Hindawi
Security and Communication Networks
Volume 2022, Article ID 2151645, 10 pages
https://doi.org/10.1155/2022/2151645

Research Article

Dynamic Detection and Placement for VSFs over Edge Computing Scenarios: An ACO-Based Approach

Chao Bu,[1] Xinyang Zhang,[1] Jianhui Lv ,[2] and Jinsong Wang[1]

[1]*School of Computer Science and Engineering, Tianjin University of Technology, Tianjin 300384, China*
[2]*Pengcheng Laboratory, Shenzhen 518000, China*

Correspondence should be addressed to Jianhui Lv; lvjianhui2012@163.com

Received 21 January 2022; Accepted 22 March 2022; Published 11 April 2022

Academic Editor: Ke Gu

As an extension of cloud computing, the edge computing has become an important pattern to deal with novel service scenarios of the Internet of Everything (IoE), especially for the rapidly increasing different kinds of service requests from edge equipment. It is still a great challenge to satisfy the demands of delay-sensitive applications, so as to optimize the service provision delay for edge equipment under 5G. In this paper, by introducing virtualized service functions (VSFs) into edge computing pattern, the mechanism of service function detection and placement among multiple Edge Computing Servers (ECSs) is proposed. We firstly improve the Ant colony optimization (ACO) method to adapt to the situation that the service requests may frequently change from different edge network domains. Based on the improved ACO, a scheme of searching for the locations (i.e., ECSs) of the requested service functions is devised, so as to optimize the service searching delay. Then, a service function placement scheme is presented, and it deploys most of appropriate service functions in each ECS by predicting the future requested frequencies of functions, which further reduces the overall service provision delay. In addition, it also improves the ECS computing capacity utilization. The simulation results show that the proposed mechanism is feasible and effective.

1. Introduction

The 5th-generation mobile system (5G) is gradually integrating into people's daily life. The requirements of the novel service scenarios brought by the 5G have changed significantly [1]. For example, the delay sensitivity has become one of the most important service demands of edge equipment. Many types of research have been done on providing services mainly by the cloud computing center due to its powerful computing capacity, which enables almost all kinds of virtualized service functions (VSFs) [2, 3] to be placed and performed there. However, the cloud computing servers are usually located far away from most of network edges. With the rapid increasing service requests from mobile equipment that mainly locates at the network edges, the service provision based on the conventional cloud computing may lead to problems such as much higher transmission delay and serious congestion [4]. As an extension of cloud computing, the edge computing deploys the service provision capability near to network edges where most of service requests generate [5, 6]. Its distributed edge computing servers (ECSs) locate much closer to edge equipment so as to adapt to the service provision delay demand.

The 5G is able to support high-speed data transmission, which significantly decreases the service delivery delay of ECSs [7]. However, comparing to the service cluster (SC) in the cloud computing center, each ECS's limited computing capacity means that it can only provide some kinds of services due to the limited number of VSFs placed in it [8]. It is hard for an ECS to satisfy all kinds of service requests for edge equipment, because the corresponding VSFs may have not yet been placed in this ECS when such requests arrive. And the method, by which all such service requests are dealt with by the remote SC or the corresponding VSFs are instantly migrated from the remote SC to this ECS, is obviously unsustainable, especially when the network load is heavy [9]. Considering the fact that other ECSs locating much closer than the SC may have already been placed such

VSFs, the corresponding services can be provided by one of these ECSs. How to effectively find an ECS that currently contains a certain VSF becomes a key issue. In practice, a tremendous amount of different kinds of service requests are constantly generated and delivered, which leave the changing traces of recent service provision information in corresponding equipment [10]. In this paper, the VSF concentration on links is presented by leveraging the above changing traces, based on which an improved ACO-inspired service function detection scheme is devised further to effectively search for the placed locations (i.e., nearby ECSs) of the requested VSFs. It optimizes the concentration updating efficiency to overcome the high delay problem caused by several rounds of iteration of the conventional ACO method.

In addition, an ECS should adapt to the frequent changes in service requests by deploying appropriate VSFs in it in time. It is impossible for an ECS to contain all VSFs due to its limited computing capacity [9]. In fact, a lot of service requests are regional and periodic for an ECS [11]. That is, an ECS may often provide some certain kinds of services due to the working features of its nearby equipment [12]. However, the ECS also may be requested the VSFs that have not yet been placed in it because new applications become popular recently. We take the recent frequencies of the VSFs being performed and requested into account, and the VSF placement scheme is devised in this paper. It enables an ECS to retain the already deployed VSFs with higher being performed frequencies in it and migrate the not yet deployed VSFs with higher being requested frequencies to it, respectively. In this approach, the VSF detection delay and the service delivery delay can be further optimized.

Some types of research have been done on service placement and migration over edge computing scenarios (e.g., [13–16]). In [13], it introduces a framework for optimal placement of service replicas proactively in the 5G edge network. It deploys the multimedia service instances on the trajectory edge nodes by integrating the user's path prediction model, so as to provide an optimal deployment technique that traded off between maximizing the QoE and minimizing the deployment cost. In [14], it studies the container-based service migration problem in edge computing and proposes a service migration mechanism based on mobility awareness. Its service migration mechanism firstly triggers the service migration according to the service density of the current node and then selects the service and the corresponding destination node for the optimization object to minimize the service delay. In [15], it proposes a novel service migration policy method based on deep reinforcement learning and dynamic adaptation in multiaccess edge computing. It innovatively analyzes the different states of edge network quantitatively and applies deep Q-learning to migration methods, which can adjust the learning rate adaptively to implement rapid convergence in the learning process. In [16], it combines prediction mechanism and migration research together to optimize the migration of VNF. It built a system cost evaluation model integrating bandwidth utilization and migration time, and devised a heuristic algorithm to obtain the near-optimal solution. However, they mainly focus on instantly deploying functions on real-time service demands or migrating functions based on long-term iterative learning, which cannot well adapt to the service requests with frequent changes and delay sensitivity.

In this paper, the mechanism of ACO-based VSF detection and placement (AVDP) is proposed, so as to minimize the service provision delay to the edge equipment and optimize the ECS computing capacity utilization. The major contribution and innovations can be summarized as follows. The VSF concentration on links is presented to reflect the frequencies of VSFs being detected recently, and the improved ACO-inspired VSF detection scheme based on the VSF concentration is devised to efficiently search for the placed locations of the requested VSFs. It adapts to the novel service scenarios that the service requests for edge equipment and the deployed locations of VSFs may frequently change under 5G. Furthermore, the scheme of dynamically deploying appropriate VSFs in each ECS is devised, and it places VSFs by predicting the future requested frequencies of VSFs according to the variations of VSFs concentration on links and VSFs requested number in ECSs. Thus, the overall service provision delay is significantly optimized and the ECS computing capacity utilization is efficiently improved.

