2022 IEEE 19th International Conference on Mobile Ad Hoc and Smart Systems (MASS) | 978-1-6654-7180-0/22/$31.00 ©2022 IEEE | DOI: 10.1109/MASS56207.2022.00052

SFC-Enabled Reliable Service Provisioning in Mobile Edge Computing via Digital Twins

Jing Li[†], Song Guo[†], Weifa Liang[‡], Quan Chen [§], Zichuan Xu[¶], and Wenzheng Xu[%]
† The Hong Kong Polytechnic University, Hong Kong, P. R. China
‡ City University of Hong Kong, Hong Kong, P. R. China
§ Guangdong University of Technology, Guangzhou, 510006, P. R. China
¶ Dalian University of Technology, Dalian, 116020, P. R. China
% Sichuan University, Chengdu, 610000, P. R. China
Emails: jing5li@polyu.edu.hk, song.guo@polyu.edu.hk, weifa.liang@cityu.edu.hk,
quan.c@gdut.edu.cn, z.xu@dlut.edu.cn, wenzheng.xu@scu.edu.cn

***Abstract*—The Mobile Edge Computing (MEC) paradigm emerges as a promising technology to provide services for various mobile applications at edges of core networks while meeting stringent service delay requirements of users. Orthogonal with the MEC, Network Function Virtualization (NFV) provides the network resource management with flexibility and scalability, where Virtual Network Functions (VNFs) are deployed over edge servers in a chained manner as Service Function Chains (SFCs) for enabling service applications. Provisioning reliable SFC-enabled services in MEC thus is fundamentally important. However, the VNF instances deployed usually are not reliable and affected by multiple factors, including the software implementation, the request execution duration, and so on. Empowered by digital twin techniques that can maintain the states of VNF instances by digital twins in a real-time manner and predict the reliability of VNF instances in edge servers, in this paper we study SFC-enabled reliable service provisioning in an MEC network by exploring the dynamics of VNF instance placement reliability. We first formulate a novel SFC-enabled reliable service problem in MEC networks – the online throughput maximization problem, and show its NP-hardness. We then propose an Integer Linear Programming (ILP) solution to the offline version of the problem, and develop an online algorithm with a provable competitive ratio for the problem. We finally evaluated the performance of the proposed algorithm through experimental simulations, and the results demonstrate that the proposed algorithm is promising.**

I. Introduction

Facilitated by the advance of telecommunication technologies, Mobile Edge Computing (MEC) is envisioned as a complement of cloud computing to provide delay-sensitive services to end users at edges of core networks with small-scale cloud computing capabilities, thereby curtailing the service delay and alleviating the core network utilization [11]–[13], [20]. The Network Function Virtualization (NFV) technique, as an appealing solution in MEC, now is prevalent in managing network resources with flexibility and scalability, through placing Virtual Network Functions (VNFs) over edge servers (cloudlets) to detach network functions from underlying dedicated hardware [22], and chain deployed VNFs as a Service Function Chain (SFC) for various applications [14], [24].

Provisioning reliable SFC-enabled services in MEC is an important challenge for 5G and beyond 5G wireless networks [7]. Notably, the failure of any constituent VNF instance will render its SFC service invalid, while the reliability of the SFC service becomes worse with a long chain length [24]. To enhance the reliability of an SFC-enabled service, adopting redundancy has become a mainstream method, where multiple backup VNF instances are mapped to a primary VNF instance for swift recovery, and these backup ones usually remain in a standby mode until the failure of the primary one [1]. Most existing studies assumed that the reliability of each VNF is given in advance and fixed when deployed into an MEC network [1], [5], [7], [9], [10], [15]–[17]. However, the most unperceived issue on existing reliable service provisioning in MEC is the reliability dynamics of VNF instances, which dramatically downgrades the performance of SFC services [7].

Recently, the emerging digital twin technique offers a promising way to capture the dynamics of the system and furnish future insights by integrating data analytics and machine learning methods [8]. The digital twin applications have received increasing attention for achieving sustainability with accurate failure prediction in a plethora of areas, such as industrial maintenance [18], aeronautics and astronautics [21], as well as in reliable service provisioning in MEC. Especially, each VNF instance is mirrored by a digital twin in an MEC network, which is the virtual representation of the VNF instance [4]. Digital twins can grasp the states of VNF instances running in an MEC network in real-time, by continuously monitoring and creating vivid virtual simulation scenarios [8]. Executing simulation tests on digital twins will indicate potential failures of VNF instances, thereby enabling accurate estimation of the reliability dynamics of VNF instances [4].

In this paper, we study SFC-enabled reliable service provisioning in an MEC network, where the digital twin technique is adopted for real-time reliability prediction of VNF instance placements for user service request admissions. To ensure each reliable SFC-enabled service provisioning in MEC networks, it poses several challenges. One is how to deploy backup VNFs for primary VNF instances of each user request to minimize its computing resource consumption, while meeting the service reliability requirement. Another challenge is how

2155-6814/22/$31.00 ©2022 IEEE
DOI 10.1109/MASS56207.2022.00052

to determine the deployments of both primary and backup VNF instances over cloudlets for each user request, as the computing resource of each cloudlet in an MEC network is limited. Finally, due to the diversity and dynamics of VNF instance reliability of different user requests, even if two user requests have the same SFC and reliability requirements, they may consume different amounts of computing resource for their reliable service provisioning. Thus, how to maximize the number of user requests admitted for a given finite time horizon, assuming that user service requests arrive one by one without the knowledge of future arrivals. In the following, we will address the mentioned challenges and deal with SFC-enabled reliable service provisioning in an MEC network.

The novelty of this paper lies in that we explore the dynamic reliability issue of VNF instances for different requests within a finite time horizon, and leverage digital twins to provide accurate and personalized reliability prediction of VNF instances for requests. We formulate a novel problem of SFC-enabled reliable service provisioning in an MEC network under the dynamic request arrivals, and for which we propose an online algorithm with a provable competitive ratio.

