Service Function Chaining with Deterministic Fault Tolerance in Optical Edge Networks

Danyang Zheng, *Member, IEEE*, Gangxiang Shen*, *Senior Member, IEEE*, Yongcheng Li, *Member, IEEE*, Xiaojun Cao, *Senior Member, IEEE*, and Biswanath Mukherjee, *Fellow, IEEE*

Abstract—In the upcoming 5G-and-beyond era, ultra-reliable low-latency communication (URLLC) services will be ubiquitous in edge networks. To improve network performance and quality of service (QoS), URLLC services could be delivered via a sequence of software-based network functions, also known as service function chains (SFCs). Towards reliable SFC delivery, it is imperative to incorporate deterministic fault tolerance during SFC deployment. However, deploying an SFC with deterministic fault tolerance is challenging because the protection mechanism needs to consider protection against physical/virtual network failures and hardware/software failures jointly. Against multiple and diverse failures, this work investigates how to effectively deliver an SFC in optical edge networks with deterministic fault tolerance while minimizing wavelength resource consumption. We introduce a protection augmented graph, called k -connected service function slices layered graph (KC-SLG), protecting against $k-1$ fiber link failures and $k-1$ server failures. We formulate a novel problem called deterministic-fault-tolerant SFC embedding and propose an effective algorithm, called most candidate first SF slices layered graph embedding (MCF-SE). MCF-SE employs two proposed techniques: k -connected network slicing (KC-NS) and k -connected function slicing (KC-FS). Through thorough mathematical proof, we show that KC-NS is 2-approximate. For KC-FS, we demonstrate that $k=3$ provides the best cost-efficiency. Our experimental results also show that the proposed MCF-SE achieves deterministic-fault-tolerant service delivery and performs better than the schemes directly extended from existing work regarding survivability and average cost-efficiency.

Index Terms—Optical edge network, network function virtualization, reliable service function chain embedding, deterministic fault tolerance.

I. INTRODUCTION

WITH the increasing deployment of 5G infrastructure, many edge servers are fiber-connected to provide ultra-reliable low-latency communication (URLLC) services for clients at the network edge [1]. With densely-connected edge servers, a service provider can enable ubiquitous access and agile provisioning while satisfying the required quality of service (QoS) for URLLC services. Traditionally, URLLC traffic traverses a sequence of middleboxes providing various functions (e.g., firewall, load balancer, etc.) [2]. However, these proprietary middleboxes only support dedicated functions and are expensive to maintain and manage [3]. Recently, network

function virtualization (NFV) is introduced to avoid dedicated middleboxes and facilitate agile service provisioning by converting the traditional hardware-based network functions into software-based modules, called service functions (SFs) [4]. In the NFV paradigm, SFs run on commodity/edge servers to process clients' data [5]. A client's network service request (NSR) generally demands a set of SFs and related resources (e.g., computing resources, bandwidth, virtual machines, etc.). To accommodate an NSR, the service provider can concatenate the demanded SFs into a service function chain (SFC) and embed it onto a shared physical network (PN). Specifically, a virtual machine that runs a specific SF is called an SF instance. The physical node that hosts an SF instance is called this SF's physical candidate, while the physical forwarding path for an SFC is called a service function path (SFP). The process of accommodating an NSR in a shared PN is referred to as service function chaining and embedding (SFCE) [6].

In 5G-and-beyond networks, strict fault tolerance is necessary for applications requiring strict QoS requirements such as intelligent transportation systems, smart manufacturing systems, and remote surgery, where a network interruption can cause a significant financial loss or casualty [7]–[10]. Delivering SFCs in such applications needs to provide fault tolerance against failures of both physical component (e.g., node or link) and software (e.g., SF instance). Most existing research concentrated on fault tolerance of a network [11]–[19], but few works have focused on fault-tolerant mechanisms of the SFC delivery [20]–[23]. In [22], two disjoint SFPs (primary and backup) are established to tolerate a single node, link, or SF instance failure. In [23], an SFP fault-tolerant mechanism is proposed to handle multiple SF instance failures. However, when delivering SFCs, these approaches cannot handle the coincidence of multiple and diverse failures (e.g., node/link failures and SF instance failures).

With ultra-reliability and fast transmission speed, the optical network is promising for reliable data transmission in edge computing systems [35]. The optical edge network (OEN) is generally deployed at an industrial park and carries many SFPs. Due to various service level agreements or different QoS requirements, SFPs need to be configured at different fault-tolerant levels. Here, the fault-tolerant level is correlated to the number of failures (including physical component and SF instance failures) that an ongoing SFP can simultaneously tolerate. In practice, delivering an SFC for applications with strict QoS demands requires a higher fault-tolerant level than regular applications. For example, an SFP for a remote surgery may need a higher fault-tolerant level than an SFP for cloud gaming. Meanwhile, a higher fault-tolerant level is at the

D. Zheng, G. Shen, Y. Li, and B. Mukherjee are with Suzhou Key Laboratory of Advanced Optical Communication Network Technology in the School of Electronic and Information Engineering, Soochow University, Suzhou, 215006, China (e-mail: dyzheng@suda.edu.cn, shengx@suda.edu.cn, ycli@suda.edu.cn, and bmukherjee@suda.edu.cn).

X. Cao is with the Department of Computer Science, Georgia State University, Atlanta, GA, 30302 USA (e-mail: cao@gsu.edu).

G. Shen is the corresponding author of this work.

expense of higher resource consumption. With limited resources at the network edge, a fundamental problem is how to provide appropriate fault tolerance for SFPs while minimizing network resource consumption in OENs. Here, we call the appropriate fault tolerance as the *deterministic fault tolerance*. Specifically, we use an integer k to denote the level of deterministic fault tolerance. An SFP with k -fault tolerance can survive up to $k-1$ link failures and $k-1$ node (including server and SF instance) failures.

In this work, we investigate how to effectively deliver an SFC in the OEN with deterministic fault tolerance. We define a new problem called deterministic-fault-tolerant SFC embedding (DFT-SFCE), which tries to satisfy a service request using a k -fault-tolerant SFP while minimizing wavelength resource consumption. We first introduce an augmented graph, called k -connected SF slices layered graph (KC-SLG). Based on such a KC-SLG, we propose an efficient heuristic algorithm, called most candidate first SF slices layered graph embedding (MCF-SE), which comprises two major processes: k -connected network slicing (KC-NS) and k -connected function slicing (KC-FS). We prove that KC-NS is 2-approximate and verify its 2-approximate performance with experimental results. When investigating the impact of k values in KC-FS, we find that (1) $k = 3$ can achieve the best cost-protection ratio (i.e., wavelength resource consumption divided by k), and (2) $k = 3$ has the best average cost-survivability ratio (i.e., wavelength resource consumption divided by the survivability of the deployed services, where the survivability is the ratio between the number of survived services after failures and the total number of deployed services). Moreover, compared to the schemes directly extended from the existing works [22][23], the proposed MCF-SE algorithm has a better survivability and a higher average cost-survivability ratio.

The rest of this paper is organized as follows. Section II summarizes the related work. Section III introduces the k -connected SF slices layered graph. We formulate the problem of deterministic-fault-tolerant SFC embedding (DFT-SFCE) in Section IV. Sections V and VI analyze the NP-hardness of DFT-SFCE, and present algorithms to solve the problem under various network scenarios. Section VII shows and analyzes experimental results. Section VIII concludes the paper.

II. RELATED WORK

With increasingly-deployed 5G infrastructures, ubiquitous access, low latency, and reliable communication become essential for delivering clients' services. In the literature, significant efforts have been put into investigating fault-tolerant mechanisms for providing reliable communications [11]–[23]. In Fig. 1, we broadly classify the existing work on network failures as: (1) one-type failure or (2) mixed-type failure. The works on “one-type failure” consider the same type of failure (i.e., either node failure, link failure, software failure, etc.), while the works on “mixed-types failure” take failures of diverse types into account. Moreover, in Fig. 1, the “single failure” represents that one failure is fixed before the subsequent failure occurs, while the term “multiple failures” means that multiple failures may coincide.

Fig. 1: Classifications of approaches to protect service delivery process against failures.

Single and One-Type Failure: Many works have been done to protect the network from a single and one-type failure [12][13][15][21]. In [12], the authors proposed a scheme that finds the most efficient backup path to protect the service from single link failure with the group table-based re-routing technique. The authors in [13] exploited bandwidth squeezing and survivable multi-path provisioning to protect a 5G network slice from single fiber failure while minimizing bandwidth consumption. In [15], the authors investigated the impact of different isolation levels on resource occupation in reliable slicing. They proposed an integer linear program (ILP) to protect the deployed services from single fiber link failure. To protect the service from a single SF instance failure, authors in [21] proposed a heuristic algorithm to steer the traffic for a given SFC reliably.

Multiple and One-Type Failures: Considering the coincidence of multiple and one-type failures, Refs. [11], [14], [16], [17] proposed heuristic or approximate approaches. To protect against multiple link failures, the authors in [11] designed backup schemes and proposed a probabilistic distribution function to minimize the required backup resource in polynomial time. In [14], exploiting content connectivity, the authors proposed an ILP model to reliably map the virtual network and make it fault-tolerant for multiple link failures. Taking multiple node failures into account, Ref. [16] proposed a graph theory-based algorithm to compute the minimum required backup transmission resource. Ref. [17] investigated how to place backup nodes to protect the network from multiple physical node failures while minimizing the backup computing resource consumption.