The remainder of this paper is structured as follows. In Section 2, we present the system framework and workflow of the proposed AVDP. In Section 3, we define the VSF detection ant and the VSF concentration, and devise the scheme of searching for VSFs among multiple ECSs based on them. In Section 4, we devise the scheme of placing appropriate VSFs in suitable ECSs with the ECS computing capacity utilization considered. In Section 5, we present simulation experiments and results. Finally, Section 6 concludes the paper.

2. System Framework

The system framework is shown in Figure 1. The conventional cloud computing center is usually located far away from network edges. It can provide almost all kinds of services due to the powerful computing capacity of the service cluster (SC), and VSFs placed in it can be dispatched to ECSs. ECSs are distributed in end network domains (ENDs) where most of service requests generate to efficiently meet the delay sensitivity demand of edge equipment. The service function pool (SFP) in each ECS only contains some VSFs because of the limited computing capacity. The VSFs already deployed in an ECS can be replaced by other VSFs due to the changing service scenarios in practice, and VSFs can be migrated from the SC or other ECSs. In addition, each switching equipment (SE) updates a table named VSFs detection record (VDR). The VDR is to reflect the current probabilities of successfully finding the locations of the requested VSFs by different next hops, so as to efficiently determine the suitable ECSs that can provide corresponding services.

The overall workflow of the system is shown in Figure 2; here, the number is the action order.

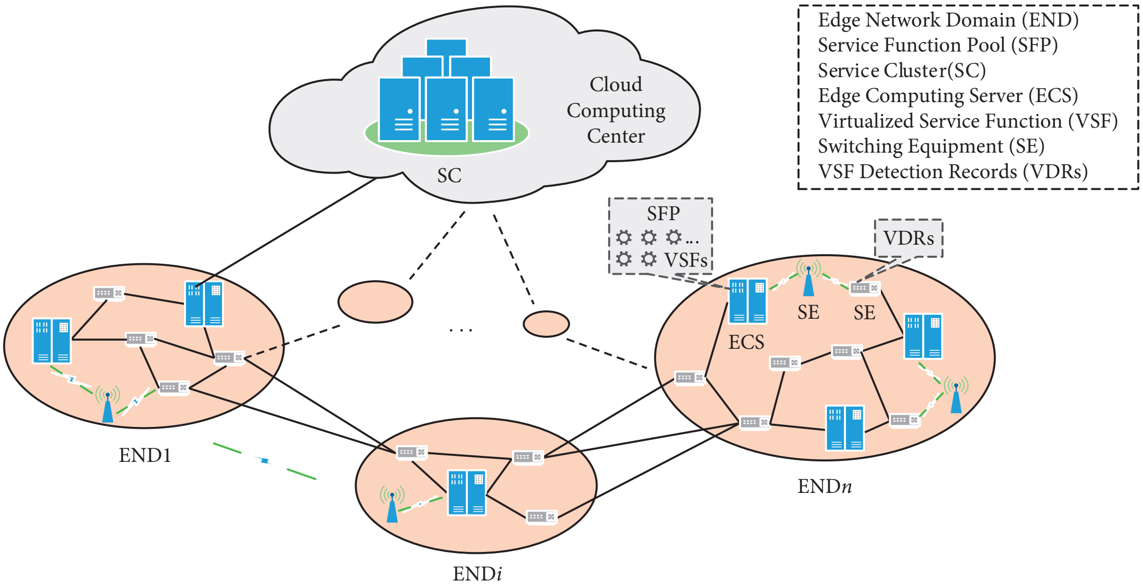

Edge Network Domain (END)
Service Function Pool (SFP)
Service Cluster(SC)
Edge Computing Server (ECS)
Virtualized Service Function (VSF)
Switching Equipment (SE)
VSF Detection Records (VDRs)
Cloud Computing Center
SC
SFP
VSFs
VDRs
SE
SE
ECS
END1
END*i*
END*n*
. . .

FIGURE 1: The system framework.

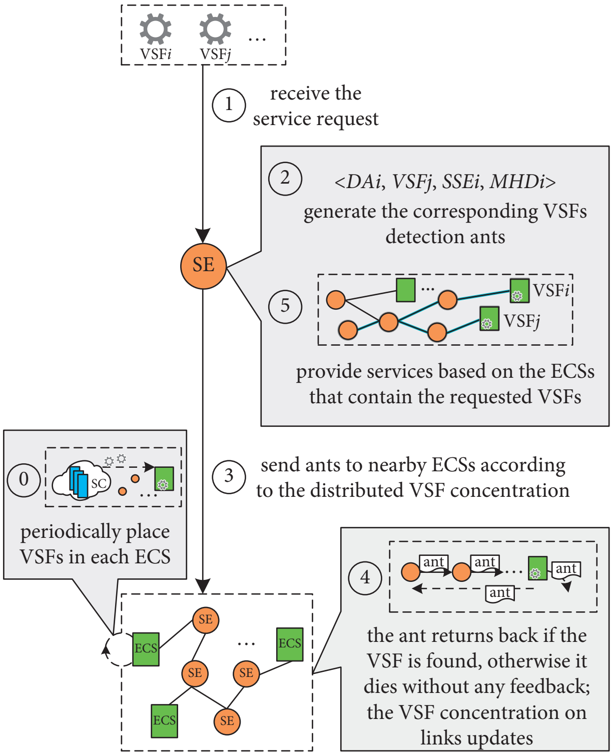

VSF*i*
VSF*j*
...
1
receive the service request
2
<*DAi*, *VSFj*, *SSEi*, *MHDi*>
generate the corresponding VSFs detection ants
SE
5
VSF*i*
VSF*j*
provide services based on the ECSs that contain the requested VSFs
0
SC
periodically place VSFs in each ECS
3
send ants to nearby ECSs according to the distributed VSF concentration
4
ant
ant
ant
ant
the ant returns back if the VSF is found, otherwise it dies without any feedback; the VSF concentration on links updates
ECS
SE
ECS
SE
SE
ECS
SE

FIGURE 2: The overall workflow of the system.

3. VSFs Detection

In order to minimize the service provision delay for edge equipment, most of services should be provided by the near ECSs rather than mainly by the SC that is far away from ENDs. It is a key issue to find the ECSs that have already been placed in the requested VSFs as soon as possible. In this section, the VSF detection ant and the VSF concentration are defined in our proposed scheme. The VSF detection ants are sent by the source SE that receives the service request to search for the locations (i.e., ECS) of the requested VSF according to this VSF concentration on links. The information on detecting the VSF is laid over the ants' trails, and the accumulated amount of information is regarded as the VSF concentration that is recorded and updated in the VDR.

In practice, the VSF concentration on links decreases with time. It also increases according to the ants' feedback if these ants have found the locations (i.e., ECSs) of the requested VSF. For instance, if the requested VSF is found by any of the VSF detection ants within the ant's living period, the ant returns back to the source SE, and the VSF concentration on the corresponding links increases. Comparing to the conventional ACO method, our presented VSF concentration updating does not depend on multiple iterations of the same group of ants. In the actual network environment, because of a large number of ongoing service requests for edge equipment, different VSF detection ants can be generated by SEs at any time, which enables different VSF concentrations continuously and efficiently to change in practice combined with the devised detection ant's feedback method.