The main contributions of this paper are given as follows. We explore the reliability dynamics of VNF instance deployments in an MEC network, by leveraging digital twin techniques. We first formulate a novel SFC-enabled service provisioning problem in an MEC network through reliability-aware VNF placement: the online throughput maximization problem, and show that the problem is NP-hard. We then formulate an Integer Linear Programming (ILP) solution to the offline version of the online throughput maximization problem. We then devise an online algorithm with a guaranteed competitive ratio for the online throughout maximization problem at the expense of moderate resource violation, by adopting the primal-dual dynamic updating technique. We finally evaluate the performance of the proposed algorithm for SFC-enabled service provisioning in MEC networks through experimental simulations. Experimental results demonstrate that the proposed algorithm is promising.

The rest of the paper is organized as follows. Section II summarizes the related work on SFC-enabled VNF reliability placement in MEC. Section III introduces notions, notations and the problem definition. Section IV proposes an online algorithm with a provable competitive ratio for the online throughput maximization problem, using the primal-dual dynamic updating technique. Section V evaluates the proposed algorithms empirically, and Section VI concludes the paper.

II. Related work

With the advent of Mobile Edge Computing (MEC) and Network Function Virtualization (NFV) techniques, SFC-enabled reliable service provisioning in MEC networks has attracted lots of attention in past years. For example, Huang *et al.* [5] studied the reliability-aware VNF service provisioning problem in MEC, and devised approximation algorithms for the problem. Ishigaki *et al.* [6] developed a recovery technique by a Deep Reinforcement Learning (DRL) method to deal with random failures in providing VNF services. Li *et al.* [9], [10] proposed online algorithms to provide reliable VNF services in MEC to maximize the accumulative revenue under both on-site and off-site backup schemes, respectively. Several recent studies in the literature investigated reliability-aware network service provisioning with SFC requirements in MEC networks. Alleg *et al.* [1] devised a novel redundancy mechanism, where each VNF instance is split into multiple thinner active instances, and multiple backups are deployed to improve the reliability of the SFC. They formulated a Mixed Integer Linear Programming (MILP) solution to minimize the total cost. Liang *et al.* [15], [16] considered the reliability augmentation problem to maximize the SFC service reliability, and proposed efficient algorithms for the problem. Wang *et al.* [22] investigated uncertain demand levels of VNFs and workload balancing among edge servers in reliable SFC service provisioning in MEC, and devised an online learning algorithm to maximize the average SFC backup hit.

It is noticed that there is only a handful of works that considered the reliability dynamics of SFC-enabled service provisioning in MEC. For example, Karimzadeh-Farshbafan *et al.* [8] formulated a Markov Decision Process (MDP) model to capture the dynamic arrivals and departures of SFC service requests with reliability requirements, and developed a heuristic algorithm to minimize the total request implementation cost while maximizing the number of requests admitted. Shang *et al.* [20] designed a self-adapting scheme, by jointly deploying static and dynamic backups over edge and cloud servers to deal with the dynamic failure of VNFs for minimizing the backup deployment cost. Yang *et al.* [23] proposed a Reinforcement Learning (RL) method to deal with random arrivals of SFC-enabled requests in MEC with the aim to maximize the number of requests admitted while meeting both reliability and delay requirements of admitted requests.

Unlike the aforementioned studies, in this paper we study the SFC-enabled reliable service provisioning in MEC networks under dynamic service request arrivals. We further consider the dynamic reliability of VNF instances by leveraging the digital twin technique to provide accurate and personalized reliability prediction of VNFs for each request.

III. Preliminaries

A. System model

We consider a Mobile Edge Computing (MEC) network, modelled by an undirected graph $G = (V \cup \{v_0\}, E)$, where V is the set of Access Points (APs), v_0 is the remote cloud, and E is the set of links connecting APs. Each AP is co-located with a cloudlet via an optical fiber cable, and the communication delay between them is negligible. For simplicity of notation and readability, we use notation $v \in V$ to represent an AP or its co-located cloudlet interchangeably if no confusion arises. Denote by cap_v the available amount of computing resource of cloudlet $v \in V$. We assume that the remote cloud v_0 has unlimited computing and storage resources.

We further assume that the digital twins of all VNF instances in the MEC network are stored at a remote cloud, which are used for dynamically predicting the reliability of VNFs during the execution of different user requests.

Denote by F the set of different types of VNFs offered by the network service provider in G and $c(f)$ the demanded amount of computing resource of an instance of a VNF $f \in F$.

B. User service requests with service reliability

Let U be the set of user requests. Each request $u \in U$ can be expressed by a tuple $\langle SC_u, R_u \rangle$, where SC_u is the SFC requirement and R_u is its reliability requirement, assuming that $f_{u,i} \in \mathcal{F}$ is the ith VNF in SFC SC_u. We assume that the VNF instance for each function in the SFC for each request is referred to as the *primary VNF instance* of its corresponding function. We also assume that each primary VNF instance may have up to K backups across cloudlets [5], where K (≥ 1) is a given positive integer. Furthermore, we assume that the reliability requirement of each request $u \in U$ can be met by deploying up to K backups for each primary VNF instance.

Due to the reliability dynamics of VNF instances of different requests, we adopt the digital twin technique to provide accurate and personalized prediction for the reliability of VNF instances for each request. Denote by $r_{u,i}$ the predicted reliability of VNF $f_{u,i}$ during the execution of request $u \in U$.

