE Trans. Netw. Service Manage.*, vol. 19, no. 3, pp. 2063–2077, Sep. 2022.

[2] R. Mijumbi, J. Serrat, J.-L. Gorricho, N. Bouten, F. De Turck, and R. Boutaba, "Network function virtualization: State-of-the-art and research challenges," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 236–262, 1st Quart., 2016.

[3] D. Zheng, C. Peng, X. Liao, and X. Cao, "Parallelism-aware service function chaining and embedding for 5G networks," in *Proc. Int. Conf. Comput. Commun. Netw. (ICCCN)*, Jul. 2021, pp. 1–9.

[4] Y. Li and M. Chen, "Software-defined network function virtualization: A survey," *IEEE Access*, vol. 3, pp. 2542–2553, 2015.

[5] D. Zheng, C. Peng, X. Liao, L. Tian, G. Luo, and X. Cao, "Towards latency optimization in hybrid service function chain composition and embedding," in *Proc. IEEE Conf. Comput. Commun.*, Jul. 2020, pp. 1539–1548.

[6] J. M. Halpern and C. Pignataro, *Service Function Chaining (SFC) Architecture*, document RFC 7665, 2015.

[7] D. Zheng, C. Peng, X. Liao, and X. Cao, "Toward optimal hybrid service function chain embedding in multiaccess edge computing," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 6035–6045, Jul. 2020.

[8] D. Bhamare, R. Jain, M. Samaka, and A. Erbad, "A survey on service function chaining," *J. Netw. Comput. Appl.*, vol. 75, pp. 138–155, Nov. 2016.

[9] D. Zheng, C. Peng, E. Guler, G. Luo, L. Tian, and X. Cao, "Hybrid service chain deployment in networks with unique function," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2019, pp. 1–6.

[10] 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 Autom. (ETFA)*, Sep. 2017, pp. 1–4.

[11] B. Wang, X. Cao, C. Peng, J. Li, and D. Zheng, "Towards service function chaining and embedding for multi-security guarantee levels," in *Proc. IEEE 8th Int. Conf. Comput. Commun. (ICCC)*, Dec. 2022, pp. 441–445.

[12] D. Zheng, X. Liu, W. Tang, H. Xu, and X. Cao, "Cost optimization in security-aware service function chain deployment with diverse vendors," in *Proc. IEEE Global Commun. Conf.*, Dec. 2023, pp. 2093–2098.

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

[14] 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.

[15] R. T. B. Ma, "Pay-as-you-go pricing and competition in congested network service markets," in *Proc. IEEE 22nd Int. Conf. Netw. Protocols*, Oct. 2014, pp. 257–268.

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

[17] 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.

[18] Y. Wang, P. Chau, and F. Chen, "Towards a secured network virtualization," *Comput. Netw.*, vol. 104, pp. 55–65, Jul. 2016.

[19] D. Zhao, L. Luo, H. Yu, V. Chang, R. Buyya, and G. Sun, "Security-SLA-guaranteed service function chain deployment in cloud-fog computing networks," *Cluster Comput.*, vol. 24, no. 3, pp. 2479–2494, Sep. 2021.

[20] D. Zheng, H. Gu, W. Wei, C. Peng, and X. Cao, "Network service chaining and embedding with provable bounds," *IEEE Internet Things J.*, vol. 8, no. 9, pp. 7140–7151, May 2021.

[21] N. Siasi, M. A. Jasim, A. Yayimli, and N. Ghani, "Service function chain survivability provisioning in fog networks," *IEEE Trans. Netw. Service Manage.*, vol. 19, no. 2, pp. 1117–1128, Jun. 2022.

[22] J. Gao, L. Feng, P. Yu, F. Zhou, Z. Wu, X. Qiu, J. Li, and Y. Zhu, "Resource consumption and security-aware multi-tenant service function chain deployment based on hypergraph matching," *Comput. Netw.*, vol. 216, Oct. 2022, Art. no. 109298.

[23] D. Zheng, E. Guler, C. Peng, G. Luo, L. Tian, and X. Cao, "Dependence-aware service function chain embedding in optical networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2019, pp. 1–6.

[24] E. Guler, D. Zheng, G. Luo, L. Tian, and X. Cao, "Embedding multicast services in optical networks with fanout limitation," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2018, pp. 1–6.

[25] M. Jalalitabar, E. Guler, D. Zheng, G. Luo, L. Tian, and X. Cao, "Embedding dependence-aware service function chains," *J. Opt. Commun. Netw.*, vol. 10, no. 8, pp. 64–74, Aug. 2018.

[26] S. Cai, F. Zhou, Z. Zhang, and A. Meddahi, "Disaster-resilient service function chain embedding based on multi-path routing," in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, May 2021, pp. 1–7.

[27] D. Zheng, C. Peng, B. Wang, and X. Cao, "Towards deterministic fault-tolerant service function slicing in edge networks," in *Proc. Int. Conf. Comput. Commun. Netw. (ICCCN)*, Jul. 2022, pp. 1–8.