Single and Mixed-Type Failure: During service delivery, failures of different types (e.g., link, server, SF instance) might occur one after another (i.e., one failure is fixed before the subsequent failure occurs). To guarantee service delivery, Refs. [18], [22], [23] proposed heuristic and approximate approaches to handle single and mixed-type failure. Ref. [18] proposed a primary and backup resource allocation model using probabilistic protection to protect the deployed services. Ref. [22] proposed a method for embedding an SFC via two link-disjoint SFPs to protect the service delivery from single mixed-type failure. Taking SF instance distribution into account, Ref. [23] proposed an algorithm to backup the in-service SF instances in an off-site manner.

Multiple and Mixed-Type Failures: In practice, failures with different types may co-occur. Compared to the single and mixed-type failure, multiple and mixed-type failures are more complex as the protection approach needs to jointly take failures of all types into account instead of handling them one at a time. To the best of our knowledge, how to protect the

Fig. 2: Typical networking failures and the corresponding solutions.

service delivery process against multiple and diverse failures is challenging and needs to be studied. In the following sections, to fill this gap, we first introduce the preliminary discussion for the proposed technique in augmented graphs, then we formulate the problem of deterministic-fault-tolerant SFC embedding, and propose efficient heuristics to solve the problem.

III. K-CONNECTED SF SLICES LAYERED GRAPH

A. Network Failures in Service Delivery

Fig. 2 shows three typical failures in SFC service delivery. In Fig. 2a, an SFC ($S \Rightarrow f_1 \Rightarrow f_2 \Rightarrow f_3 \Rightarrow D$) is delivered through an SFP ($S \rightarrow SEV_1 \rightarrow SEV_3 \rightarrow D$). In this SFC, S, D are the source and destination, while f_1, f_2, f_3 are the required SFs. In the SFP, SEV_1 hosts service function instances for f_1 and f_2 , while SEV_3 hosts f_3 's instance.

Fig. 2b shows that a link failure occurs between SEV_1 and SEV_2 . It interrupts all SFPs that bypass this failed link. Here, a backup route $SEV_1 \rightarrow SEV_2 \rightarrow SEV_3$ can help the SFPs recover from the link failure that occurs between SEV_1 and SEV_3 .

Fig. 2c shows an SF instance failure at SEV_1 . It only impacts one SFP that employs this failed SF instance. Here, a backup SF instance f'_2 can be initialized at SEV_2 to protect the service.

Fig. 2d shows SEV_1 's failure, which impacts all SF instances on this server (f_1 and f_2), leading to service disruption of multiple SFPs that require data processing from this server. Similar to [36], this work assumes that a server failure does not affect the node's routing functionality. Therefore, SEV_2 can perform the role of the backup server that initializes all backup SF instances for SEV_1 , and a backup route $SEV_1 \rightarrow SEV_2 \rightarrow SEV_3$ is employed to ensure SFC delivery.

The above three schemes protect SFC delivery from a single and one-type failure. However, directly applying these approaches might not be efficient when diverse failures coincide. For example, when applying the technique in Fig. 2c for joint failures of f_2 's instance and the link between SEV_1 and SEV_2 , the SFC delivery process will be disrupted. To efficiently protect service delivery against multiple and diverse failures, we broadly specify the SFC delivery into two processes: (1) SF execution and (2) routing. The SF execution process deals with

the client's data through the required SFs, while the routing process transmits the client's data from the source to the destination. Making an SFC delivery fault-tolerant is equivalent to making these two processes fault-tolerant. To this end, we propose two inter-related augmented graphs to achieve fault tolerance for the SF execution and routing processes in the following subsections.

B. k-Connected Service Function Slicing (KC-SFS)

We use an integer k to indicate the level of deterministic fault tolerance from a client's request. The k -fault tolerance means the SFC delivery process can survive up to $k-1$ link failures and $k-1$ node (including server and SF instance) failures. Here, we propose a k -connected SF slice (KC-SFS) augmented graph based on network slicing and k -edge-connected graph techniques to facilitate deterministic fault tolerance and resource sharing during SFC embedding.

Inspired by the network slicing technique, we introduce the service function slice (SFS) augmented graph. We use a 3-tuple $SFS_f = \langle f, V_f, E_f \rangle$ to represent an SF slice, where f represents a specific SF, V_f is the set of f 's instances, and E_f represents the virtual connections among f 's instances. Fig. 3a shows an example of an f_1 's slice, where f_1^i is the i^{th} instance of f_1 . Different from the network slice, an SFS may not be connected. Embedding an SFS is similar to virtual network embedding, whereas one SF instance is hosted by one dedicated physical node, and one virtual link in SFS is accommodated by reserving bandwidth either at a physical link (i.e., $f_1^1 \Leftrightarrow f_1^2$ to $A \Leftrightarrow B$) or along a physical path (i.e., $f_1^3 \Leftrightarrow f_1^4$ to $C \Leftrightarrow E \Leftrightarrow F$). SFS allows resource sharing among multiple SFCs. SFS is scalable as one newly-installed SF instance can directly join the corresponding SFS, and one new SF can instantiate a corresponding SFS without impacting the existing SF slices.

In graph theory, a k -edge-connected graph stays connected after removing any $k-1$ edges¹ [24]. As a link failure in a PN is similar to removing an edge from a graph, we extend the SF slice augmented graph to the k -connected SF slice (KC-SFS) augmented graph. A KC-SFS is represented by a 4-tuple $k-SFS_f = \langle f, k, V_f, E_f \rangle$. When using δ as the degree of a node, a KC-SFS owns the following features: (1) it is connected, (2) each SF instance connects to at least k other SF instances (i.e., $\delta_{f_i} \geq k, \forall f_i \in V_f$), (3) the number of maximum flows between any two SF instances is no less than k , and (4) there are no parallel edges between two SF instances (i.e., $|V_f| \geq k$). When embedding a KC-SFS, each SF instance is initialized at one dedicated server, while edges must be embedded onto link-disjoint paths in the PN. Specifically, for every two edges in

Fig. 3: SF slice and k -connected SF slice.

¹ In this work, "edge" represents an arc in the graph or service function slice, while "link" refers to a connection in the network.

KC-SFS, there is no intersection between two physical paths that host them. Fig. 3b illustrates an example of embedding a 2-connected SFS onto a PN, where the circled number represents the virtual link embedding result. Each edge in Fig. 3b is hosted by a physical path (link) that is link-disjoint from the others. As in work [36], this work also assumes that every physical node comprises a server and a routing component (e.g., router, switch, etc.). Thus, a server failure only impacts SF executions on this physical node instead of routing functionality.

Lemma 1. *The embedding result of a KC-SFS remains connected after $k-1$ physical link failures.*

Proof. Since each SF instance connects to at least k other SF instances in a KC-SFS, the minimum degree of an SF instance is k . Based on Menger's theorem, a KC-SFS is a k -edge-connected graph [24]. As every edge in KC-SFS is embedded onto a link-disjoint path, every link failure in PN only impacts one edge in KC-SFS. Due to a link-disjoint embedding policy, $k-1$ physical link failures can at most break $k-1$ edges in KC-SFS. Therefore, KC-SFS and its embedding result remain connected after $k-1$ physical link failures in the PN. ■

Lemma 2. *The embedding result of a KC-SFS can protect against $k-1$ SF instance failures.*

Proof. Since the quantity of SF instances in a KC-SFS is at least k , at least one instance survives after $k-1$ SF instance failures. As KC-SFS is connected, the embedding result of a KC-SFS can tolerate $k-1$ SF instance failures. ■

Lemma 3. *The embedding result of a KC-SFS can protect against $k-1$ physical server failures.*

Proof. Since each server only installs one instance, $k-1$ server failures are equivalent to $k-1$ SF instance failures at one specific KC-SFS. Based on Lemma 2, this lemma holds. ■

Theorem 1. *The embedding result of a KC-SFS can protect against $k-1$ link failures and $k-1$ server failures.*

Proof. Based on Lemmas 1 and 3, the embedding result remains connected after $k-1$ link failures, while at least one server survives after $k-1$ server failures. In other words, the embedding result is connected, and there is at least one available path from any failed server to the survived server. Thus, no matter which failed server is employed for the reliable service, one available path always connects this failed server to the survived server. As a result, the embedding result can protect against $k-1$ link failures and $k-1$ server failures. ■

Based on Theorem 1, Lemma 2, and Lemma 3, the following lemma holds.

Lemma 4. *The embedding result of a KC-SFS can jointly protect against $k-1$ link failures and $k-1$ SF instance failures.*

According to the above analysis, employing KC-SFS for every required SF makes the SF execution process fault-tolerant. Next, we discuss how to provide fault tolerance on the routing process for a KC-SLG.

C. k -Connected SFS based Layered Graph

As various SFs have diverse instances distributed across the PN, each required SF must generate one KC-SFS to provide

deterministic fault tolerance. We use $\mathbb{SFS}$ to represent the set of k -connected SF slices. For convenience, we denote an SFC request by $\text{SFC} = (S, f_1, f_2, \dots, f_\tau, D)$, where τ represents the number of SFs in the SFC. Correspondingly, the set of KC-SFS is represented by $\mathbb{SFS} = (S, k\text{-SFS}_{f_1}, \dots, k\text{-SFS}_{f_\tau}, D)$. Next, we propose the k -connected SFS-based layered graph (KC-SLG) to make the routing process deterministically fault-tolerant.

A KC-SLG is represented by a 2-tuple $\text{KC-SLG} = \langle E_{\text{SLG}}^r, \mathbb{SFS} \rangle$, where E_{SLG}^r represents the set of routing edges in KC-SLG. Specifically, there are k routing edges connecting each two adjacent k -connected SF slices in KC-SLG. Based on the index of each SF, the set of routing edges in a KC-SLG can be represented by $E_{\text{SLG}}^r = (E_{S,1}^r, E_{1,2}^r, \dots, E_{\tau,D}^r)$, where $E_{i,i+1}^r$ is the set of routing edges between the i^{th} and $(i+1)^{\text{th}}$ KC-SFS. Fig. 4 illustrates a 2-connected SFS-based layered graph, where the edges in the red circle are routing edges. Note that a KC-SFS with higher fault tolerance can join the KC-SLG with a relatively lower level of fault tolerance. For example, in Fig. 4, the SFS_{f_2} is 3-fault-tolerant.