Assuming that the primary VNF instance of VNF $f_{u,i}$ is deployed in a cloudlet with the reliability of $r_{u,i}$, and it has $(L-1)$ backups deployed in cloudlets, respectively. Then the reliability $R(f_{u,i})$ of VNF $f_{u,i}$ is calculated as follows.

$$R(f_{u,i}) = 1 - (1 - r_{u,i})^L, \tag{1}$$

The reliability $R(SC_u)$ of SFC SC_u of user request u can be calculated as follows.

$$R(SC_u) = \prod\nolimits_{f_{u,i} \in SC_u} R(f_{u,i}). \tag{2}$$

To meet the reliability requirement R_u of user request $u \in U$, we have

$$R(SC_u) \geq R_u. \tag{3}$$

C. Problem definition

We assume that there is a sequence of incoming SFC requests arriving one by one without knowledge of future request arrivals, and aim to maximize the number of requests admitted. With predicting the reliability of VNF instances by digital twins, we determine whether to admit an incoming request or not, because a request with a high reliability requirement is likely to consume more computing resource to meet its reliability requirement. If a request is admitted, we then need to deal with the deployment of its primary and backup VNF instances on cloudlets to meet its reliability requirement, subject to the computing capacity on each cloudlet.

Definition 1: Given an MEC network $G = (V \cup \{v_0\}, E)$, a sequence U of SFC requests that arrive one by one without knowledge of future arrivals, and digital twins running in the remote cloud provide reliability prediction of VNF instances for each request in real-time, *the online throughput maximization problem* is to maximize the number of requests admitted, subject to the computing capacity on each cloudlet in G, while meeting the reliability requirements of admitted requests.

Theorem 1: The online throughput maximization problem is NP-hard.

The proof is omitted due to space limitation.

IV. Online algorithm for the online throughput maximization problem

In this section, we study the online throughput maximization problem. We first formulate an Integer Linear Programming (ILP) solution to an offline version of the problem where all requests for the monitoring period are given in advance. We then propose an online algorithm for the online problem with a guaranteed competitive ratio, based on the ILP formulation, by adopting the primal-dual dynamic updating technique [2].

A. ILP formulation

To meet the reliability requirement R_u of user request $u \in U$, we have

$$\prod\nolimits_{f_{u,i} \in SC_u} R(f_{u,i}) \geq R_u, \tag{4}$$

which is equivalent to

$$\sum\nolimits_{f_{u,i} \in SC_u} \log_2 R(f_{u,i}) \geq \log_2 R_u. \tag{5}$$

Recall that we assume each primary VNF instance can have at most K backups, with K (≥ 1) a fixed positive integer. Let $x_{u,i,k,v}$ be a binary variable with $u \in U$, $f_{u,i} \in SC_u, k \in [0, K]$ and $v \in V$, where $(x_{u,i,k,v} = 1)$ indicates that the kth backup of VNF $f_{u,i}$ is deployed in cloudlet v, and the 0th backup is the primary VNF instance; otherwise, $x_{u,i,v,k} = 0$.

From Eq. (1), assuming that there are k (≥ 0) backups for VNF $f_{u,i}$, the reliability $\mathcal{R}(f_{u,i}, k)$ of VNF $f_{u,i}$ with k backups is calculated as follows.

$$\mathcal{R}(f_{u,i}, k) = 1 - (1 - r_{u,i})^{k+1}. \tag{6}$$

We then define a *utility function* $H(u, i, k)$ for the k-th backup of VNF $f_{u,i}$ as follows.

$$H(u,i,k) = \begin{cases} \log_2 \mathcal{R}(f_{u,i}, k) - \log_2 \mathcal{R}(f_{u,i}, k-1) & k \geq 1 \\ \log_2 r_{u,i} & k = 0 \end{cases} \tag{7}$$

It can be seen that $\sum_{k=1}^{k} H(u,i,k) + \log_2 r_{u,i} = \log_2 \mathcal{R}(f_{u,i}, k)$. By Ineq. (5), we have

$$\sum_{f_{u,i} \in SC_u} \sum_{k=1}^{k} H(u,i,k) \geq \log_2 R_u - \log_2 \prod_{f_{u,i} \in SC_u} r_{u,i}, \tag{8}$$

For notation simplicity, we define $H'(u)$ as follows.

$$H'(u) = \log_2 R_u - \sum\nolimits_{f_{u,i} \in SC_u} \log_2 r_{u,i} \tag{9}$$

Then we have the reliability constraint as follows.

$$\sum\nolimits_{f_{u,i} \in SC_u} \sum\nolimits_{k=1}^{k} H(u,i,k) \geq H'(u). \tag{10}$$

Recall that we assume the reliability requirement of a request $u \in U$ can be met by deploying at most K backups for each primary VNF instance of the request. Then we have $\sum_{f_{u,i}\in SC_u}\sum_{k=1}^{K} H(u,i,k) \ge H'(u)$. We further assume that the reliability requirement of request u cannot be met by deploying the primary VNF instances only, i.e., $H'(u) > 0$.

Let z_u be a binary decision variable for each user request $u \in U$, where $z_u = 1$ indicates that u is admitted; otherwise, u is rejected. Denote by **P1** the online throughput maximization problem. An Integer Linear Programming (ILP) formulation for the offline version of problem **P1** is given as follows.

$$\mathbf{P1}: \quad \text{Maximize} \quad \sum\nolimits_{u\in U} z_u, \tag{11}$$

subject to:

$$\text{Eq. (7), Eq. (9),}$$

$$\sum_{f_{u,i}\in SC_u}\sum_{k=1}^{K}\sum_{v\in V} H(u,i,k)\cdot x_{u,i,k,v} \ge H'(u)\cdot z_u, \forall u\in U \tag{12}$$

$$\sum_{u\in U}\sum_{f_{u,i}\in SC_u}\sum_{k=0}^{K} c(f_{u,i})\cdot x_{u,i,k,v} \le cap_v, \quad \forall v \in V \tag{13}$$

$$\sum\nolimits_{v\in V} x_{u,i,0,v} = z_u, \quad \forall u \in U,\ \forall f_{u,i} \in SC_u \tag{14}$$

$$\sum\nolimits_{v\in V} x_{u,i,k,v} \le z_u, \ \forall u \in U, \forall f_{u,i} \in SC_u, \forall k \in [1,K] \tag{15}$$

$$x_{u,i,k,v} \in \{0,1\}, \quad \forall u\in U,\ \forall f_{u,i}\in SC_u,\ \forall k\in[0,K],\ \forall v\in V \tag{16}$$

$$z_u \in \{0,1\}, \quad \forall u \in U, \tag{17}$$

where Constraint (13) is the computing capacity constraint on each cloudlet. If a request $u \in U$ is admitted ($z_u = 1$), Constraint (12) is the reliability requirement constraint shown in Ineq. (10), Constraint (14) ensures that each primary VNF instance of the request is deployed on a cloudlet, and Constraint (15) ensures that each backup of the request is deployed on at most one cloudlet. Otherwise (request u is rejected and $z_u = 0$), we have $x_{u,i,k,v} = 0$, $\forall f_{u,i} \in SC_u$, $\forall k \in [0,K]$, and $\forall v \in V$, by Constraints (12), (14) and (15), i.e., no VNF instance is deployed for any rejected request.