3.1. *The VSF Detection Ant.*

In this paper, a VSF detection ant is defined as a four-tuple $\langle \mathrm{DA}_i, \mathrm{VSF}_k, \mathrm{SSE}_i, \mathrm{HDA}_i \rangle$. Here, DA_i is the unique identifier of a detection ant; VSF_k indicates the VSF that is detected by DA_i; SSE_i is the set of SEs that DA_i has passed; HDA_i denotes the number of survival hops of DA_i.

In detail, DA_i is one of the ants generated by the source SE that receives the service request and is forwarded to search for VSF_k according to the VSF_k concentration on links. The SEs passed by DA_i are orderly recorded in SSE_i to avoid loopback. Meanwhile, once DA_i finds VSF_k, it returns back to the source SE according to SSE_i. In addition, an ant is not allowed to detect the ECSs that are far away from the source SE due to the delay limitation. HDA_i is used to constrain the survival time of DA_i, and DA_i dies without any feedback if it has been forwarded exceeds HDA_i.

3.2. The VSF Detection Concentration. We assume two factors will influence the VSF concentration on links, which are time and ants' feedback. $\mathrm{VCL}_{\mathrm{SE}_u,\mathrm{SE}_v}^{\mathrm{VSF}_k}(t)$ is defined as the VSF_k concentration on the link from the SE SE_u to the SE SE_v at time t, and its value changes with time. At time $t+1$, its value is shown as follows:

$$\mathrm{VCL}_{\mathrm{SE}_u,\mathrm{SE}_v}^{\mathrm{VSF}_k}(t+1) = \mathrm{RVCL}_{\mathrm{SE}_u,\mathrm{SE}_v}^{\mathrm{VSF}_k}(t,t+1) + \mathrm{AVCL}_{\mathrm{SE}_u,\mathrm{SE}_v}^{\mathrm{VSF}_k}(t,t+1). \tag{1}$$

Here, $\mathrm{RVCL}_{\mathrm{SE}_u,\mathrm{SE}_v}^{\mathrm{NSF}_k}(t,t+1)$ and $\mathrm{AVCL}_{\mathrm{SE}_u,\mathrm{SE}_v}^{\mathrm{NSF}_k}(t,t+1)$ are the remaining concentration after volatilizing and the additive concentration after ants' feedback from t to $t+1$, respectively. They are defined as follows:

$$\begin{aligned} \mathrm{RVCL}_{\mathrm{SE}_u,\mathrm{SE}_v}^{\mathrm{VSF}_k}(t,t+1) &= \int_t^{t+1} \mathrm{VCL}_{\mathrm{SE}_u,\mathrm{SE}_v}^{\mathrm{VSF}_k}(t)\cdot e^{-\gamma\cdot t}\mathrm{d}t, \\ \mathrm{AVCL}_{\mathrm{SE}_u,\mathrm{SE}_v}^{\mathrm{VSF}_k}(t,t+1) &= \sum_{w=1}^{m} \Delta\mathrm{VCL}_{\mathrm{SE}_u,\mathrm{SE}_v}^{\mathrm{VSF}_k}(t,t+1)\cdot x_w. \end{aligned} \tag{2}$$

Different from the conventional ACO method, the VSF remaining concentration volatilizes faster and faster with time, because the frequent changing service requests from edge equipment may lead to the frequent migrating of VSFs. $\mathrm{RVCL}_{\mathrm{SE}_u,\mathrm{SE}_v}^{\mathrm{VSF}_k}(t,t+1)$ is designed to be continuously differentiable, and γ is a positive constant. The VSF additive concentration can only be brought by the survival ants that have found the requested VSF. The closer the link to the ECS that contains the requested VSF, the more this VSF concentration on the link increases. $\mathrm{AVCL}_{\mathrm{SE}_u,\mathrm{SE}_v}^{\mathrm{VSF}_k}(t,t+1)$ is designed to promote the searching efficiency for the following ants. m is the number of the VSF_k detection ants that have returned back between t and $t+1$. And x_w is related to the sequential position of SE_u in SSE_i, and it is defined as follows:

$$x_w = \begin{cases} \dfrac{\mathrm{SSE}_i[\mathrm{SE}_u]}{|\mathrm{SSE}_i|}, & \mathrm{SE}_u \in \mathrm{SSE}_i, \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

Here, $|\mathrm{SSE}_i|$ is the element number of SSE_i, and $\mathrm{SSE}_i[\mathrm{SE}_u]$ is the position of SE_u in SSE_i.

3.3. The Forwarding Probability. SE_v is defined as an adjacent SE of SE_u if a detection ant can be forwarded by SE_u to SE_v by only one hop. Not all adjacent SEs of SE_u can receive the detection ants from SE_u due to the limited number of the ants generated by the source SE. According to $\langle \mathrm{DA}_i, \mathrm{VSF}_k, \mathrm{SSE}_i, \mathrm{HDA}_i\rangle$, we define SASE_u is the set of adjacent SEs of SE_u, and $\mathrm{SASE}_u^{\mathrm{VSF}_k}$ is the set of adjacent SEs that can receive the VSF_k detection ant (e.g., DA_i) at the next hop from SE_u, which are shown as follows:

$$\mathrm{SASE}_u^{\mathrm{VSF}_k} = \mathrm{SASE}_u - \mathrm{SSE}_i \cap \mathrm{SASE}_u. \tag{4}$$

When searching for VSF_k, DA_i stops searching without any feedback if $\mathrm{SASE}_u^{\mathrm{VSF}_k}$ is empty. We assume $\mathrm{FP}_{\mathrm{SE}_u,\mathrm{SE}_v}^{\mathrm{VSF}_k}(t+1)$ is the forwarding probability that DA_i is forwarded from SE_u to SE_v, $\mathrm{SE}_v \in \mathrm{SASE}_u$. Here, the adjacent SE with higher $\mathrm{FP}_{\mathrm{SE}_u,\mathrm{SE}_v}^{\mathrm{VSF}_k}(t+1)$ only means that it can be forwarded more VSF_k detection ants rather than be forwarded all VSF_k detection ants. The $\mathrm{FP}_{\mathrm{SE}_u,\mathrm{SE}_v}^{\mathrm{VSF}_k}(t+1)$ is defined as follows:

$$\mathrm{FP}_{\mathrm{SE}_u,\mathrm{SE}_v}^{\mathrm{VSF}_k}(t+1) = \frac{\mathrm{VCL}_{\mathrm{SE}_u,\mathrm{SE}_v}^{\mathrm{VSF}_k}(t+1)}{\sum_{\mathrm{SE}_z\in\mathrm{SASE}_u^{\mathrm{VSF}_k}} \mathrm{VCL}_{\mathrm{SE}_u,\mathrm{SE}_z}^{\mathrm{VSF}_k}(t+1)}. \tag{5}$$

Especially, $\mathrm{FP}_{\mathrm{SE}_u,\mathrm{SE}_v}^{\mathrm{VSF}_k}(0) = 1/|\mathrm{SASE}_u^{\mathrm{VSF}_k}|$.