Fig. 4: An example of 2-connected SFS based layered graph.

Based on Menger's theorem, a KC-SLG is a k -edge-connected graph. If every edge in KC-SLG is embedded onto a link-disjoint path, based on Theorem 1, the embedding result is k -fault-tolerant at the expense of high bandwidth consumption. To save the network resource while guaranteeing deterministic fault tolerance, we employ the following embedding policy: (1) edges in the same KC-SFS (e.g., all edges in 2-SFS_{f_1}) or the same routing edge set (e.g., all edges in $E_{1,2}^r$) should be embedded onto link-disjoint paths, and (2) edges in different KC-SFS (e.g., an edge in 2-SFS_{f_1} and an edge in 3-SFS_{f_2}) and routing edge sets (e.g., an edge in $E_{1,2}^r$ and an edge in $E_{2,D}^r$) can share the same physical path (link). Theorem 2 illustrates that the above embedding policy can guarantee deterministic fault tolerance. Meanwhile, there is only one activated SFP when delivering an SFC. Given bw as the bandwidth resource demand, every link that supports an edge in KC-SLG only needs to reserve bw bandwidth. That is, even though one physical link may host multiple edges in one KC-SLG, only bw bandwidth in each direction is reserved by this physical link.

Theorem 2. *The embedding result of a KC-SLG can protect against $k-1$ link failures and $k-1$ server failures.*

Proof. Based on the proposed KC-SLG embedding policy, a link failure in the PN can impact at most one edge at each routing edge set or KC-SFS in KC-SLG. This is because every edge in the same routing edge set or KC-SFS is embedded onto link-disjoint paths. When $k-1$ link failures coincide, there is at least one edge survived at each routing edge set, and each KC-SFS remains connected according to Lemma 1. Thus, KC-SLG remains connected. In addition, a server failure impacts at most one SF instance in every KC-SFS. As geographically-

distributed servers host at least k instances for one KC-SFS, when $k-1$ server failures coincide, at least one server survives. Overall, the embedding result can survive after $k-1$ link failures and $k-1$ server failures. ■

Similarly, based on Theorem 2, Lemma 2, and Lemma 3, the following lemma holds.

Lemma 5. *The embedding result of a KC-SLG can protect against $k-1$ link failures and $k-1$ SF instance failures.*

IV. DETERMINISTIC-FAULT-TOLERANT SERVICE FUNCTION CHAIN EMBEDDING (DFT-SFCE)

A. Optical Edge Network (OEN)

We employ the optical edge network as the substrate (or physical) network, where edge servers are connected via fiber links [14], [25]-[27]. The optical edge network (OEN) is denoted by $OEN = (\mathbb{K}, N, L)$, where OEN is $\mathbb{K}$ -edge-connected, while N and L represent the set of nodes and bidirectional fiber links, respectively. In our assumption, every physical node has a wavelength converter, and we leave the extension to the case of wavelength continuity for future work. A physical node $n \in N$ hosts a set of SF instances denoted by $\mathbb{F}_n$, and it can support at most ψ_f^n instances of an SF $f \in \mathbb{F}_n$. Each fiber link $l \in L$ is bidirectional and has ω_l available wavelength resources, each of which supports a certain amount of bandwidth. For convenience, a fiber link l is represented by $l_{m,n}$, where $m, n \in N$ are its endpoints. A lightpath p is composed of multiple fiber links, and P denotes the set of lightpaths in the OEN. As there can be several lightpaths from one node to another, we use $p_{m,n,j}$ to represent the j^{th} lightpath from m to n . Note that $j \in [0, |\mathbb{P}_{m,n}|]$, where $\mathbb{P}_{m,n}$ is a lightpath set and includes all lightpaths whose endpoints are m, n , and $\mathbb{P}_{m,n} \subset P$.

B. Ultra-Reliable SFC Request (USR)

An ultra-reliable SFC request is denoted by a 4-tuple $USR = \langle k, V, SFC, bw \rangle$, where k denotes the level of deterministic fault tolerance, V is the set of required SFs (including source s and destination d), SFC represents the SFs' execution order, and bw is the number of demanded wavelength resources. To guarantee deterministic fault tolerance, k must be no more than $\mathbb{K}$ of a given OEN (i.e., $k \leq \mathbb{K}$). The number of required SFs in an SFC is τ .

C. Problem Formulation

Given a USR and an OEN, we investigate a problem of how to deliver USR over OEN with a deterministic fault tolerance such that wavelength resource consumption is minimized. We call this problem deterministic-fault-tolerant SFC embedding (DFT-SFCE), which jointly creates and embeds a KC-SLG onto the shared OEN. The DFT-SFCE is specified to the k -connected SF slicing and routing processes to guarantee the deterministic fault tolerance of delivering a reliable SFC. Table I lists the notations used in the problem formulation. For each SF f^i in the request, we consider composing and embedding the slice of f^i onto the physical nodes that have installed the corresponding instance, and this set of physical nodes is denoted by N_{f^i} . The objective of the proposed DFT-SFCE is to minimize wavelength resource consumption as below:

Table I: Notations for Problem Formulation

Notations	Definition
a, b, m, n	Physical nodes $a, b, m, n \in N$.
τ	Number of SFs in an ultra-reliable SFC request.
f^i	i^{th} SF in the USR.
N_{f^i}	Set of physical node that hosts f^i 's instance, $N_{f^i} \subseteq N$.
$p_{m,n,j}$	j^{th} lightpath from m to n .
$p_{m,n,j}^i$	= 1, if lightpath $p_{m,n,j}$ is used to host an edge in i^{th} KC-SLG; = 0, otherwise.
$l_{m,n}^i$	= 1, if fiber link $l_{m,n}$ is used to host an edge in i^{th} KC-SLG; = 0, otherwise.
$p_{m,n,j}^{i,i+1}$	= 1, if lightpath $p_{m,n,j}$ is used to host a routing edge between i^{th} and $(i+1)^{th}$ KC-SFS; = 0, otherwise.
$l_{m,n}^{i,i+1}$	= 1, if fiber link $l_{m,n}$ is used to host a routing edge between i^{th} and $(i+1)^{th}$ KC-SFS; = 0, otherwise.
$l_{m,n}^{SLG}$	= 1, if fiber link $l_{m,n}$ is used to host an edge for KC-SLG; = 0, otherwise.
k	Level of deterministic fault tolerance.
bw	Number of required wavelength resources.
$\omega_{l_{m,n}}$	Number of available wavelength resources at $l_{m,n}$.

$$\min \sum_{m,n \in N} l_{m,n}^{SLG} * bw \quad (1)$$

k -Connected Service Function Slicing: In this process, edges are embedded via link-disjoint lightpaths to connect the instances of f^i . Eqs. (2) and (3) check whether a lightpath $p_{m,n,j}$ and a fiber link $l_{m,n}$ are used to host an edge in the i^{th} KC-SFS or not. Eq. (4) ensures that there are k lightpaths connecting one physical node that hosts an instance of f^i to other physical nodes in N_{f^i} . Eq. (5) constrains that each fiber link can host at most one edge for the i^{th} KC-SFS.

$$p_{m,n,j}^i = \begin{cases} 1, & p_{m,n,j} \text{ is used to host an edge in } i^{th} \text{ KC-SFS,} \\ 0, & \text{otherwise.} \end{cases} \quad (2)$$

$$l_{m,n}^i = \begin{cases} 1, & l_{m,n} \text{ is used to host an edge in } i^{th} \text{ KC-SFS,} \\ 0, & \text{otherwise.} \end{cases} \quad (3)$$

$$\sum_{n \in N_{f^i}} p_{m,n,j}^i \geq k, \forall m \in N_{f^i}, \forall i \in [1, \tau], \forall j \in [0, |\mathbb{P}_{m,n}|] \quad (4)$$

$$\sum_{p_{m,n,j} \in P} p_{m,n,j}^i \leq l_{a,b}^i \leq 1, \forall l_{a,b} \in p_{m,n}, \forall i \in [1, \tau] \quad (5)$$

KC-SLG Routing Embedding: The edges in the same E_{SLG}^r must be embedded onto link-disjoint lightpaths. Eqs. (6) and (7) represent whether $p_{m,n,j}$ and $l_{m,n}$ are used to host a routing edge or not. Eq. (8) ensures that there are at least k lightpaths between any two adjacent KC-SFS. Note that $i = 0$ represents the source node, while $i = \tau + 1$ represents the destination. Eq. (9) constrains each fiber link to host at most one routing edge between any two KC-SFS. In other words, edges in the same routing edge set are embedded onto link-joint lightpaths.

$$p_{m,n,j}^{i,i+1} = \begin{cases} 1, & p_{m,n,j} \text{ hosts a routing edge between two KC-SFS} \\ 0, & \text{otherwise.} \end{cases} \quad (6)$$

$$l_{m,n}^{i,i+1} = \begin{cases} 1, & l_{m,n} \text{ hosts a routing edge between two KC-SFS} \\ 0, & \text{otherwise.} \end{cases} \quad (7)$$

$$\sum_{m \in N_{f^i}} p_{m,n,j}^{i,i+1} \geq k, \forall n \in N_{f^{i+1}}, \forall i \in [0, \tau], \forall j \in [0, |\mathbb{P}_{m,n}|] \quad (8)$$

$$\sum_{p_{m,n,j} \in P} p_{m,n,j}^{i,i+1} \leq l_{a,b}^{i,i+1} \leq 1, \forall l_{a,b} \in p_{m,n}, \forall i \in [0, \tau] \quad (9)$$

KC-SLG Embedding: Eq. (10) identifies whether a fiber link is used for KC-SLG embedding or not. Eq. (11) determines whether a fiber link $l_{m,n}$ is used to host an edge in one KC-SFS or routing edge set. Note that, in Eq. (11), M is a very large integer that ensures $l_{m,n}^{SLG}$ is either 0 or 1. Eq. (12) requires that any fiber links used for embedding KC-SLG should have enough wavelength resources.

$$l_{m,n}^{SLG} = \begin{cases} 1, & l_{m,n} \text{ hosts an edge in KC-SLG} \\ 0, & \text{otherwise.} \end{cases} \quad (10)$$

$$l_{m,n}^{SLG} \geq \frac{(\sum_{i \in [1, \tau]} l_{m,n}^i + \sum_{i \in [0, \tau]} l_{m,n}^{i,i+1})}{M}, \forall m, n \in N \quad (11)$$

$$l_{m,n}^{SLG} * bw \leq \omega_{m,n}, \forall m, n \in N \quad (12)$$

V. NP-HARDNESS ANALYSIS

In this section, we analyze the NP-hardness of the proposed DFT-SFCE problem in various networking scenarios. We first illustrate that, when given an OEN and a USR, there exists a solution for DFT-SFCE if wavelength resources are enough at each fiber link and $k \leq \mathbb{k}$.