B. Online algorithm

Denote by **P1** the online throughput maximization problem. Let problem **P2** be the relaxed Linear Programming (LP) of the offline version of problem **P1**. Let problem **P3** be the dual of problem **P2**. A solution to the online version of problem **P1** can then be obtained from a feasible solution to problem **P3**, with a provable competitive ratio at the expense of moderate resource capacity violations. Specifically, problem **P2** is obtained by performing the LP relaxation on the offline version of problem **P1**, which is expressed as follows.

$$\mathbf{P2}: \quad \text{Maximize} \quad \sum\nolimits_{u\in U} z_u, \tag{18}$$

subject to:

$$\text{Eq. (7), (9), (12), (13), (14), and (15),}$$

$$z_u \le 1, \quad \forall u \in U \tag{19}$$

$$x_{u,i,k,v} \ge 0,\ \forall u\in U,\ \forall f_{u,i}\in SC_u,\ \forall k\in[0,K],\ \forall v\in V \tag{20}$$

$$z_u \ge 0, \quad \forall u \in U \tag{21}$$

where Constraints (19) and (21) show that the binary decision variable z_u is relaxed to a real number between 0 and 1. By Constraints 14, 15, and (19), we have $x_{u,i,k,v} \le 1$. Then the binary decision variable $x_{u,i,k,v}$ is also relaxed to a real number between 0 and 1 by Constraint 20.

Recall that problem **P3** is the dual of problem **P2**, while the former is expressed as follows.

$$\mathbf{P3}: \quad \text{Minimize} \quad \sum\nolimits_{v\in V} cap_v \cdot \beta_v + \sum\nolimits_{u\in U} \mu_u, \tag{22}$$

subject to:

$$\text{Eq. (7), Eq. (9),}$$

$$H'(u)\cdot\alpha_u - \sum_{f_{u,i}\in SC_u}\sigma_{u,i} - \sum_{f_{u,i}\in SC_u}\sum_{k=1}^{K}\lambda_{u,i,k} + \mu_u - 1 \ge 0, \quad \forall u\in U \tag{23}$$

$$c(f_{u,i})\cdot\beta_v + \sigma_{u,i} \ge 0, \quad \forall u\in U,\ \forall f_{u,i}\in SC_u,\ \forall v\in V \tag{24}$$

$$-H(u,i,k)\cdot\alpha_u + \beta_v\cdot c(f_{u,i}) + \lambda_{u,i,k} \ge 0, \qquad \forall u \in U,\ \forall f_{u,i} \in SC_u, \forall k \in [1,K], \forall v \in V \tag{25}$$

$$\alpha_u \ge 0, \beta_v \ge 0, \lambda_{u,i,k} \ge 0, \mu_u \ge 0, \qquad \forall u \in U,\ \forall f_{u,i} \in SC_u, \forall k \in [1,K], \forall v \in V \tag{26}$$

where $\alpha_u, \beta_v, \sigma_{u,i}, \lambda_{u,i,k}, \mu_u$ are dual variables for Constraints (12), (13), (14), (15) and (19), respectively. Especially, $\sigma_{u,i}$ is unconstrained, while $\alpha_u, \beta_v, \lambda_{u,i,k}, \mu_u$ are non-negative, as shown in Constraint (26).

From Constraint (24), we have

$$\sum_{f_{u,i}\in SC_u}\sigma_{u,i} \ge -\frac{1}{|V|}\cdot\sum_{f_{u,i}\in SC_u} c(f_{u,i})\cdot\sum_{v\in V}\beta_v \tag{27}$$

Combining Constraints (23) and (27), we have

$$\sum_{f_{u,i}\in SC_u}\sum_{k=1}^{K}\lambda_{u,i,k} \le H'(u)\cdot\alpha_u + \frac{1}{|V|}\cdot\sum_{f_{u,i}\in SC_u} c(f_{u,i})\cdot\sum_{v\in V}\beta_v + \mu_u - 1. \tag{28}$$

Because $\lambda_{u,i,k} \ge 0$, we have

$$H'(u)\cdot\alpha_u + \frac{1}{|V|}\cdot\sum_{f_{u,i}\in SC_u} c(f_{u,i})\cdot\sum_{v\in V}\beta_v + \mu_u - 1 \ge 0. \tag{29}$$

Recall that we assume $H'(u) > 0$, i.e., the reliability requirement of any request u cannot be met by only deploying the primary VNF instances, and we have

$$\alpha_u \ge \frac{1 - \frac{1}{|V|}\cdot\sum_{f_{u,i}\in SC_u} c(f_{u,i})\cdot\sum_{v\in V}\beta_v - \mu_u}{H'(u)}. \tag{30}$$

From Constraint (25), we have

$$\sum_{f_{u,i}\in SC_u}\sum_{k=1}^{K}\lambda_{u,i,k} \ge \sum_{f_{u,i}\in SC_u}\sum_{k=1}^{K} H(u,i,k)\cdot\alpha_u - \frac{1}{|V|}\cdot\sum_{f_{u,i}\in SC_u}\sum_{k=1}^{K} c(f_{u,i})\cdot\sum_{v\in V}\beta_v. \tag{31}$$