The working process of searching for the VSF_k is shown as Algorithm 1. In detail, firstly, when an SE receives the service request, as the source SE, it generates a certain number of detection ants to search for the requested VSF that is VSF_k here (line 2). Secondly, the same kinds of ants are divided into several groups in each arriving SE, with SSE_i of each ant being considered to avoid the loopback, the numbers of ants in different groups are determined according to the current concentrations of VSF_k on different links. Then, different groups of ants are forwarded to adjacent nodes of the current node (lines 3–5). Thirdly, if an VSF_k detection ant finds VSF_k, it adds the current node into its SSE_i and returns back to the source SE. Meanwhile, the VSF_k concentrations on the corresponding searching links update and the current node (i.e., ECS) is added into $\mathrm{ECS}^{\mathrm{VSF}_k}$ (lines 6–11). Else if HDA_i of the ant is not zero, the ant adds this node into its SSE_i and joins to the next-hop searching (lines 12–14). Otherwise, the ant dies without any feedback (lines 15–16). Finally, the set of ECSs that are found currently contain VSF_k (i.e., $\mathrm{ECS}^{\mathrm{VSF}_k}$) is obtained (line 20).

4. VSFs Placement

In order to optimize the service delivery delay, the ECSs locating near to the ENDs where the service requests generate should have been deployed in or migrated to the requested VSFs. In this paper, two kinds of VSFs that should have been placed in each ECS are considered, which are deployed VSFs (DVSFs) and migrated VSFs (MVSFs). DVSFs are the VSFs that have already been deployed in the ECS and are still being massively requested recently. MVSFs are the VSFs that should be migrated to the ECS due to the rapidly growing requests for them recently. Therefore, due to the ECS limited computing capacity, the VSFs that have already been deployed but rarely be requested recently in this ECS should be replaced by the MVSFs.

The performed frequencies of the already deployed VSFs in an ECS can be estimated according to these VSFs' current concentrations on the links around this ECS. Assume that SASE_q is defined as the set of adjacent SEs of the ECS, ECS_q. The ratio of performed frequency of VSF_k to the performed frequencies of all already deployed VSFs in ECS_q from t to $t+1$ is defined as $\mathrm{RPF}_q^{\mathrm{VSF}_k}(t,t+1)$, shown as follows:

$$\mathrm{RPF}_q^{\mathrm{VSF}_k}(t,t+1)=\frac{\sum_{\mathrm{SE}_z\in\mathrm{SASE}_q}\mathrm{VCL}_{\mathrm{SE}_z,\mathrm{ECS}_q}^{\mathrm{VSF}_k}(t+1)}{\sum_{\mathrm{VSF}_e\in\mathrm{SVSF}_q(t,t+1)}\sum_{\mathrm{SE}_z\in\mathrm{SASE}_q}\mathrm{VCL}_{\mathrm{SE}_z,\mathrm{ECS}_q}^{\mathrm{VSF}_e}(t+1)}. \tag{6}$$

Here, $\mathrm{SVSF}_q(t,t+1)$ is the set of VSFs that have been performed in ECS_q from t to $t+1$.

Let $\mathrm{CS}_q^{\mathrm{VSF}_k}(t+1-m,t+1)$ be the concentration stability of VSF_k on the links around ECS_q during the recent m time periods, shown as follows:

$$\mathrm{CS}_q^{\mathrm{VSF}_k}(t+1-m,t+1)=\sqrt{\frac{1}{m}\sum_{x=t+1-m}^{t+1}\left(\sum_{\mathrm{SE}_z\,\in\,\mathrm{SASE}_q}\mathrm{VCL}_{\mathrm{SE}_z,\mathrm{ECS}_q}^{\mathrm{VSF}_k}(x)-\frac{\sum_{y=t+1-m}^{t+1}\sum_{\mathrm{SE}_z\,\in\,\mathrm{SASE}_q}\mathrm{VCL}_{\mathrm{SE}_z,\mathrm{ECS}_q}^{\mathrm{VSF}_k}(y)}{m}\right)^2},\ t+1\geq m. \tag{7}$$

Assume RTV and CTV are the threshold values of the VSF's ratio of performed frequency and concentration stability, respectively. The current set of DVSFs that can be retained in ECS_q is defined as CSD_q, obviously, $\mathrm{CSD}_q\subseteq\mathrm{SVSF}_q(t,t+1)$. The VSFs in CSD_q should satisfy the following conditions:

$$\mathrm{RPF}_q^{\mathrm{VSF}_k}(t,t+1)\geq\mathrm{RTV}, \tag{8}$$

$$\mathrm{CS}_q^{\mathrm{VSF}_k}(t+1-m,t+1)\leq\mathrm{CTV},\quad t+1\geq m, \tag{9}$$

or

$$\mathrm{RPF}_q^{\mathrm{VSF}_k}(t,t+1)<\mathrm{RTV}, \tag{10}$$

$$\sum_{\mathrm{SE}_z\in\mathrm{SASE}_q}\mathrm{VCL}_{\mathrm{SE}_z,\mathrm{ECS}_q}^{\mathrm{VSF}_k}(h)\geq\sum_{\mathrm{SE}_z\in\mathrm{SASE}_q}\mathrm{VCL}_{\mathrm{SE}_z,\mathrm{ECS}_q}^{\mathrm{VSF}_k}(h-1),\quad t+1-m<h\leq t+1. \tag{11}$$

Let $\mathrm{SCV}_q(t+1)$ be the set of candidate VSFs that have been requested but not yet been deployed in ECS_q during the recent m time periods, these VSFs may replace the VSFs in $\mathrm{SVSF}_q(t,t+1)-\mathrm{CSD}_q$ with $\mathrm{SCV}_q(t+1)\cap\mathrm{SVSF}_q(t,t+1)=\varnothing$ satisfied. Let CSM_q be the current set of MVSFs that should be migrated to ECS_q, obviously, $\mathrm{CSM}_q\subseteq\mathrm{SCV}_q(t+1)$. For the VSF $\mathrm{VSF}_d\mathrm{VSF}_d\in\mathrm{SCV}_q(t+1)$, its total requested number in ECS_q during the recent m time periods is defined as $\mathrm{TRN}_q^{\mathrm{VSF}_d}$, and its incremental requested number in ECS_q in each recent time period is denoted as follows:

$$\mathrm{IRN}_q^{\mathrm{VSF}_d}(h)=\mathrm{TRN}_q^{\mathrm{VSF}_d}-\mathrm{IRN}_q^{\mathrm{VSF}_d}(h),\quad t+1-m\leq h\leq t+1. \tag{12}$$

Furthermore, the requested number growth rate of VSF_d in ECS_q in the $(h)th$ day is denoted as follows:

$$\mathrm{RNGR}_q^{\mathrm{VSF}_d}(h)=\frac{\mathrm{IRN}_q^{\mathrm{VSF}_d}(h)}{\mathrm{TRN}_q^{\mathrm{VSF}_d}},\quad t+1-m\leq h\leq t+1. \tag{13}$$