Theorem 3. *When given an OEN with enough wavelength resources, there exists a solution for DFT-SFCE if $k \leq \mathbb{k}$.*

Proof. Since each fiber link in OEN has enough wavelength resources, the given OEN is $\mathbb{k}$ -edge-connected. As $k \leq \mathbb{k}$ and one SF slice with a higher fault tolerance level can participate in the KC-SLG with a lower fault tolerance level, the given OEN can be regarded as a KC-SLG composed of τ replicas of itself which is one solution for DFT-SFCE. ■

According to Theorem 3, the following lemma holds.

Lemma 6. *After pruning the fiber links without enough wavelength resources, there must exist a solution for DFT-SFCE if the number of maximum flows between any two nodes is no less than k in the pruned OEN.*

Next, we prove that minimizing wavelength resource consumption for composing and embedding a KC-SLG over a given OEN is NP-hard. Specifically, we discuss the NP-hardness of the k -connected SF slicing and routing processes.

Theorem 4. *Minimizing wavelength resource consumption for composing and embedding one k -connected service function slice is NP-hard.*

Proof. We start our proof with a particular scenario where all physical nodes have instantiated one required SF instance. In this scenario, using the minimum number of wavelength

resources to compose and embed this KC-SFS is equivalent to finding the k -edge-connected spanning subgraph with the minimum size (i.e., the number of edges in the subgraph). Since finding the minimum k -edge-connected spanning subgraph is NP-hard, minimizing wavelength resource consumption of composing and embedding a KC-SFS when all physical nodes have one required SF instance is also NP-hard [28]. When some physical nodes instantiate the required SF instance, this problem remains NP-hard. When $k = 1$, regarding the physical node that instantiates one required SF instance as a sink, this problem becomes the minimum Steiner tree problem [29]. Overall, minimizing wavelength resource consumption for composing and embedding one KC-SFS is NP-hard. ■

Since diverse KC-SFS can share wavelength resources to host their edges, composing and embedding all KC-SFS in a KC-SLG is equivalent to a collaborative process of maximizing the sharing of wavelength resources among all required KC-SFS. Thus, the following lemma holds according to Theorem 4.

Lemma 7. *Minimizing the wavelength resource cost for composing and embedding all KC-SFS in a KC-SLG is NP-hard.*

Next, we discuss the NP-hardness of the routing process in DFT-SFCE. The edges in the same routing edge set are required to be embedded onto link-disjoint lightpaths. Minimizing wavelength resource consumption of this process is equivalent to the k edge-disjoint shortest paths (KDSP) problem, which is NP-hard [30]. Accordingly, the following lemma holds.

Lemma 8. *Minimizing wavelength resource consumption for the routing process in DFT-SFCE is NP-hard.*

Overall, optimizing DFT-SFCE is challenging as the optimization problem of either process (i.e., k -connected SF slicing process or routing process) is NP-hard. To study this problem, we first investigate the k -connected SFS embedding process in the following scenarios: (1) all physical are candidates of the required SF, and (2) some physical nodes are candidates of the required SF. Correspondingly, we propose the k -connected network slicing (KC-NS) scheme for scenario (1) and subsequently extend it for scenario (2), namely, k -connected function slicing (KC-FS). With the above techniques, we then propose the most candidate first SF layered graph embedding (MCF-SE) algorithm to jointly optimize KC-SLG's composing and embedding processes in DFT-SFCE.

VI. MOST CANDIDATE FIRST SLG EMBEDDING (MCF-SE)

A. k -Connected Network Slicing

First, we investigate the k -connected SFS embedding process when the required SF has been installed at every edge server in a shared OEN. Ideally, when the OEN is a complete graph (i.e., each edge server is directly connected to each other through a dedicated fiber link), a lower bound on wavelength resource consumption of embedding a k -connected SFS is $\left\lceil \frac{|N|*k}{2} \right\rceil * bw$, where each SF instance connects to at least k other neighbors, and the round-up process works when both $|N|$ and k are odd.

Since the OEN is probably not a complete graph in practice, it is hard to achieve the lower bound, as proved by Theorem 4. Based on this lower bound, wavelength resource consumption of embedding a k -connected SFS is correlated to the node

degree in the embedding result. We use Fig. 5 to demonstrate this relationship. Fig. 5a shows an embedding result for a 2-connected SFS with wavelength resource consumption of $4 * bw$, where the average degree in the embedding result is 2. In contrast, another embedding result is shown in Fig. 5b, where $5 * bw$ wavelength resources are needed and the average node degree is 2.5. If we use δ_{ave} to denote the average degree in the embedding result, wavelength resource consumption can be represented by $\frac{|N| * \delta_{ave} * bw}{2}$. Thus, reducing wavelength resource consumption is equivalent to lowering the average degree in the embedding result.

Following this logic, we propose k -connected network slicing (KC-NS) in Algorithm 1. KC-NS takes OEN, bw , and k as inputs, and outputs an embedding result $S = (N, L_S)$, where L_S represents the set of fiber links used in S . Line 1 prunes the fiber links without enough wavelength resources, and the pruned graph is denoted by G_p . Lines 2-4 verify whether G_p meets the level of deterministic fault tolerance or not. When G_p meets the level of deterministic fault tolerance, KC-NS first uses the depth-first search (DFS) algorithm to generate a DFS tree Γ to make the embedding result S connected (Line 5). Meanwhile, to achieve a low average degree in S , an index number i ($i \leq k$) is introduced, and Line 6 calls the i -based node degree balancing (i -NDB) process to reserve fiber links for S until it is k -edge-connected. At last, KC-NS returns the embedding result.

Algorithm 1 k -Connected Network Slicing (KC-NS).

Input: OEN, bw, k ;

Output: $S = (N, L_S)$;

1: Prune links without enough wavelengths $\omega_{l_{m,n}} \leq bw$;

2: **If** $\min_{n \in N} \delta_n^{G_p} \leq k$

3: Reject the request;

4: **End If**

5: Use DFS algorithm to traverse G_p , generate a DFS spanning tree Γ , $L_S = L_S \cup L_\Gamma$, and set $i = 2$;

6: $S = i\text{-NDB}(G_p, S, k, i)$;

7: **Return** S ;

Next, we elaborate on how fiber links are reserved by the i -based node degree balancing process in Algorithm 2. Lines 3-10 maintain a node-set Δ_N including the nodes in S whose degree is less than i . When Δ_N is not empty, to make every node in Δ_N have i degrees, fiber links are repeatedly reserved according to two policies: (1) if there exists a neighbor set N_n ($\delta_m \leq i, \forall m \in N_n$) for a node n ($n \in \Delta_N$), a fiber link is then reserved between n and the physical node with the least degree in N_n (Lines 6-7), or (2) if N_n does not exist, a fiber link

Algorithm 2 i -based Node Degree Balancing (i -NDB).

Input: G_p, S, k, i ;

Output: $S = (N, L_S)$;

1: **While** $i \leq k$ **do**

2: **While** $\min_{n \in N} \delta_n^S \leq i$ **do**

3: Create a node-set $\Delta_N \subseteq N$ ($\delta_n^S \leq i, \forall n \in \Delta_N$);

4: **While** $\Delta_N \neq \emptyset$ **do**

5: Pick a physical node n from Δ_N ;

6: **If** there exists a neighbor set N_n ($\delta_m \leq i, \forall m \in N_n$)

7: Connect n to the node m that has the least $\delta_m^{G_p}$,
 $L_S = L_S \cup l_{m,n}$, and $\Delta_N = \Delta_N - \{m, n\}$;

8: **Else** Connect n to its neighbor m that has the highest
 $\delta_m^{G_p} - \delta_m^S$, then $L_S = L_S \cup l_{m,n}$, $\Delta_N = \Delta_N - n$;

9: **End If**

10: **End While**

11: **End While**

12: Identify a node-set Δ_B including endpoints of i -bridges;

13: **While** $\Delta_B \neq \emptyset$ **do**

14: Find a lightpath $p_{m,n}$ ($m, n \in \Delta_B$) to reduce i -bridges
 with the least length;

15: $L_S = L_S \cup (L_S - L_S \cap L_{p_{m,n}})$, and update Δ_B ;

16: **End While**

17: $i = i + 1$;

18: **End While**

19: **Return** S ;

between n and the node m with the highest available degree (i.e., $\delta_m^{G_p} - \delta_m^S$) is reserved, where $\delta_m^{G_p}$ and δ_m^S represent the node degree of m in G_p and S (Line 8). In graph theory, a graph is k -edge-connected if and only if the number of maximum flows between any pair of nodes is no less than k [24]. Here, if the number of maximum flows between two nodes in S is less than i , the set of lightpaths between these two nodes in S is called the i -bridge, and these two nodes are called endpoints of this i -bridge. In Line 12, the algorithm first identifies endpoints of all i -bridges by applying the maximum flow algorithm on every pair of nodes [31], and saves them into a node-set Δ_B . According to Theorem 3, OEN is k -edge-connected ($k \geq k$), and an unreserved lightpath can always be found to make the embedding result i -edge-connected at each iteration. Therefore, Lines 12-15 repeatedly eliminate i -bridges by reserving new lightpaths between their endpoints. At last, the algorithm returns S as the output.