Combining Ineq. (28) and (31), we have

$$H'(u)\cdot\alpha_u+\frac{1}{|V|}\cdot\sum\nolimits_{f_{u,i}\in SC_u}c(f_{u,i})\cdot\sum\nolimits_{v\in V}\beta_v+\mu_u-1$$
$$\geq\sum_{f_{u,i}\in SC_u}\sum_{k=1}^{K}H(u,i,k)\cdot\alpha_u-\frac{1}{|V|}\cdot\sum_{f_{u,i}\in SC_u}\sum_{k=1}^{K}c(f_{u,i})\cdot\sum_{v\in V}\beta_v. \tag{32}$$

Recall that we assume $\sum_{f_{u,i}\in SC_u}\sum_{k=1}^{K}H(u,i,k)\geq H'(u)$. By Ineq. (30) and (32), we have

$$\mu_u\geq 1-(1+\frac{K\cdot H'(u)}{\sum_{f_{u,i}\in SC_u}\sum_{k=1}^{K}H(u,i,k)})\cdot\frac{\sum_{f_{u,i}\in SC_u}c(f_{u,i})}{|V|}\cdot\sum\nolimits_{v\in V}\beta_v. \tag{33}$$

For notation simplicity, we define a constant $\psi_u\geq 0$ for each user request $u\in U$ as follows.

$$\psi_u=(1+\frac{K\cdot H'(u)}{\sum_{f_{u,i}\in SC_u}\sum_{k=1}^{K}H(u,i,k)})\cdot\frac{\sum_{f_{u,i}\in SC_u}c(f_{u,i})}{|V|}. \tag{34}$$

We then have

$$\mu_u\geq 1-\psi_u\cdot\sum\nolimits_{v\in V}\beta_v. \tag{35}$$

We claim that given the dual variables $\mu_u\geq 0$ and $\beta_v\geq 0$ subject to Ineq. (35), it always exists feasible dual variables α_u, $\lambda_{u,i,k}$ and $\sigma_{u,i}$ to deliver a feasible solution to problem **P3**. This claim will later be shown in Lemma 1. Therefore, we only focus on the two dual variables $\mu_u\geq 0$ and $\beta_v\geq 0$ to devise a feasible solution, considering Ineq. (35). We then update the variables in both the primal and dual problems simultaneously, by adopting the primal-dual dynamic updating technique [2].

The detailed online algorithm is as follows. Dual variables μ_u and β_v are set as 0s initially. Upon the arrival of request u, we need to set the dual variables subject to Ineq (35). Especially, we propose an admission control policy to determine the admission of request u as follows: if $1-\psi_u\cdot\sum_{v\in V}\beta_v\leq 0$, it will be rejected; otherwise, it will be admitted and the dual variable μ_u is updated as follows.

$$\mu_u\leftarrow 1-\psi_u\cdot\sum\nolimits_{v\in V}\beta_v. \tag{36}$$

We now obtain the VNF instance set $\mathbb{F}_u$ for request u to meet its reliability requirement as follows. Initially, $\mathbb{F}_u$ consists of primary VNF instances of request u, i.e., $\mathbb{F}_u\leftarrow SC_u$. Recall that we assume the reliability requirement of each request $u\in U$ can be met by deploying no more than K backups for each primary VNF instance, with $K\geq 1$. Let B_u be the set of all potential backups for request $u\in U$ with $|B_u|=K\cdot|SC_u|$. Let $\gamma_b=\frac{c(b)}{H(b)}$ be the ratio of a backup b, where $c(b)$ is the computing resource consumption of b and $H(b)$ is the utility gain of deploying b by Eq. (7). We then deploy backups in B_u one by one in a non-increasing order of γ_b until meeting the reliability requirement constraint (10). Especially, we first deploy backup b_1. If the reliability requirement of request u can be met by deploying b_1 (i.e., $H(b_1)\geq H'(u)$), we deploy backup b_1 by letting $\mathbb{F}_u\leftarrow\mathbb{F}_u\cup\{b_1\}$ and this is done. Otherwise ($H(b_1)<H'(u)$), we deploy backup b_1 by letting $\mathbb{F}_u\leftarrow\mathbb{F}_u\cup b_1$, and deploy the next backup b_2. The procedure continues until we deploy a backup b_j with $1\leq j\leq|B_u|$ and $\sum_{1\leq j'\leq j}H(b_{j'})\geq H'(u)$, i.e., its reliability requirement is met. We then sort the VNF instances in $\mathbb{F}_u$ in a non-increasing order of their computing resource consumption, and show their deployment on cloudlets as follows. Let $\mathcal{F}_{u,v}$ be the set of VNF instances for request u deployed on cloudlet v with $\cup_{f\in\mathcal{F}_{u,v}}=\mathbb{F}_u$. We set $\mathcal{F}_{u,v}\leftarrow\emptyset$ initially, $\forall v\in V$. To model the usage cost of computing resource in each cloudlet, we define a function $\eta(\beta_v,\mathcal{F}_{u,v})$ as follows.

$$\eta(\beta_v,\mathcal{F}_{u,v})=\beta_v\cdot(1+\frac{\psi_u\cdot\sum_{f\in\mathcal{F}_{u,v}}c(f)}{cap_v\cdot(K+1)\cdot\sum_{f_{u,i}\in SC_u}c(f_{u,i})})+\frac{\psi_u\cdot\sum_{f\in\mathcal{F}_{u,v}}c(f)}{cap_v\cdot(K+1)\cdot\sum_{f_{u,i}\in SC_u}c(f_{u,i})}. \tag{37}$$

For each VNF instance $f\in\mathbb{F}_u$ in the sorted order, we identify the cloudlet $v'\in V$ with the minimum $\eta(\beta_{v'},\mathcal{F}_{u,v'})$, and update $\mathcal{F}_{u,v'}\leftarrow\mathcal{F}_{u,v'}\cup\{f\}$. This procedure continues until all VNF instances in $\mathbb{F}_u$ are deployed.