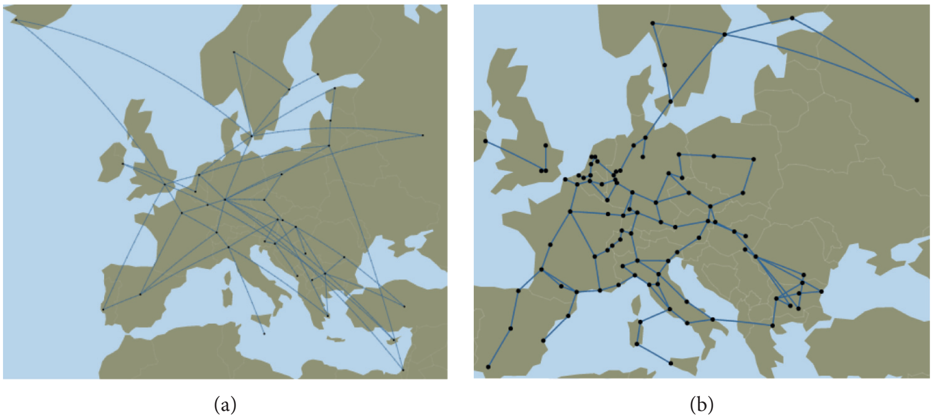
In this approach, the VSFs in CSM_q can be selected from $\mathrm{SCV}_q(t+1)$ by comparing the VSF's requested number growth rate from high to low with the ECS's available computing capacity considered. And these VSFs can be migrated from the SC in the cloud computing center or other ECSs. In addition, a VSF in CSM_q may become one element of CSD_q if it satisfies the conditions of equations (8) and (9), or equations (10) and (11) in the following time periods. Therefore, the VSFs that should be placed in ECS_q before the next time period are the ones belonging to $\mathrm{CSD}_q\cup\mathrm{CSM}_q$.

5. Simulation and Results

5.1. The Simulation Setup. In the simulation, the proposed schemes are implemented in Python and all experiments are performed on a computer with one Intel(R) Core(TM) i7-6700 CPU @ 3.40 GHz and 16 GB of RAM. We evaluate the approaches in two typical and real network topologies (e.g., ISP and backbone networks) with different numbers of nodes and links, called Geant and Interroute, which can be obtained from the Internet Topology Zoo [17]. Specifically, Geant is a middle-scale network topology with 41 nodes and 65 links, and Interroute is a large-scale network topology with 110 nodes and 148 links. They are shown in Figure 3.

```
Input: $SE_u$/ * the source switching equipment * /, $VSF_k$/ * the service function that is requested * /
Output: $ECS^{VSF_k}$/ * the set of ECSs that contain $VSF_k$, * /
  Begin
    $SE_u$ generates detection ants, each of which is defined as $\langle DA_i, VSF_k, SSE_i, HDA_i \rangle$;
    while the set of $VSF_k$ detection ants is not empty do
    Divide ants into several groups according to the $SSE_i$ of each ant and the distributed $VSF_k$ concentration
    Forward different groups to different adjacent nodes of the current node according to equation (5);
    for each $DA_i$do
      if$VSF_k$ is found then
        Add this node into $SSE_i$;
          Return back to $SE_u$ according to $SSE_i$;
        Update $VSF_k$ concentrations on links according to equation (2);
        Add this node (i.e., ECS) into $ECS^{VSF_k}$;
      else if$HDA_i \geq 1$then
        Add this node into $SSE_i$;
        Join the next-hop searching;
      else if$HDA_i = 0$then
        Die without any feedback;
      end if
    end for
  end while
  return$ECS^{VSF_k}$
End
```

Algorithm 1: The search for a VSF.


(a)

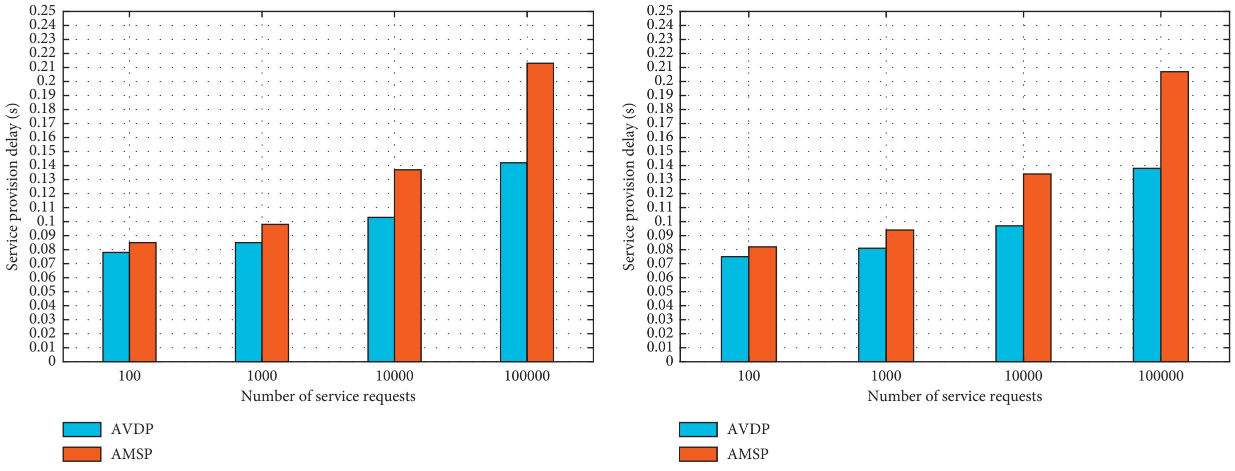
(b)

Figure 3: Network topologies. (a) Geant topology. (b) Interroute topology.

We assume that the network topology is divided into 6 edge network domains, and one ECS is placed among 10 nodes in the simulation. A variety of VSFs are simulated by ClickOS [18], and it can create small virtual machines, each of which is able to host a VSF. We assume that the computing capacity of each ECS follows a uniform distribution between 100 and 200 units. The number of VSFs requested by each request follows a uniform distribution between 2 and 4, and the type of each VSF is random. The ECS computing capacity needed to support each VSF follows the uniform distribution between 5 and 10 units. The simulation parameters and the corresponding distribution model are motivated by the literature [19, 20] that studies the network function provision problem. We compare the proposed AVDP with the scheme of AI-enabled mobile multimedia service instance placement (AMSP) according to [13]. We use the following performance metrics: the service provision delay (SPD), the computing capacity utilization (CCU), and the service access success ratio (SASR).

5.2. The Simulation Results. We compare the SPDs of the two approaches of AVDP and AMSP under Geant and Interroute. The SPD is defined as the time interval from receiving the service request to successfully providing the service. The results are shown in Figures 4 and 5.