Theorem 5. KC-NS embeds a k -connected SFS onto a shared OEN with at most twice wavelength resource consumption of the optimal solution.

Proof. This theorem uses OPT to represent the optimal wavelength resource consumption of embedding a k -connected SFS onto a shared OEN, and the lower bound of embedding a k -connected SF slice is shown in Eq. (13).

$$\left\lceil \frac{|N| * k}{2} \right\rceil * bw \leq \text{OPT} \quad (13)$$

Since the DFS algorithm spans the given OEN, the length of the generated DFS tree (i.e., number of fiber links) is $|N| - 1$. As

OEN is k -edge-connected, each node has at least k neighbors [24]. When using θ_i to represent the number of fiber links reserved for making the embedding result k -edge-connected at each iteration, Eq. (14) holds. In the worst case, all physical nodes are connected via policy (2) at each iteration (at most $|N-1|$ fiber links are reserved), and no i -bridge will be generated.

$$\theta_i \leq |N-1|, \forall i \in [2, k] \quad (14)$$

After $k-1$ iterations (i.e., from $i = 2$ to $i = k$), the size of L_S is shown in Eq. (15).

$$|L_S| = |N-1| + \sum_{i=2}^k \theta_i \quad (15)$$

When using α to present the approximation ratio, Eq. (16) shows the relationship between the optimal solution and our proposed KC-NS algorithm.

$$\alpha = \frac{|L_S| * bw}{OPT} = \frac{(|N-1| + \sum_{i=2}^k \theta_i) * bw}{OPT} \quad (16)$$

We combine Eq. (16) with Eqs. (13) and (14) to get Eq. (17).

$$\alpha \leq \frac{(|N-1| + \sum_{i=2}^k \theta_i) * bw}{\left\lceil \frac{|N| * k}{2} \right\rceil * bw} \leq \frac{k * |N-1|}{\left\lceil \frac{|N| * k}{2} \right\rceil} \leq 2 \quad (17)$$

Therefore, we proved that the proposed algorithm achieves a 2-approximation when embedding a k -connected SFS onto an OEN, where all nodes have installed a required SF instance. ■

B. k -Connected Function Slicing

The embedding result of KC-NS includes all nodes across the OEN. When some physical nodes are candidates of a required SF slice, the embedding result generated by KC-NS can also accommodate this SF slice with the same level of deterministic fault tolerance. When accommodating such an SF slice, the physical nodes without the required SF instances and associated fiber connections should be removed from the KC-NS's result to save wavelength resources. But, simply removing those physical nodes and fiber connections may break the k -edge-connected feature. To reduce wavelength resource consumption and maintain the k -edge-connected feature, we propose k -connected function slicing (KC-FS) as Algorithm 3.

KC-FS takes the pruned OEN G_p , bw , k , and one required SF v as inputs, and outputs an SF slice embedding result S_v . An initial solution is obtained by applying the KC-NS algorithm as Line 1. In Line 2, the physical nodes that provide the required SF instance are included in a node-set N_v . In Lines 3-7, KC-FS repeatedly deletes lightpaths bypassing physical nodes without the required SF instances. In Lines 8-12, KC-FS repeatedly substitutes a long lightpath with a shorter lightpath, which is link-disjoint with $\{L_{S_v} - L_{p_{m,n}}\}$. At last, in Line 13, KC-FS returns the embedding result of v 's slice.

Specifically, we elaborate on how the KC-FS algorithm can reduce wavelength resource consumption while satisfying the required level of deterministic fault tolerance. We use $L_{p_{m,n}}$ to represent the set of fiber links in a lightpath $p_{m,n}$. For each lightpath $p_{m,n}$ only bypassing physical nodes without the required SF instances, KC-FS determines whether removing $p_{m,n}$ makes physical node m and n violate the level of

Algorithm 3 k -Connected SF Slicing (KC-FS).

Input: G_p , bw , k , v ;

Output: $S_v = (N_v, L_{S_v})$;

1: $S_v = KC-NS(G_p, bw, k)$;

2: Create node-set N_v ($v \in F_n, \forall n \in N_v$);

3: **For** each lightpath $p_{m,n}$ ($m, n \in N_v$) that only bypasses physical nodes in $\{N - N_v\}$ **do**

4: **If** Removing $p_{m,n}$ does not make the number of maximum flows between m and n less than k

5: $L_{S_v} = L_{S_v} - L_{p_{m,n}}$;

6: **End If**

7: **End For**

8: **For** each pair of physical nodes $m, n \in N_v$ **do**

9: **If** there is a lightpath $p'_{m,n}$ that has a shorter length, than $p_{m,n}$ in S_v and $L_{p'_{m,n}} \cap \{L_{S_v} - L_{p_{m,n}}\} = \emptyset$

10: $L_{S_v} = \{L_{S_v} - L_{p_{m,n}}\} \cup L_{p'_{m,n}}$;

11: **End If**

12: **End For**

13: **Return** S_v ;

deterministic fault tolerance or not (Lines 3-4). If not, $p_{m,n}$ is removable and will be eliminated from L_{S_v} (Lines 5-7). After eliminating the removable light paths, long lightpath connections may exist in S_v . For each lightpath $p_{m,n}$ in S_v , Lines 8-12 check whether there exists a shorter lightpath $p'_{m,n}$ compared to $p_{m,n}$ or not. If yes, $p'_{m,n}$ substitutes $p_{m,n}$ to save wavelength resources. Note that $L_{p'_{m,n}}$ must be link-disjoint to $\{L_{S_v} - L_{p_{m,n}}\}$ such that fault tolerance can be satisfied.

C. k -Connected SF Slices Layered Graph Composition and Embedding

Intuitively, we can compose and embed the k -connected SF slices layered graph by (1) independently running KC-FS on every required SF, and (2) finding k link-disjoint paths to connect their embedding results for each two adjacent SFs in SFC. But, we find that additional wavelength resources might be introduced when independently optimizing each SF slice embedding process. This is because there is only one activated SFP, and different SFS embedding results can share usages of fiber connections to reduce wavelength consumption as they may have common physical candidates. We use Fig. 6 to show this finding via two independently embedded SF slices with $k = 2$. In Fig. 6, curly brackets beside each physical node indicate the SF instances, while the blue-dashed and red-dotted links are the fiber links reserved for slices of f_1 and f_2 . To independently minimize wavelength resource consumption, fiber links $l_{A,C}$, $l_{A,B}$, $l_{B,D}$, and $l_{C,D}$ are employed to connect SF instances of f_1 in Fig. 6a, while $l_{A,C}$, $l_{A,D}$ and $l_{C,D}$ are employed

Fig. 6: Impact of independently embedding SF slices.

for connecting instances for f_2 . In this way, $5 * bw$ wavelength resources are reserved. The candidates for f_2 are the subset of the candidates for f_1 (i.e., f_2 and f_1 have physical candidates in common). When directly employing the embedding result of f_1 's slice to accommodate f_2 , there is no need to reserve $l_{A,D}$. In this way, only $4 * bw$ wavelength resources are needed.

Generally, an SFC includes more than one SFs, so multiple SF slices need to be accommodated onto the OEN. We specify the required SF slices into two types: the embedded SF slices and the unembedded ones. We now discuss what if there exist uncommon candidates between the unembedded and embedded SF slices. In graph theory, when a new vertex joins a k -edge-connected graph, to maintain the k -edge-connectivity, k edge-disjoint paths are needed to connect this new vertex to the graph [24]. Similarly, when incorporating an uncommon physical candidate into the embedding result, k link-disjoint lightpaths are required. For an unembedded SF slice, this reduces wavelength resource consumption by connecting the uncommon candidate to the common ones through reserved fiber connections. We use Fig. 7 to show this benefit, where the blue-dashed links are already reserved by slices of f_1 and f_2 in Fig. 6 and the red-dotted links are newly reserved by f_3 's slice. In Fig. 7a, nodes E and F host an instance for f_3 . When independently composing and embedding f_3 's slice while ensuring the 2-fault tolerance, l_{AE} , l_{AF} , l_{FG} , and l_{GE} are reserved, where $3 * bw$ wavelength resources are required. However, when taking the embedded SF slice into account, only $l_{A,F}$ and $l_{C,F}$ need to be reserved. This is because (1) physical node E lies on the lightpath of $A \leftrightarrow E \leftrightarrow B$ that other SF slices have reserved, and (2) physical node F needs to connect with the embedded SF slices (i.e., A, B, C, and D) via two link-disjoint lightpaths (i.e., $l_{A,F}$ and $l_{C,F}$). The embedding result with $2 * bw$ wavelength resources is shown in Fig. 7b.

Fig. 7: Embedding an SF slice with uncommon physical candidates.