Having obtained $\{\mathcal{F}_{u,v}\mid\forall v\in V\}$ for request u, we update the dual variable β_v of each cloudlet v as follows.

$$\beta_v\leftarrow\beta_v\cdot(1+\frac{\psi_u\cdot\sum_{f\in\mathcal{F}_{u,v}}c(f)}{cap_v\cdot(K+1)\cdot\sum_{f_{u,i}\in SC_u}c(f_{u,i})})+\frac{\psi_u\cdot\sum_{f\in\mathcal{F}_{u,v}}c(f)}{cap_v\cdot(K+1)\cdot\sum_{f_{u,i}\in SC_u}c(f_{u,i})}. \tag{38}$$

The online algorithm for the online throughput maximization problem (problem **P1**) is detailed in `Algorithm` 1.

C. Algorithm analysis

Lemma 1: (1) Having obtained dual variables $\mu_u\geq 0$ and $\beta_v\geq 0$ subject to Ineq. (35), it always exists feasible dual variables α_u, $\lambda_{u,i,k}$ and $\sigma_{u,i}$ to deliver an feasible solution to problem **P3**; and (2) `Algorithm` 1 delivers a feasible solution for problem **P3**.

The proof is omitted due to space limitation.

Lemma 2: In the solution for problem **P1** delivered by `Algorithm` 1, the violation on the capacity constraint on each cloudlet is upper bounded by Λ, where $\Lambda=\frac{(K+1)\cdot|SC|_{max}\cdot c_{max}}{cap_{min}}\cdot\frac{\ln(\frac{1}{\psi_{min}}\cdot(1+\frac{\psi_{max}}{cap_{min}})+\frac{\psi_{max}}{cap_{min}}+1)}{\ln(1+\frac{\psi_{min}\cdot c_{min}}{(K+1)\cdot cap_{max}\cdot|SC|_{max}\cdot c_{max}})}-1$, with $\psi_{max}=\max_{u\in U}\{\psi_u\}$, $\psi_{min}=\min_{u\in U}\{\psi_u\}$, $|SC|_{max}=\max_{u\in U}\{|SC_u|\}$, $|SC|_{min}=\min_{u\in U}\{|SC_u|\}$, $c_{max}=\max_{f\in F}c(f)$, $c_{min}=\min_{f\in F}c(f)$, $cap_{max}=\max_{v\in V}cap_v$, and $cap_{min}=\min_{v\in V}cap_v$.

The proof is omitted due to space limitation.

Theorem 2: Given an MEC network $G=(V\cup\{v_0\},E)$, a sequence U of SFC requests with reliability requirements, there is an online algorithm, `Algorithm` 1, for the online throughput maximization problem with the competitive ratio of $(1+\psi_{max})$, and the computing capacity violation on any cloudlet is upper bounded by Λ, where $\psi_{max}=\max_{u\in U}\psi_u$

Algorithm 1 An online algorithm for the problem.

```
Input: An MEC network, a sequence $U$ of incoming SFC requests arriving
    one by one without future knowledge, and digital twins running in the
    remote cloud to provide reliability prediction of VNF instances.
Output: An online scheduling of incoming requests with the aim of maximizing
    the number of requests admitted.

$\mu_u \leftarrow 0, \forall u \in U$; $\beta_v \leftarrow 0, \forall v \in V$;
while a request $u$ arrives do
    if $1 - \psi_u \cdot \sum_{v \in V} \beta_v > 0$ then
        Admit request $u$, and update $\mu_u$ by Eq. (36);  $\mathbb{F}_u \leftarrow SC_u$;
        Sort the backups $B_u = \{b_1, b_2, \ldots, b_{|B_u|}\}$ in non-decreasing
            order of ratios $\gamma_{b_j} = \frac{c(b_j)}{H(b_j)}$ with $1 \leq j \leq |B_u|$;
        for each backup $b_j \in B_u$ in sorted order with $1 \leq j \leq |B_u|$ do
            $\mathbb{F}_u \leftarrow \mathbb{F}_u \cup \{b_j\}$;
            if $\sum_{1 \leq j' \leq j} H(b_{j'}) \geq H'(u)$ then
                Break;
            end if
        end for
        Sort the VNF instances in $\mathbb{F}_u$ in non-increasing order of their
            computing resource demands; $\mathcal{F}_{u,v} \leftarrow \emptyset, \forall v \in V$;
        for each VNF instance $f \in \mathbb{F}_u$ in its sorted order do
            Identify cloudlet $v' \in V$ with the minimum $\eta(\beta_{v'}, \mathcal{F}_{u,v'})$;
            $\mathcal{F}_{u,v'} \leftarrow \mathcal{F}_{u,v'} \cup \{f\}$;
        end for
        for each cloudlet $v \in V$ do
            Deploy the VNF instances in $\mathcal{F}_{u,v}$ on cloudlet $v$;
            Update $\beta_v$ by update function (38);
        end for
    else
        Rejecte request $u$;
    end if
end while
```

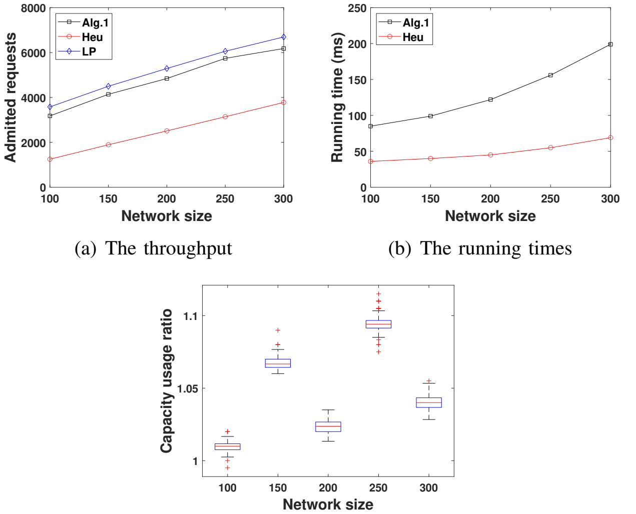
with the constant ψ_u defined in Eq. (34), and Λ is a constant given in Lemma 2. The algorithm takes $O(|SC|_{max} \cdot K \cdot (\log(|SC|_{max} \cdot K) + |V|))$ time to admit each SFC request, where $|SC|_{max}$ is the maximum length of any SFC, K is the maximum number of backups deployed for a primary VNF instance, and V is the cloudlet set.