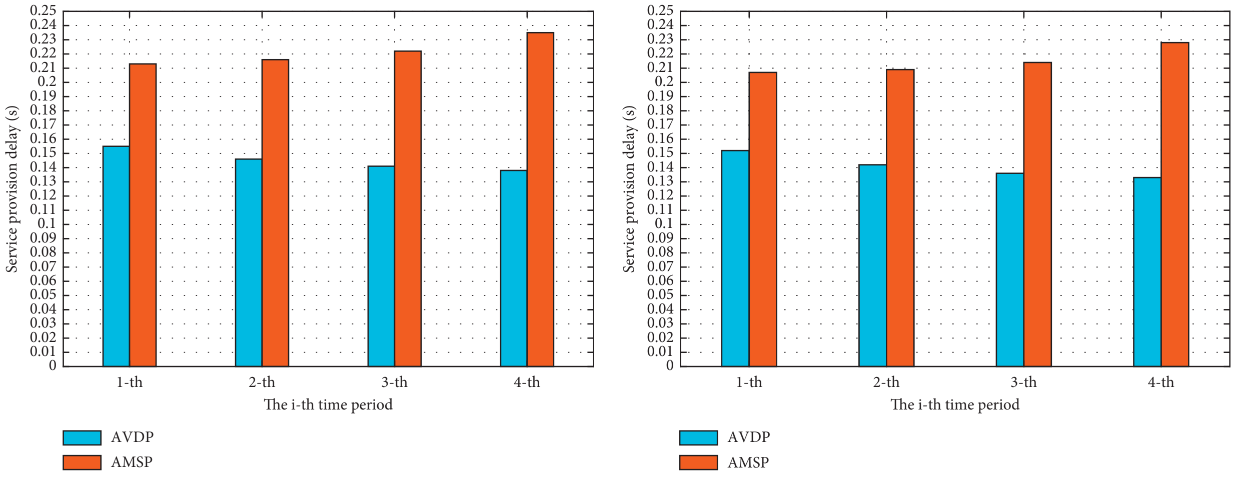
As shown in Figures 4 and 5, the SPD of AVDP is always lower than that of AMSP with the number of service requests increasing (i.e., Figure 4). In addition, we also compare the SPDs of the two approaches with the time period increasing under the network load of 10000 service requests (i.e., Figure 5). In more detail, when the number of service



(a)

(b)

Figure 4: Service provision delay over different network loads. (a) Under Geant. (b) Under Interroute.



(a)

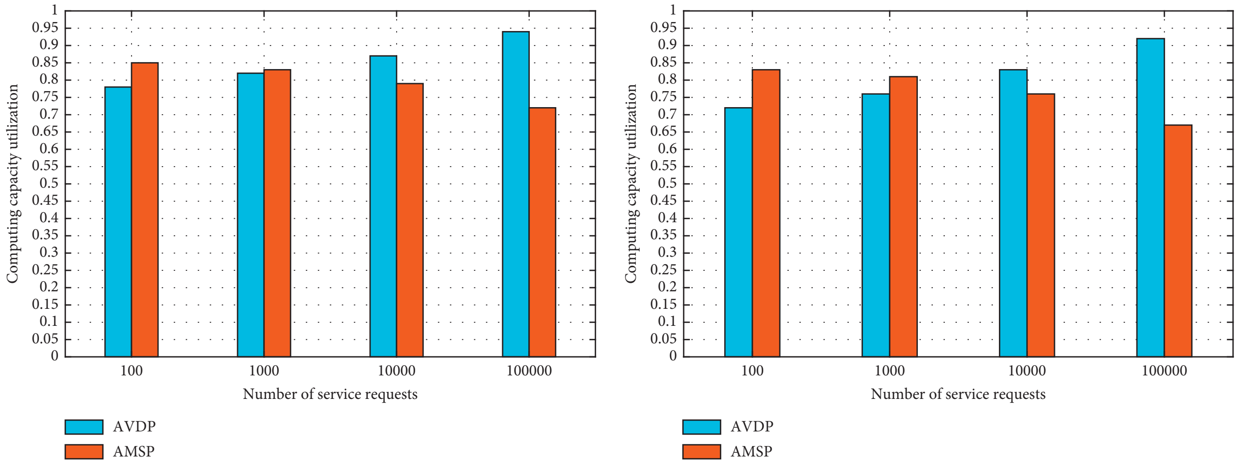
(b)

Figure 5: Service provision delay over different network loads. (a) Under Geant. (b) Under Interroute.

requests is low, the SPDs of AVDP and AMSP are 0.079 s and 0.085 s, respectively. When the number of service requests reaches the maximum, the SPD of AVDP just increases by 0.061 s, while the SPD of AMSP increases by 0.125 s. Moreover, when the network load is the heaviest, the SPD of AVDP reduces to 0.132 s with the time period increasing; that is, the SPD of AVDP is optimized with time, while that of AMSP increases to 0.235 s in the same situation. The reasons are as follows. In AVDP, the improved ACO method enables the VSF concentration on links to be updated in time by efficiently accelerating volatilization and enhancing feedback. Thus, the requested VSFs can be quickly found from the nearest ECSs, which significantly reduces the time overhead of searching for services. Furthermore, AVDP continuously optimizes the deployed locations of VSFs over time, which enables the VSF searching delay of AVDP to be reduced over time. Thus, the SDP of AVDP is further improved. However, AMSP barely changes over time, it mainly provides services on demands by instantly deploying service functions rather than leveraging already deployed functions. Real-timely deploying most of requested service functions leads to large delay for AMSP to provide new services, especially when the network load becomes heavy.

We compare the CCUs of the two approaches of AVDP and AMSP under Geant and Interroute. The CCU is defined as the ratio of the VSFs performed in an ECS to the total already deployed VSFs in this ECS. The results are shown in Figures 6 and 7.

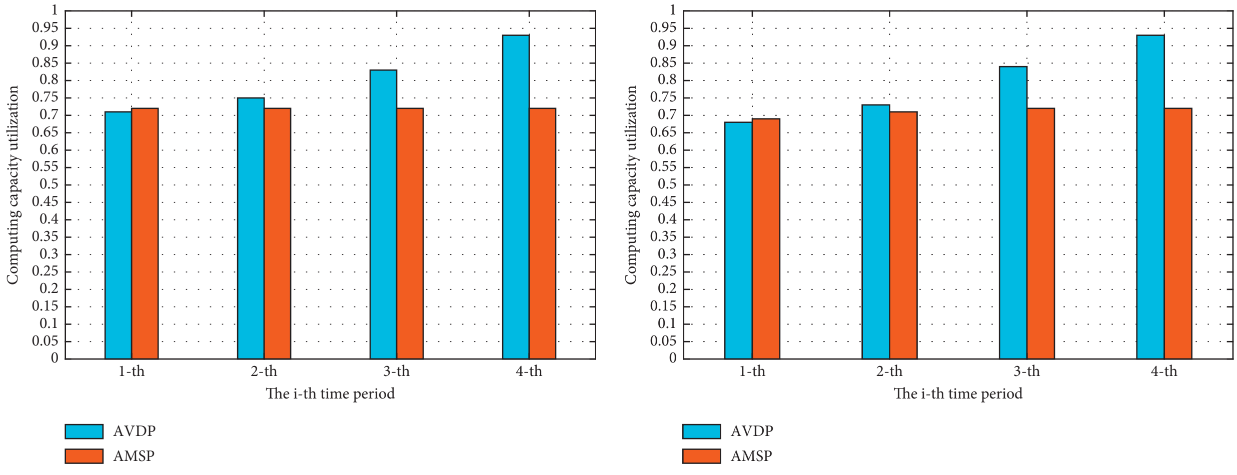
As shown in Figures 6 and 7, the CCU of AVDP increases with the number of service requests increasing, while that of AMSP decreases when the network load becomes heavy (i.e., Figure 6). We also compare the CCUs of the two approaches with the time period increasing under the network load of 10000 service requests (i.e., Figure 7). In more detail, when the number of service requests is low, the CCUs of AVDP and AMSP are about 77% and 84%,



(a)

(b)

Figure 6: Computing capacity utilization over different network loads. (a) Under Geant. (b) Under Interroute.