As one can see, sharing usages of fiber connections among the required SF slices can save wavelength resources, and the size of the first embedded SF slice matters. This is because the larger the size of the first embedded SF slice is, the higher probability that the connections among the candidates of the embedded slice can be effectively shared with the unembedded ones. To incorporate the above findings, we propose the most candidate first SF slices layered graph embedding (MCF-SE) algorithm as Algorithm 4. The MCF-SE algorithm takes the pruned OEN and USR as inputs, and outputs a k -connected SF slices layered graph (KC-SFG). The algorithm selects the SF v' with the most instances in the pruned OEN (Line 1). Then, it employs the KC-FS algorithm to compose and embed the SF slice for v' over the OEN, where $N_{v'}$ and $L_{S_{v'}}$ represent the

Algorithm 4 Most candidate first SF slices layered graph embedding (MCF-SE).

Input: $G_p, USR = (V, SFC, k, bw)$;
Output: $KC-SFG$;
1: Select the SF v' with the most instances in the OEN;
2: $S_{v'} = KC-FS(G_p, bw, k, v')$;
3: **For** each SF v in $\{V - v'\}$ **do**
4: Initialize an empty lightpath set $\mathcal{P}$;
5: **For** each physical node n in $\{N_v - (N_v \cap N_{v'})\}$ **do**
6: Connect n to the nearest k nodes in $N_{v'}$ via k link-disjoint lightpaths $P_{m,n}$ ($P_{m,n} \cap p_{a,b} = \emptyset, \forall p_{m,n} \in P_{m,n}, \forall p_{a,b} \in \mathcal{P}$);
7: $N_{v'} = N_{v'} \cup n, \mathcal{P} = \mathcal{P} \cup \{P_{m,n} - (P_{m,n} \cap \mathcal{P})\}$;
8: **End For**
9: $L_{S_{v'}} = L_{S_{v'}} \cup \{L_{\mathcal{P}} - (L_{S_{v'}} \cap L_{\mathcal{P}})\}$;
10: **End For**
11: **For** each lightpath $p_{m,n}$ ($m, n \in N_v$) **do**
12: **If** Removing $p_{m,n}$ does not lead to $\delta_m^{S_v} < k$ and $\delta_n^{S_v} < k$
13: $L_{S_{v'}} = L_{S_{v'}} - L_{p_{m,n}}$;
14: **End If**
15: **End For**
16: $KC-SFG = \{N_{v'}, L_{S_{v'}}\}$;
17: **Return** $KC-SFG$;

physical candidates and the fiber links used for hosting this slice (Lines 2). Next, the algorithm creates an empty lightpath set $\mathcal{P}$ to avoid a fiber link being reserved more than once by the same SF slice. For one SF v other than v' , the algorithm checks whether uncommon candidates exist between v and v' (Line 3-5). If they do exist, for each uncommon candidate between v and v' , the algorithm finds k link-disjoint lightpaths to connect it to $S_{v'}$ (Line 6). These link-disjoint lightpaths are included in a set $P_{m,n}$ and must be link-disjoint from the lightpaths in $\mathcal{P}$. Then, this physical candidate is added to $N_{v'}$, and $P_{m,n}$ is updated in $\mathcal{P}$ (Line 7). When all physical candidates of v are included in $N_{v'}$, the newly-reserved fiber links are added to $L_{S_{v'}}$ (Line 9). The algorithm repeats Lines 4-9 until all SF slices are accommodated. Lines 11-15 eliminate lightpaths associated with the over-protected physical candidates while not violating the deterministic fault tolerance. At last, this algorithm returns the KC-SLG composed of $N_{v'}$ and $L_{S_{v'}}$.

We employ Fig. 8 to demonstrate how the proposed MCF-SE algorithm works. Fig. 8a shows an example of an OEN, where the curly bracket beside the physical node shows the instances installed on this node. There is an SFC request with $k = 2$, which starts at node C, terminates at node F, and demands f_1 and f_2 . In Fig. 8, dark links represent connections in OEN, while red-dotted and blue-dashed links are newly added to $L_{S_{v'}}$ and exist in $L_{S_{v'}}$, respectively. Taking the OEN in Fig. 8a and USR as inputs, MCF-SE generates a 2-edge-connected spanning subgraph (red-dotted links) of the OEN as Fig. 8b. Next, the MCF-SE algorithm applies KC-FS to generate the embedding result for f_1 's slice by (1) eliminating unnecessary lightpath $C \leftrightarrow A \leftrightarrow B \leftrightarrow D$, and (2) substituting the long lightpath $E \leftrightarrow G \leftrightarrow F$ with the shorter lightpath $E \leftrightarrow F$ as shown in Fig. 8c. To accommodate f_2 slice onto the OEN while

Note that the KC-SFG created by the MCF-SE algorithm is a k -edge-connected graph. Supported by Theorem 3, a k -edge-connected subgraph must exist for any SF slice, and k link-disjoint lightpaths must exist for any routing edge set. Based on Theorem 2 and Lemma 5, an available SFP can always be found by running SFP construction algorithms such as [32][33] after at most $k-1$ fiber link(s) and $k-1$ node(s) failures occur.

VII. EXPERIMENTAL RESULTS AND ANALYSIS

In this section, we implement the proposed schemes on diverse networking scenarios to evaluate their performances.

A. Simulation Settings

Based on [14], [25]–[27], OEN can be composed of a main ring and multiple sub-rings with dense connectivity, and we employ three network structures to perform our evaluations, (1) 2-D n -Torus network, (2) 4-degree n -ring network, and (3) $\mathbb{k}$ -edge-connected random network ($\mathbb{k} \geq 4$). According to [34], the size of a 2-D n -Torus network is n^2 . Fig. 9a shows an example of a 2-D 4-Torus network. With wrap-around links (red-dashed links in Fig. 9a), there is a ring in every line and column, while every node has a degree of 4. The size of a 4-degree n -ring is $4 * n$. Fig. 9b shows an example of such a network, where every ring in the dark represents a main ring (e.g., a ring of 13, 14, 15, 16), while every ring with red-dashed

curves is a sub-ring (e.g., a ring of 1, 5, 13, 12). The network in Fig. 9b comprises 4 sub-rings. It is worth noting that both network structures in Fig. 9 are scalable based on the number n .

If not otherwise specified, in an OEN, each link provides [100, 200] wavelengths, each edge server hosts [1, 4] types of SF instances, and each SF has [4, 16] physical candidates. Specifically, in a random network, the minimum degree is set in the range of [4, 7]. We generate 1000 copies of the 2-D 4-Torus network, 4-degree 4-ring network, and 12-,16-,20-,24-nodes random networks within the above settings. For a reliable SFC request (USR), the length of the SFC is set in [2, 10], the level of deterministic fault tolerance (k) is set in [2, 4], while the number of demanded wavelength resources is set in [1, 4]. Meanwhile, the number of USR is set to 100. To evaluate the survivability of the embedded requests, we randomly generate [2, 12] mixed-types failures (including server failure, link failure, and SF instance failure) in the OEN.

B. Evaluation Metrics

We evaluate the performance of the proposed schemes in terms of the following metrics:

Wavelength Resource Consumption: We use $cost_\beta$ to represent wavelength resource consumption of scheme β .

Survivability: We use δ_β to represent the ratio between the number of ongoing services deployed by the scheme α after failures and the total number of services.

Cost-Protection Ratio: Intuitively, a higher deterministic-fault-tolerant requires more wavelength resource costs. Cost-protection ratio (CPR) is introduced as Eq. (18) to measure the efficiency of diverse deterministic fault tolerances:

$$\text{CPR}_\beta = \frac{\text{cost}_\beta}{k} \quad (18)$$

Cost-Survivability Ratio: The efficiency of a scheme depends on both wavelength resource consumption and the survivability of the services when multiple and diverse failures coincide. Cost-survivability ratio (CSR) is then introduced in Eq. (19) as the efficiency of the proposed schemes:

$$\text{CSR}_\beta = \frac{\text{cost}_\beta}{\mathcal{S}_\beta} \quad (19)$$

Approximation Ratio: We use approximation ratio (AR) to evaluate the gap between the performance of the proposed KC-NS and the optimal solution.

Moreover, we extend and implement the exhaustive search algorithm, primary and backup service function chaining and embedding (PB-SFCE) [22], and protection centrality-based pairwise node protections (PC-PNP) [23] for comparisons. To verify the approximate performance of KC-NS, we implement the exhaustive search algorithm to obtain the optimal solution. For PB-SFCE, we extend it by (1) generating one primary SFP and saving it in a set ϕ , (2) finding an SFP link-disjoint to any SFP in ϕ , and adding it to ϕ , and (3) repeating step (2) until the size of ϕ is k . We name this method “ k -PB-SFCE”. For PC-PNP, we extend it by (1) deploying the required SFC via state-of-the-art [32][33], and (2) for each primary SF instance, finding the $k-1$ nearest backup instances and connecting the primary instance to the backup instances via the shortest path.

Fig. 10: Network size vs. wavelength consumption with diverse deterministic-fault-tolerant requirements.

C. Approximation Analysis for KC-NS

When every physical node provides one instance for the required SF, we compare the performance of the proposed KC-NS algorithm to the exhaustive search approach to verify its 2-approximation. In Figs. 10 and 11, blue and red bars represent wavelength resource consumption of KC-NS and exhaustive search as indicated by the left y-axis. The yellow-solid curve is the approximation ratio between these two approaches, labeled by the right y-axis.