The proof is omitted due to space limitation.

V. Performance evaluation

A. Experimental environment setting

We consider an MEC network $G = (V \cup \{v_0\}, E)$ consisting of from 100 to 300 APs, where each AP is co-located with a cloudlet. Each MEC network instance is generated by adopting the widely used tool GT-ITM [3]. The computing capacity on each cloudlet is drawn from $4,000$ MHz to $14,000$ MHz randomly. We assume that the network service provider offers 20 different types of VNFs, and the computing resource consumption of a VNF instance is drawn from 20 MHz to 100 MHz, while the reliability of each VNF for each request over time is randomly drawn from 0.8 to 0.9 [24]. There are 30 different SFCs, and the length of each SFC is set between 2 and 6 [24]. We assume that there are $10,000$ incoming requests, and a random SFC from the preset SFCs is selected for each request with its reliability requirement ranging from 0.9 to 0.99 [24]. We then assume that each primary VNF instance can have up to 3 backups, i.e., K is set at 3. The value in each figure is the mean of the results out of 30 MEC instances with the same size. The actual running time of each algorithm is obtained on a desktop with a 3.60 GHz Intel 8-Core i7 CPU and 16 GB RAM. These parameters are adopted as the default settings unless otherwise specified.



(a) The throughput

(b) The running times

(c) The utilization ratios of cloudlets by `Alg.1`

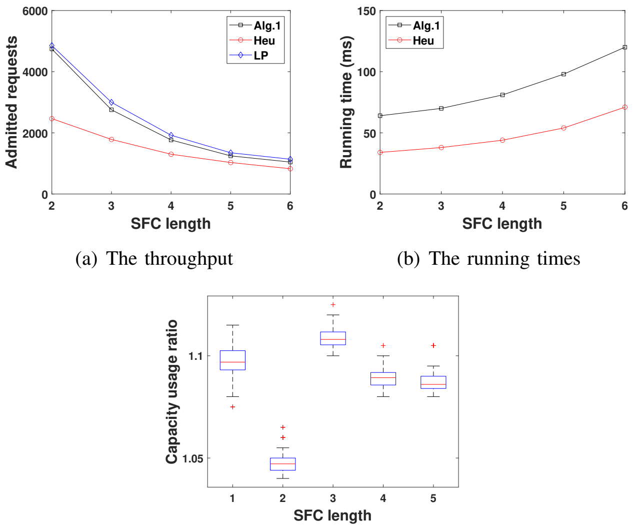
Fig. 1. Performance of different algorithms for the problem.

To evaluate the proposed algorithm `Algorithm` 1, referred to as `Alg.1`, for the online throughput maximization problem, we introduce a heuristic algorithm `Heu` as follows. For each incoming request, algorithm `Heu` tries to admit the request by deploying backups one by one with the largest reliability argumentation of its SFC until meeting its reliability requirement, and then all its primary and backup VNF instances are deployed one by one on cloudlets with sufficient residual computing resource. We also formulate a Linear Programming solution (18) to the offline version of the online throughput maximization problem, referred to as `LP`, which is an upper bound on the optimal solution of the problem.

B. Algorithm performance

We first studied the performance of algorithm `Alg.1` for the online throughput maximization problem against `Heu` and `LP`, by varying the network size from 100 to 300. We define the utilization ratio of a cloudlet v as the ratio of its consumed computing resource to its computing capacity. Fig. 1 shows the throughput and running times of different algorithms, and the utilization ratios of cloudlets by `Alg.1`. Fig. 1(c) depicts the maximum, minimum and average utilization ratios of cloudlets by `Alg.1`, with computing capacity violation on each cloudlet no more than 11.5% of its capacity. From Fig. 1(a), when the network size is 300, algorithm `Alg.1` outperforms algorithm `Heu` by 63.6%, while the performance of algorithm `Alg.1` is 92.3% of `LP`. The rationale behind is that the proposed algorithm `Alg.1` has an efficient admission control policy for each incoming request admission, as well as determining the VNF placements of the SFC of the request efficiently, while meeting the reliability requirement of the request.

We then evaluated the impact of the SFC length on the performance of algorithm `Alg.1` against `Heu` and `LP`, by varying the SFC length from 2 to 6 with the network



(a) The throughput

(b) The running times

(c) The utilization ratios of cloudlets by `Alg.1`

Fig. 2. The impact of the SFC length on the performance of `Alg.1`.

size at 100. Fig. 2 plots the performance curves of different algorithms. From Fig. 2(a), when the SFC length is 6, the performance of algorithm `Alg.1` is 26.4% higher than that of `Heu`, while the performance of algorithm `Alg.1` is 91.8% of that of `LP`. Also, Fig. 2(c) shows the computing capacity violation on each cloudlet by `Alg.1`, which is no more than 12.7% of its capacity. From Fig. 2(a), the performance of algorithm `Alg.1` when the SFC length is 6 is 22.1% of itself when the SFC length is 2. The justification is that a shorter SFC length leads to less computing resource consumption, and more requests can be admitted with a shorter SFC length.

VI. Conclusion

In this paper, we studied a novel online throughput maximization problem of reliable SFC-enabled service provisioning in an MEC network, and showed the NP-hardness of the problem. Specifically, by leveraging the digital twin technique, the reliability of a VNF instance can be dynamically predicted and adopted. Built upon the predicted reliability of each VNF instance, we considered the online throughput maximization problem by formulating an ILP solution to its offline version first, followed by devising an online algorithm with a guaranteed competitive ratio for the problem, at the expense of moderate computing resource violations. We finally evaluated the performance of the proposed algorithm through experimental simulations. Experimental results demonstrate that the proposed algorithm is promising.