[28] Y. Wang, L. Zhang, P. Yu, K. Chen, X. Qiu, L. Meng, M. Kadoch, and M. Cheriet, "Reliability-oriented and resource-efficient service function chain construction and backup," *IEEE Trans. Netw. Service Manage.*, vol. 18, no. 1, pp. 240–257, Mar. 2021.
[29] K. D. Joshi and K. Kataoka, "PSMART: A lightweight, privacy-aware service function chain orchestration in multi-domain NFV/SDN," *Comput. Netw.*, vol. 178, Sep. 2020, Art. no. 107295.
[30] D. Dwiardhika and T. Tachibana, "Virtual network embedding based on security level with VNF placement," *Secur. Commun. Netw.*, vol. 2019, pp. 1–11, Feb. 2019.
[31] X. Gang, H. Yuxiang, D. Tong, and L. Julong, "A dynamic composition mechanism of software-defined network security service chain," *J. Electron. Inf.*, vol. 38, pp. 1234–1241, Sep. 2016.
[32] J. W. Suurballe and R. E. Tarjan, "A quick method for finding shortest pairs of disjoint paths," *Networks*, vol. 14, no. 2, pp. 325–336, Jun. 1984.
[33] A. A. Barakabitze, A. Ahmad, R. Mijumbi, and A. Hines, "5G network slicing using SDN and NFV: A survey of taxonomy, architectures and future challenges," *Comput. Netw.*, vol. 167, Feb. 2020, Art. no. 106984.
[34] N. Shahriar, S. Taeb, S. R. Chowdhury, M. Zulfiqar, M. Tornatore, R. Boutaba, J. Mitra, and M. Hemmati, "Reliable slicing of 5G transport networks with bandwidth squeezing and multi-path provisioning," *IEEE Trans. Netw. Service Manage.*, vol. 17, no. 3, pp. 1418–1431, Sep. 2020.
[35] C. Peng, D. Zheng, S. Philip, and X. Cao, "Latency-bounded off-site virtual node protection in NFV," *IEEE Trans. Netw. Service Manage.*, vol. 18, no. 3, pp. 2545–2556, Sep. 2021.
[36] G. Le, S. Ferdousi, A. Marotta, S. Xu, Y. Hirota, Y. Awaji, M. Tornatore, and B. Mukherjee, "Survivable virtual network mapping with content connectivity against multiple link failures in optical metro networks," *J. Opt. Commun. Netw.*, vol. 12, no. 11, pp. 301–311, Nov. 2020.
[37] I. Pelle, F. Paolucci, B. Sonkoly, and F. Cugini, "Latency-sensitive edge/cloud serverless dynamic deployment over telemetry-based packet-optical network," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 9, pp. 2849–2863, Sep. 2021.
[38] B. Pan, F. Yan, X. Guo, X. Xue, and N. Calabretta, "On demand network services deployment in optical metro edge computing network based on user and application requests and infrastructure telemetry," in *Proc. Eur. Conf. Opt. Commun. (ECOC)*, Dec. 2020, pp. 1–4.
[39] J. Liu, W. Lu, F. Zhou, P. Lu, and Z. Zhu, "On dynamic service function chain deployment and readjustment," *IEEE Trans. Netw. Service Manage.*, vol. 14, no. 3, pp. 543–553, Sep. 2017.

BEN WANG received the bachelor's degree from the School of Optoelectronic Information Science and Engineering, Qingdao University, Qingdao, Shandong, in 2021. He is currently pursuing the master's degree with Soochow University, China. His research interests include network function virtualization and network optimization.

JUN LI received the Ph.D. degree from the KTH Royal Institute of Technology, in 2019. From 2018 to 2019, he was a Visiting Scholar with Princeton University. Currently, he is an Associate Professor with Soochow University. His research interests include optical access networks, 5G/6G optical transport networks, and network function virtualization. He has received the 2017 IEEE/Optical ACP Best Student Paper Award and the 2023 IEEE/Optical Best Paper Award.

SHAOHUA CAO (Member, IEEE) is currently an Associate Professor with the College of Computer and Communication, China University of Petroleum. His main research interests include network performance optimization, software-defined networks, cloud computing, the IoT, internet technology, computer hardware technology, computer software, and computer applications.

EVRIM GULER is currently an Associate Professor with the Department of Computer Engineering, Bartin University. His main research interests include network performance optimization, software-defined networks, cloud computing, the IoT, internet technology, computer hardware technology, computer software, and computer author biography applications.

DANYANG ZHENG (Member, IEEE) received the Ph.D. degree in computer science from Georgia State University, Atlanta, GA, USA, in 2021. He is currently an Associate Professor with Southwest Jiaotong University, China. His research interests include network function virtualization, software-defined networks, networking performance optimization, and combinational optimization.