In Fig. 10, both KC-NS and exhaustive search need more wavelength resources when increasing the network size. The larger the network is, the more wavelength resources are reserved to achieve the k -fault tolerance. As one can see from Figs. 10a, 10b, and 10c, the performance of KC-NS is near the optimal result when the network size is small, and the approximation ratio increases with the network size. The DFS process likely generates a tree with fewer leaves in a small network than in a large network. When converting the DFS tree to a 2-edge-connected spanning subgraph (i.e., Lines 2-11 in Algorithm 2), a tree with fewer leaves likely generates a 2-edge-connected spanning subgraph with a relatively low average degree. Since wavelength resource consumption of KC-NS is strongly dependent on the average degree in the embedding result, the performance of KC-NS is near-optimal when the network size is small. The DFS likely generates a tree with more leaves in a large network, so the approximation ratio increases with the network size. Moreover, when increasing the k value, the KC-NS algorithm can achieve a lower approximation ratio even though more wavelength resources are needed. A higher k value means more iterations are needed by the i -based degree balancing process, thus reducing the average node degree in the embedding result. Again, as wavelength resource consumption is positively related to the average degree, a higher k value makes the performance of KC-NS closer to the optimal result and leads to a lower approximation ratio. Numerically, when varying the network size, the average approximation ratio that the KC-NS algorithm can achieve with $k = 2, 3$, and 4 is 1.16, 1.12, and 1.09, respectively.

In Fig. 11, when increasing the network connectivity, KC-NS needs fewer wavelength resources to achieve k -fault tolerance while the performance of exhaustive search remains the same. For a specified network, the number of links in a k -edge-connected spanning subgraph is fixed. Increasing the connectivity does not change the number of links in the optimal

Fig. 11: Network connectivity vs. wavelength consumption.

k -edge-connected spanning subgraph. Thus, the performance of exhaustive search remains the same. For KC-NS, when increasing the connectivity, more physical candidates are processed via policy (1) (i.e., Line 6-7 in Algorithm 2), and the average degree in the embedding result reduces. Therefore, increasing the connectivity of the network benefits the performance of the KC-NS algorithm. Similarly, the approximation ratio decreases when k rises. Numerically, when varying network connectivity, the average approximation ratio of KC-NS with $k = 2$ and 3 is 1.25 and 1.23, respectively.

According to the above experimental results and analysis, we find that (1) the proposed KC-NS algorithm achieves near-optimal performance when the network size is small, (2) increasing k value benefits the approximation ratio of the KC-NS algorithm, (3) increasing network connectivity benefits the performance of the proposed KC-NS algorithm, and (4) the average approximation ratio that the KC-NS algorithm can achieve in our experiments is 1.17.

D. Impact of k Analysis for KC-FS

When some physical nodes installed the instance of the required SF, we evaluate the performance of the proposed KC-FS algorithm regarding different deterministic fault tolerance levels. Fig. 12 estimates the impact of the number of physical candidates on wavelength resource consumption and cost-protection ratio. Fig. 13 evaluates the impact of the number of coincided failures on the survivability and cost-survivability ratio. For both Figs. 12 and 13, the red bar (solid curve), blue bar (dotted curve), and green bar (dashed curve) represent the performance of KC-FS when $k = 2, 3$, and 4 , respectively. Bars are measured by the left y-axis, and curves are measured by the right y-axis. Specifically, k -cost ($k = 2, 3, 4$) represents wavelength resource consumption of KC-FS in both figures,

Fig. 12: Impact of physical candidates for KC-FS.

and k -ratio is the cost-protection ratio and cost-survivability ratio in Figs. 12 and 13, respectively.

In Fig. 12, when increasing the number of physical candidates of the required SF, wavelength resource consumption and the cost-protection ratio increase for KC-FS with all k values. This is because more physical candidates need more wavelength resources for KC-FS to perform k -fault tolerance. Among all cases, KC-FS with $k = 4$ requires the highest wavelength resources because a higher deterministic fault tolerance level comes with higher expenses. Interestingly, KC-FS with $k = 2$ needs the least wavelength resource consumption to perform 2-fault tolerance but has the highest cost-protection ratio. Meanwhile, KC-FS with $k = 3$ achieves the lowest cost-protection ratio. From these results, we find that, the wavelength resource reserved by an embedding result can be specified into two types: (1) making the embedding result connected (i.e., DFS tree), or (2) achieving k -fault tolerance. Regardless of the k value, a particular SF slice needs the same amount of wavelength resources to construct the DFS tree on a given OEN. In other words, given an OEN, wavelength resource consumption in type (1) is fixed for a particular SF slice regardless of the k value. Regarding this, we further transform Eq. (18) as:

$$\text{CPR}_\beta = \frac{\text{cost}_\beta^1}{k} + \frac{\text{cost}_\beta^2}{k} \quad (20)$$

where cost_β^1 and cost_β^2 represent wavelength resource consumption in type (1) and type (2), respectively. As cost_β^1 is fixed, when increasing the k value, KC-FS's cost-protection ratio depends on the variation of wavelength resource consumption in type (2) and the k value itself. With the impact of a small k value, KC-FS with $k = 2$ has the highest cost-protection ratio. Since KC-FS with a relatively low k value first employs short lightpaths and long lightpaths are then employed by KC-FS with a relatively high k value, the variation of wavelength resource consumption in type (2) is positively correlated to the k value. That is, a larger k value leads to a faster increase in cost_β^2 . Thus, with a moderate k value, KC-FS with $k = 3$ achieves the lowest cost-protection ratio.

In Fig. 13, we evaluate the performance of KC-FS with diverse k values in terms of survivability and cost-survivability ratio. Here, we set each fiber link with enough wavelength resources such that all USRs are successfully embedded. When increasing the number of random failures, the survivability of all cases decreases, while the cost-survivability ratio of all cases increases. For survivability, KC-FS with $k = 4$ outperforms the other two cases. For the cost-survivability ratio, KC-FS with $k = 2$ outperforms the other two cases when the number of random failures is low (i.e., from 2 to 6). When the number of random failures exceeds 8, most USRs deployed by KC-FS with $k = 2$ fail, leading to a high cost-survivability ratio for KC-FS with $k = 2$. Even though the USR deployed by KC-FS with $k = 4$ has the highest survivability, its cost-survivability ratio is high due to high wavelength resource consumption. Numerically, the average cost-survivability ratio of KC-FS with $k = 2$, $k = 3$, and $k = 4$ are 29.31, 27.08 and 34.41, respectively.

Fig. 13: Impact of random mixed-types failures for KC-FS.

According to the above results, we find that (1) wavelength resource consumption of employing KC-FS to embed an SF slice is positively correlated to the number of its physical candidates, (2) KC-FS with $k = 3$ has the best cost-protection ratio when k is in the range of [2, 4], and (3) on average, KC-FS with $k = 3$ has the best cost-survivability ratio.

E. Efficiency Evaluation and Analysis for MCF-SE

Based on the above discussions, the KC-FS scheme achieves the best cost-protection ratio and the best cost-survivability ratio on average when $k = 3$. Accordingly, we set $k = 3$ for MCF-SE, k -PB-SFCE, and PC-PNP. Specifically, 3-PB-SFCE creates one primary SFP and two link-disjoint backup SFPs, while each primary SF instance in PC-PNP connects to two nearest backup SF instances via link-disjoint lightpaths. In Figs. 14, 15, and 16, the red bar (circle curve), blue bar (square curve), and green bar (triangle curve) represent the performance of PC-PNP, 3-PB-SFCE, and MCF-SE, respectively. Bars and curves are evaluated by the left y-axis and the right y-axis, respectively.

Fig. 14: Impact of physical candidates for MCF-SE.

In Fig. 14, when increasing the number of physical candidates, wavelength resource consumption of PC-PNP and 3-PB-SFCE decreases. More physical candidates facilitate reducing the length of the SFP, thus consuming fewer wavelength resources for both PC-PNP and 3-PB-SFCE. In contrast, the performance of KC-FS is positively correlated to the number of physical candidates, and MCF-SE needs more

wavelength resources when increasing the number of physical candidates. Due to this, when the number of physical candidates is small (i.e., 4 or 6), MCF-SE needs fewer wavelength resources than the other two schemes. When the number of physical candidates is large (i.e., 10), it needs more wavelength resources. Interestingly, when the number of physical candidates increases from 4 to 6, wavelength resource consumption of MCF-SE increases sharply. Meanwhile, when the number of physical candidates increases from 6 to 10, wavelength resource consumption of MCF-SE increases slowly. More physical candidates increase the probability that different SFs can share the connections among the common physical candidates. Therefore, the more physical candidates exist in the OEN, the more pronounced impact of wavelength resources sharing is. Due to this, when the number of physical candidates for each SF is large, the wavelength resources sharing among the required SF is pronounced, leading to a slow increase in wavelength resource consumption. On the contrary, when the number of physical candidates of each SF is small, the impact of physical candidate sharing is minor. Additional wavelength resources are needed to achieve the deterministic fault tolerance (i.e., Lines 3-10 in Algorithm 4), leading to a sharp increase in wavelength resource consumption. As the acceptance ratio is strongly associated with resource usage, the acceptance ratio of 3-PB-SFCE and PC-PNP increases with the number of physical candidates. At the same time, the acceptance ratio decreases with the number of physical candidates for MCF-SE. Numerically, the average acceptance of PC-PNP, 3-PB-SFCE, and MCF-SE are 75.25%, 90.64%, and 85.74%, while the average wavelength resource consumption for PC-PNP, 3-PB-SFCE, and MCF-SE is 28.75, 24.56, and 25.65, respectively.

Fig. 15: Impact of SFC length for MCF-SE.