References

[1] A. Alleg, T. Ahmed, M. Mosbah, and R. Boutaba. Joint diversity and redundancy for resilient service chain provisioning. *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 7, pp. 1490 – 1504, 2020,

[2] M. X. Goemans and D. P. Williamson. The primal-dual method for approximation algorithms and its application to network design problems. *Book Chapter of Approximation Algorithms for NP-Hard Problems.* pp. 144 – 191, PWS Publishing Company, 1997.

[3] GT-ITM. http://www.cc.gatech.edu/projects/gtitm/, 2019.

[4] B. He and K. Bai. Digital twin-based sustainable intelligent manufacturing: a review. *Advances in Manufacturing*, vol.9, no.1, pp.1–21, 2021.

[5] M. Huang, W. Liang, X. Shen Y. Ma, and H. Kan. Reliability-aware virtualized network function services provisioning in mobile edge computing. *IEEE Transactions on Mobile Computing*, vol. 19, no. 11, pp. 2699 – 2713, 2020.

[6] G. Ishigaki, S. Devic, R. Gour, and J. P. Jue. DeepPR: progressive recovery for interdependent VNFs with deep reinforcement learning. *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 10, pp. 2386 – 2399, 2020.

[7] M. Karimzadeh-Farshbafan, V. Shah-Mansouri, and D. Niyato. A dynamic reliability-aware service placement for network function virtualization (NFV). *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 2, pp. 318 – 333, 2020.

[8] L. U. Khan, W. Saad, D. Niyato, Z. Han, and C. S. Hong. Digital-twin-enabled 6G: vision, architectural trends, and future directions. *IEEE Communications Magazine*, vol. 60, no. 1, pp. 74 – 80, 2022.

[9] J. Li, W. Liang, M. Huang, and X. Jia. Providing reliability-aware virtualized network function services for mobile edge computing. *Proc. of ICDCS'19*, pp. 732 – 741, 2019.

[10] J. Li, W. Liang, M. Huang, and X. Jia. Reliability-aware network service provisioning in mobile edge-cloud networks. *IEEE Transactions on Parallel and Distributed Systems*, vol. 31, no. 7, pp. 1545–1558, 2020.

[11] J. Li, W. Liang, Y. Li, Z. Xu, X. Jia, and S. Guo. Throughput maximization of delay-aware DNN inference in edge computing by exploring DNN model partitioning and inference parallelism. To appear in *IEEE Transactions on Mobile Computing*, 2021, doi: 10.1109/TMC.2021.3125949.

[12] J. Li, W. Liang, W. Xu, Z. Xu, X. Jia, W. Zhou, and J. Zhao. Maximizing user service satisfaction for delay-sensitive IoT applications in edge computing. *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 5, pp. 1199 – 1212, 2022,

[13] J. Li, W. Liang, W. Xu, Z. Xu, Y. Li, and X. Jia Service home identification of multiple-source IoT applications in edge computing. To appear in *IEEE Transactions on Services Computing*, 2022, doi: 10.1109/TSC.2022.3176576.

[14] J. Li, W. Liang, Z. Xu, X. Jia, and W. Zhou. Service provisioning for multi-source IoT applications in mobile edge computing. *ACM Transactions on Sensor Networks*, vol. 18, no. 2, Article 17, pp. 17:1 – 17:25, May, 2022.

[15] W. Liang, Y. Ma, W. Xu, X. Jia, and S. Chau. Reliability augmentation of requests with service function chain requirements in mobile edge-cloud networks. *Proc. of ICPP'20*, ACM, 2020.

[16] W. Liang, Y. Ma, W. Xu, Z. Xu, X. Jia, and W. Zhou. Request reliability augmentation with service function chain requirements in mobile edge computing. To appear in *IEEE Transactions on Mobile Computing*, 2021, doi: 10.1109/TMC.2021.3081681.

[17] S. Lin, W. Liang, and J. Li. Reliability-aware service function chain provisioning in mobile edge-cloud networks. *Proc. of ICCCN'20*, pp. 1 – 9, IEEE, 2020.

[18] S. Mi, Y. Feng, H. Zheng, Y. Wang, Y. Gao, and J. Tan, Prediction maintenance integrated decision-making approach supported by digital twin-driven cooperative awareness and interconnection framework, *Journal of Manufacturing Systems*, vol. 58, pp. 329 – 345, 2021.

[19] T. Öncan. A survey of the generalized assignment problem and its applications. *Information Systems and Operational Research*, vol. 45, no. 3, pp. 123 – 142, 2007.

[20] X. Shang, Y. Huang, Z. Liu, and Y. Yang. Reducing the service function chain backup cost over the edge and cloud by a self-adapting scheme. *IEEE Transactions on Mobile Computing*, vol. 21, no. 8, pp. 2994 – 3008, 2022.

[21] F. Tao, J. Cheng, Q. Qi, M. Zhang, H. Zhang, and F. Sui. Digital twin-driven product design, manufacturing and service with big data. *The International Journal of Advanced Manufacturing Technology*, vol. 94, no. 9, pp. 3563 – 3576, 2018.

[22] C. Wang, Q. Hu, D. Yu, and X. Cheng. Proactive deployment of chain-based VNF backup at the edge using online bandit learning. 2021 *Proc. of ICDCS'21*, IEEE, pp. 740 – 750, 2021.

[23] L. Yang, J. Jia, H. Lin, and J. Cao. Reliable dynamic service chain scheduling in 5G networks. To appear in *IEEE Transactions on Mobile Computing*, 2022, doi: 10.1109/TMC.2022.3157312.

[24] J. Zhang, Z. Wang, C. Peng, L. Zhang, T. Huang, and Y. Liu. Raba: resource-aware backup allocation for a chain of virtual network functions. *Proc. of INFOCOM'19*. pp. 1918 – 1926, IEEE, 2019.