(a)

(b)

Figure 7: Computing capacity utilization over different time periods. (a) Under Geant. (b) Under Interroute.

respectively. With the network load becoming heavy, the CCU of AVDP increases and it reaches about 94% under the network load of 10000 service requests, while that of AVDP drops to 72% at the same network status. Moreover, when the network load is the heaviest, the CCU of AVDP increases from 71% to 93% with the time period increasing while that of AMSP is always keeping about at 72%. That is, the CCU of AVDP continuously optimizes while that of AMSP barely changes over time. The reasons are as follows. AVDP dynamically deploys VSFs by taking the frequencies of the VSFs being requested and performed in an ECS into account. With the number of service requests increasing, the VSFs rarely performed recently can be replaced by the ones with higher requested frequencies in an ECS; thus, the computing capacity of each ECS can be fully used. In addition, the requested VSFs can be efficiently found by AVDP, which enables the appropriate VSFs to be more frequently performed in each ECS. Thus, the CCU of AVDP is significantly optimized. However, AMSP mainly makes full use of some VSFs that are currently requested frequently in an ECS, and it does not consider replacing already deployed VSFs that are rarely requested by new popular VSFs, which cannot further optimize the ECS computing capacity utilization. In addition, AMSP does not have the ability to achieve the VSF future popularity prediction, and the CCU of AMSP cannot be improved over time.

We compare the SASRs of the two approaches of AVDP and AMSP under Geant and Interroute. The SASR is defined as the ratio of the requests that successfully obtain services to the total requests asking for services. The results are shown in Figures 8 and 9.

As shown in Figures 8 and 9, the SASR of AVDP is much higher than that of AMSP when the number of service requests increases rapidly (Figure 8). We also compare the SASRs of the two approaches with the time period increasing under the network load of 10000 service requests (Figure 9).

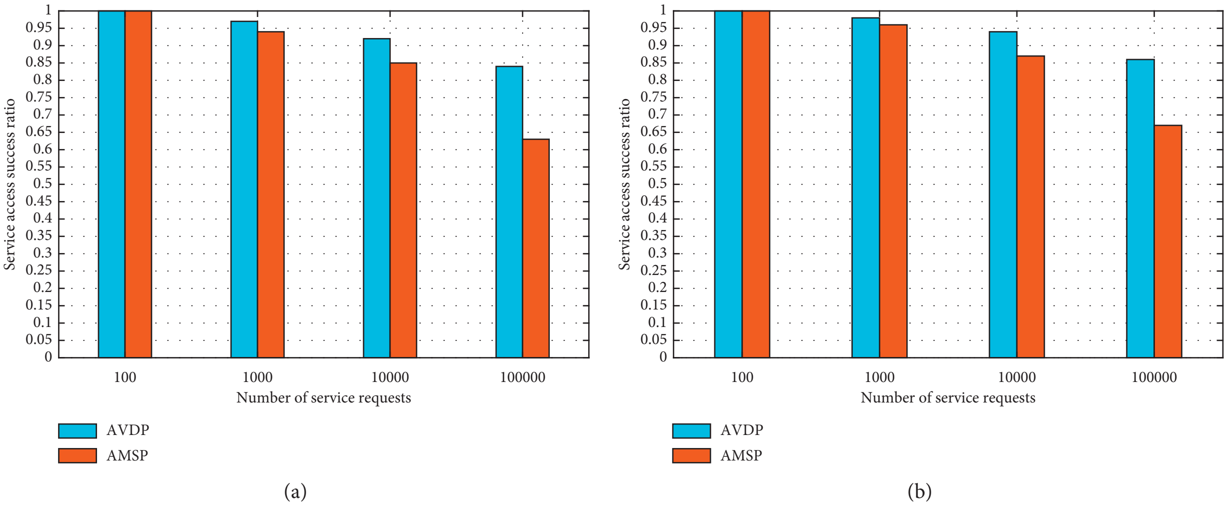


Figure 8: Service access success ratio over different network loads. (a) Under Geant. (b) Under Interroute.

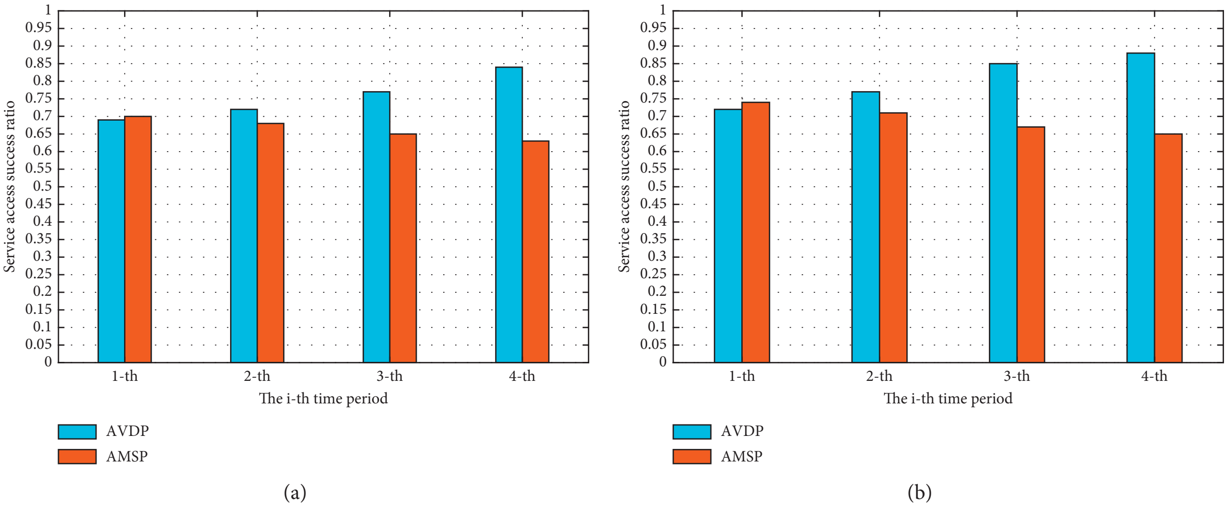


Figure 9: Service access success ratio over different time periods. (a) Under Geant. (b) Under Interroute.