In Fig. 15, 3-PB-SFCE and PC-PNP require more wavelength resources to deliver the USR reliably when increasing the SFC length. As a longer SFC probably leads to a longer SFP, 3-PB-SFCE and PC-PNP need more wavelength resources. The performance of MCF-SE is not affected by increasing the length of the SFC and outperforms the other two schemes. In MCF-SE, the SFP is generated from the embedding result of the k -connected SF slices layered graph. Since the wavelength resources are reserved for constructing KC-SLG rather than establishing SFP, the length of SFC does not directly impact the performance of MCF-SE. Also, to reduce wavelength resource consumption, MCF-SE tries to maximize the wavelength resources shared among the required SF slices by (1) embedding the SF slice with the most physical candidates first (Line 1) and (2) removing unnecessary connection(s) (Lines 11-15). As a result, it outperforms the other two schemes

regarding wavelength resource consumption when varying the SFC length. Again, as the acceptance ratio is strongly associated with resource usage, MCF-SE outperforms the other two schemes. Numerically, when increasing the SFC length, the average acceptance ratios for PC-PNP, 3-PB-SFCE, and MCF-SE are 75.75%, 85.25%, and 92.5%, while wavelength resource consumption of PC-PNP, 3-PB-SFCE, and MCF-SE is 29.56, 25.12, and 22.31, respectively.

Fig. 16: Impact of mixed-types failures for MCF-SE.

In Fig. 16, when multiple and diverse failures coincide, we evaluate the performance of all three schemes according to their survivability and cost-survivability ratio. Here, we set each fiber link with enough wavelength resources to host all USRs. Even though MCF-SE requires relatively higher wavelength resources when the number of physical candidates is large compared to PC-PNP and 3-PB-SFCE, it provides the best fault tolerance. Note that the cost-survivability ratio of 3-PB-SFCE is lower than MCF-SE when the number of random failures is two due to a lower average wavelength consumption. When further increasing the number of random failures, with the best survivability and moderate wavelength resource consumption, MCF-SE has the best cost-survivability ratio compared to the other two schemes. Numerically, the average survivability of PC-PNP, 3-PB-SFCE, and MCF-SE are 63.45%, 73.65%, and 86.8%, respectively, while the cost-survivability ratios of PC-PNP, 3-PB-SFCE, and MCF-SE are 47.58, 34.89, and 28.47.

According to the above results, we find that (1) using wavelength resource sharing, MCF-SE can achieve a relatively high average acceptance ratio, and (2) MCF-SE outperforms the approaches directly extended from the existing work regarding survivability and average cost-survivability ratio.

VIII. CONCLUSION

In this work, we studied the problem of delivering a reliable SFC request over a shared optical edge network while tolerating multiple and diverse failures. We defined the deterministic fault tolerance (k) to indicate the number of failures that a deployed service can jointly tolerate, and a k -fault-tolerant SFP can jointly tolerate any $k-1$ node failures and $k-1$ link failures during the delivery process. We formulated a novel problem called deterministic-fault-tolerant SFC embedding (DFT-SFCE). To solve this problem, we proposed an efficient heuristic algorithm called most candidate first SF slices layered graph embedding (MCF-SE), which comprises two inter-related techniques: k -connected network slicing (KC-NS) and k -connected function slicing (KC-FS). Through thorough mathematical analysis, we proved that KC-NS is 2-approximate. Our extensive simulation

results verified the efficiency of the proposed schemes and demonstrated that KC-FS yields the best cost-protection ratio and average cost-survivability ratio with $k = 3$. In addition, we showed that the proposed MCF-SE achieves better survivability and average cost-survivability ratio compared to the approaches that are directly extended from existing work.

REFERENCES

- [1] H. Fourati, R. Maaloul, L. Chaari, and M. Jmaiel, "Comprehensive survey on self-organizing cellular network approaches applied to 5G networks," *Comput. Networks*, vol. 199, pp. 108435–108458, 2021.
- [2] 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. Networks*, vol. 167, 2020.
- [3] I. Farris, T. Taleb, Y. Khettab, and J. Song, "A survey on emerging SDN and NFV security mechanisms for IoT systems," *IEEE Commun. Surv. Tutorials*, vol. 21, no. 1, pp. 812–837, 2019.
- [4] "ETSI GS NFV-MAN 001: Network functions virtualization (NFV); management and orchestration," v. 1.1.1, Dec. 2014.
- [5] J. M. Halpern and C. Pignataro, "Service function chaining (SFC) architecture," *RFC*, vol. 7665, pp. 1–32, 2015.
- [6] P. Quinn and T. D. Nadeau, "Problem statement for service function chaining," *RFC*, vol. 7498, pp. 1–13, 2015.
- [7] A. Cockcroft, "A closer look at the Christmas Eve outage," techblog.netflix.com/2012/12/a-closer-look-at-christmas-eve-outage.html, 2012.
- [8] R. Potharaju and N. Jain, "Demystifying the dark side of the middle: a field study of middlebox failures in data centers," in *Proc. ACM IMC* 2013, pp. 9–22.
- [9] The AWS Team, "Summary of the October 22, 2012 AWS service event in the US-East region," <https://aws.amazon.com/message/680342/>, 2012.
- [10] The Google APP Team, "Data center outages generate big losses," http://static.googleusercontent.com/external_content/untrusted_dlcp/www.google.com/en/us/appsstatus/ir/plibxfjh8whr44h.pdf, 2012.
- [11] Y. Hirano, F. He, T. Sato, and E. Oki, "Backup network design against multiple link failures to avoid link capacity overestimation," *IEEE Trans. Netw. Serv. Manag.*, vol. 17, no. 2, pp. 1254–1267, 2020.
- [12] S. Petale and J. Thangaraj, "Link failure recovery mechanism in software defined networks," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 7, pp. 1285–1292, 2020.
- [13] N. Shahriar, S. Taeb, S. R. Chowdhury, M. Zulfikar, 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. Serv. Manag.*, vol. 17, no. 3, pp. 1418–1431, 2020.
- [14] 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, 2020.
- [15] A. Marotta, D. Cassioli, M. Tornatore, Y. Hirota, Y. Awaji, and B. Mukherjee, "Reliable slicing with isolation in optical metro aggregation networks," in *Proc. OFC*, 2020, pp. 1–3.
- [16] F. He and E. Oki, "Backup allocation model with probabilistic protection for virtual networks against multiple facility node failures," *IEEE Trans. Netw. Serv. Manag.*, vol. 18, no. 3, pp. 2943–2959, 2021.
- [17] Y. Fouquet, D. Nace, M. Pi'oro, and M. Poss, "An optimization framework for traffic restoration in optical wireless networks with partial link failures," *Opt. Switch. Netw.*, vol. 23, pp. 108–117, 2017.
- [18] F. He, T. Sato, B. C. Chatterjee, T. Kurimoto, U. Shigeo, and E. Oki, "Robust optimization model for primary and backup resource allocation in cloud providers," *IEEE Tran. on Cloud Comput.*, Early Access, 2021.
- [19] X. Liu and D. Medhi, "Optimally selecting standby virtual routers for node failures in a virtual network environment," *IEEE Trans. Netw. Serv. Manag.*, vol. 14, no. 2, pp. 275–288, 2017.
- [20] M. Ghaznavi, E. Jalalpour, B. Wong, R. Boutaba, and A. J. Mashtizadeh, "Fault tolerant service function chaining," in *Proc. ACM SIGCOMM*, 2020, pp. 198–210.
- [21] R. Yu, G. Xue, and X. Zhang, "Qos-aware and reliable traffic steering for service function chaining in mobile networks," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 11, pp. 2522–2531, 2017.
- [22] P. K. Thiruvassagam, A. Chakraborty, A. Mathew, and C. S. R. Murthy, "Reliable placement of service function chains and virtual monitoring functions with minimal cost in softwarized 5G networks," *IEEE Trans. Netw. Serv. Manag.*, vol. 18, no. 2, pp. 1491–1507, 2021.
- [23] C. Peng, D. Zheng, S. Philip, and X. Cao, "Latency-bounded off-site virtual node protection in NFV," *IEEE Trans. Netw. Serv. Manag.*, vol. 18, no. 3, pp. 2545–2556, 2021.
- [24] R. Diestel, *Graph Theory*. Springer, 2012, vol. 173.
- [25] 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, 2021.
- [26] C. Li, H. Yang, Q. Yao, Z. Sun, and J. Zhang, "High-precision edge cloud collaboration with federated learning in edge optical network," in *Proc. OFC* 2021, pp. 1–3.
- [27] 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. ECOC* 2020, pp. 1–4.
- [28] J. Cheriyan and R. Thurimella, "Approximating minimum-size k-connected spanning subgraphs via matching," *SIAM Journal on Computing*, vol. 30, no. 2, pp. 528–560, 2000.
- [29] G. Robins and A. Zelikovsky, "Improved Steiner tree approximation in graphs," in *Proc. SODA* 2000, pp. 770–779.
- [30] M. Andrews, J. Chuzhoy, V. Guruswami, S. Khanna, K. Talwar, and L. Zhang, "Inapproximability of edge-disjoint paths and low congestion routing on undirected graphs," *Comb.*, vol. 30, no. 5, pp. 485–520, 2010.
- [31] V. King, S. Rao, and R. Tarjan, "A faster deterministic maximum flow algorithm," *Elsevier J. Algorithms*, vol. 17, no. 3, pp. 447–474, 1994.
- [32] 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 INFOCOM* 2020, pp. 1–10.
- [33] 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, 2021.
- [34] J. Lin, G. Shen, Z. Zhai, D. Zheng, Y. Li, Z. Chang, L. Zong, and T. Chang, "Delivering distributed machine learning services in all-optical data center networks with torus topology," in *Proc. ACP* 2021, pp. 1–3.
- [35] "ETSI GR F5G 002: Fifth generation fixed network (F5G); F5G use cases release #1," v. 1.1.1, Feb. 2021.
- [36] J. Fan, Z. Ye, C. Guan, X. Gao, K. Ren, and C. Qiao, "GREP: Guaranteeing reliability with enhanced protection in NFV," in *proc. ACM SIGCOMM workshop 2015*, pp. 13–18.