In more detail, when the number of service requests is low, the SASRs of AVDP and AMSP are almost approaching 1. With the number of service requests increasing, the SASRs of AVDP and AMSP drop to about 84% and 63%, respectively. However, the SASR of AVDP still keeps beyond 84%, especially under the heaviest network load. Moreover, the SASR of AVDP increases from about 72% to about 84% with the time period increasing, while that of AMSP just improves a little and keeps at about 65% over time. Under the heaviest network load, the SASR of AVDP can achieve about 20% higher than that of AMSP. The reasons are as follows. AVDP mainly deploys most of VSFs before these VSFs are massively requested, which does not occupy much network resource to real-timely dispatch the requested VSFs. Thus, the available network resource can support as many new requests as possible, especially under the heavy network load. In addition, in AVDP, multiple ECSs that contain the requested VSFs can be found, which increases the probabilities that successfully providing corresponding services to new receiving requests when most of ECSs are busy. Thus, the SASR of AVDP is significantly optimized. However, AMSP mainly provides services by the server on the mobile path without considering balancing the working load by cooperating with multiple nearby servers. Under the heavy network load, the SASR of AMSP decreases rapidly.

6. Conclusions

In this paper, by introducing VSFs into edge computing pattern, we propose the mechanism of improved ACO-inspired VSFs detection and placement. By efficiently detecting the already deployed locations of VSFs and dynamically placing appropriate VSFs in suitable ECSs, the service provision delay to the edge equipment and the computing capacity utilization of each ECS under the edge computing scenario are significantly optimized. In this mechanism, the approach of searching for the requested VSFs from multiple ECSs is devised, and it improves the ACO method to adapt to the edge computing scenario by defining the VSF detection ant and the VSF concentration on links. Furthermore, the approach of deploying VSFs in each ECS is devised, and it takes the frequency variations of the VSF being performed and being requested in an ECS into account, so as to select the most appropriate VSFs to place in the ECS with this ECS computing capacity utilization considered. Simulation results show that the proposed mechanism has significant improvements in the service provision delay optimization and the computing capacity utilization improvement compared with the current state of the art.

Data Availability

The data used in this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (nos. 62002261 and 62072336) and the Tianjin Municipal Education Commission Scientific Research Project (no. 2018KJ145).

References

[1] F. Alvarez, D. Breitgand, D. Griffin et al., "An edge-to-cloud virtualized multimedia service platform for 5G networks," *IEEE Transactions on Broadcasting*, vol. 65, no. 3, pp. 369–380, 2019.

[2] J. Gil Herrera and J. F. Botero, "Resource allocation in NFV: a comprehensive survey," *IEEE Transactions on Network and Service Management*, vol. 13, no. 3, pp. 518–532, 2016.

[3] K. Kaur, V. Mangat, and K. Kumar, "A comprehensive survey of service function chain provisioning approaches in SDN and NFV architecture," *Computer Science Review*, vol. 38, p. 100298, 2020.

[4] T. Taleb, K. Samdanis, B. Mada, H. Flinck, S. Dutta, and D. Sabella, "On multi-access edge computing: a survey of the emerging 5G network edge cloud architecture and orchestration," *IEEE Communications Surveys and Tutorials*, vol. 19, no. 3, pp. 1657–1681, 2017.

[5] J. Pan and J. McElhannon, "Future edge cloud and edge computing for internet of things applications," *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 439–449, 2018.

[6] M. Laroui, B. Nour, H. Moungla, M. A. Cherif, H. Afifi, and M. Guizani, "Edge and fog computing for IoT: a survey on current research activities & future directions," *Computer Communications*, vol. 180, pp. 210–231, 2021.

[7] Y. Liu, M. Peng, G. Shou, Y. Chen, and S. Chen, "Toward edge intelligence: multiaccess edge computing for 5G and internet of things," *IEEE Internet of Things Journal*, vol. 7, no. 8, pp. 6722–6747, 2020.

[8] C. Li, Q. Cai, and Y. Lou, "Optimal data placement strategy considering capacity limitation and load balancing in geographically distributed cloud," *Future Generation Computer Systems*, vol. 127, pp. 142–159, 2022.

[9] A. Laghrissi and T. Taleb, "A survey on the placement of virtual resources and virtual network functions," *IEEE Communications Surveys & Tutorials*, vol. 21, pp. 1409–1434, 2019.

[10] X. Ma, S. Wang, S. Zhang, P. Yang, C. Lin, and X. Shen, "Cost-efficient resource provisioning for dynamic requests in cloud assisted mobile edge computing," *IEEE Transactions on Cloud Computing*, vol. 9, no. 3, pp. 968–980, 2021.

[11] C. Bu, X. Wang, M. Huang, and K. Li, "SDNFV-based dynamic network function deployment: model and mechanism," *IEEE Communications Letters*, vol. 22, no. 1, pp. 93–96, 2018.

[12] C. Bu and J. Wang, "Computing tasks assignment optimization among edge computing servers via SDN," *Peer-to-Peer Networking and Applications*, vol. 14, pp. 1190–1206, 2021.

[13] P. Roy, S. Sarker, M. A. Razzaque et al., "AI-enabled mobile multimedia service instance placement scheme in mobile edge computing," *Computer Networks*, vol. 182, no. 9, pp. 1–14, 2020.

[14] L. Yin, P. Li, and J. Luo, "Smart contract service migration mechanism based on container in edge computing," *Journal of Parallel and Distributed Computing*, vol. 152, pp. 157–166, 2021.

[15] L. Rui, M. Zhang, Z. Gao, X. Qiu, Z. Wang, and Ao Xiong, "Service migration in multi-access edge computing: a joint state adaptation and reinforcement learning mechanism," *Journal of Network and Computer Applications*, vol. 183-184, pp. 1–17, 2021.

[16] L. Tang, X. He, P. Zhao, G. Zhao, Yu Zhou, and Q. Chen, "Virtual network function migration based on dynamic resource requirements prediction," *IEEE Access*, vol. 7, pp. 112348–112362, 2019.

[17] The Univesity of Adelaide, "The internet topology Zoo," 2012, https://www.topology-zoo.org/.

[18] J. Martins, M. Ahmed, C. Raiciu et al., "ClickOS and the art of network function virtualization," in *Proceedings of the 11th USENIX Conference on Networked Systems Design and Implementation*, pp. 459–473, Seattle, USA, April 2014.

[19] S. Yang, F. Li, M. Shen, X. Chen, X. Fu, and Y. Wang, "Cloudlet placement and task allocation in mobile edge computing," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 5853–5863, 2019.

[20] D. Li, P. Hong, K. Xue, and J. Pei, "Virtual network function placement considering resource optimization and SFC requests in cloud datacenter," *IEEE Transactions on Parallel and Distributed Systems*, vol. 29, no. 7, pp. 1664–1677, 2018.