IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received July 29, 2018, accepted September 6, 2018, date of publication September 17, 2018, date of current version October 12, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2870129

The Stochastic-Learning-Based Deployment Scheme for Service Function Chain in Access Network

YOUCHAO YANG, QIANBIN CHEN, (Senior Member, IEEE), GUOFAN ZHAO, PEIPEI ZHAO, AND LUN TANG

School of Communication and Information Engineering, Chongqing University of Posts and Telecommunications, Chongqing 400065, China

Corresponding author: Qianbin Chen (chenqb@cqupt.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61571073 and in part by the Chongqing Science and Technology Commission through the generic key technology innovation projects for key industries under Grant cstc2015zdcy-ztzx40008.

ABSTRACT The research on topology-aware deployment for service function chain (SFC) is very important to a network slice technique. Nevertheless, most of the existing schemes on topology-aware deployment assume that the network topology information (NTI) is completely observed, which is unrealistic, considering the topology observation errors in practical network environment. In this paper, we consider the SFC deployment based on realistic topology sensing in fifth-generation cloud-radio access network (C-RAN). Due to the unavoidable errors, the realistic topology observation results merely represent partial NTI. Therefore, the partial observation Markov decision process (POMDP) is used in this paper to estimate the whole real topology condition. Then, a POMDP-based SFC deployment scheme is proposed. In this scheme, considering the particularity of SFC deployment in C-RAN, the SFC deployment problem is defined as a series of deployment decisions, including repair decisions, selection decisions, and resource allocation decisions. Our objective is to maximize the utility associated with the total delay and server-repair cost. And the POMDP scheme, according to the queue state information and partially observable NTI, makes deployment policies to maximize the utility by Bellman iteration. To reduce the iteration complexity, a point-based mingled heuristic value iteration algorithm is formulated in this paper. The simulation results show that the performance of C-RAN in terms of the system total delay and throughput can be significantly improved by using the proposed POMDP scheme.

INDEX TERMS Network slice, deployment of service function chain, network topology observation, partial observation Markov decision process.

I. INTRODUCTION

Network slice (NS) is a critical technology for 5G mobile communication network [1]–[3]. This technology is expected to reduce capital expenditure and operating expenditure [4], and satisfy the requirement of diverse services [5]–[8]. An NS usually contains several service function chains (SFCs) [9], [10]. Since all these SFCs in an NS pass through evolution packet core (EPC) network and cloud-radio access network (C-RAN), the NS is also made up of a EPC slice and a RAN slice. When a specific NS is created by the operator, these SFCs in the NS need to be deployed optimally [11]. There are massive schemes on the deployment for SFCs. Especially, topology-aware deployment scheme is considered as a promising method in the dynamic network environment [12]. However, most of schemes on topology-aware deployment for SFCs assume that network topology information (NTI) is completely observed. This assumption is unrealistic when we consider the topology observation errors in practical network environment.

In practical network with network function virtualization (NFV) structure, the system obtains NTI by observing each server's work state. Usually, an observation mechanism is used to observe whether a server work normally or not. In fact, due to the accidental errors in the observation mechanism, the observation results may not accord with the some servers' actual work states. If a topology-aware scheme ignores the observation errors and directly considers the observable NTI as the whole real topology condition, it's performance may decrease when they are applied in practical network. Therefore, considering the partial observation, the scheme based on realistic topology sensing is a more significant method for SFC deployment in practical dynamic network.

Most of recent topology-aware schemes on SFC deployment consider that the network topology is completely observed. In [13], a dynamic resource allocation scheme is proposed with completely observable network topology. The author considers the joint allocation of computing resource and bandwidth for SFCs so as to minimize the utility of system scheduling delay and cost. In [14], with considering completely observable topology, Lei *et al.* proposes a mixed integer linear program (MILP) model for SFC deployment to optimize the delay, and an online algorithm is proposed for solving the MILP model. In [15], the SFC deployment is formulated as finite Markov states in dynamic hybrid network. Due to the dynamic network topology, the state space may change, and a dynamic Markov approximation (MA) scheme is proposed to maximize the utility associated with the network throughput and resource utilization. Considering dynamic network topology, a SFC deployment scheme based on Markov decision processes (MDP) is proposed in [16]. The scheme dynamically makes the decisions on SFC deployment according to the completely observable NTI. In [17], Qu *et al.* considers the joint problem of traffic steering and virtual network function (VNF) scheduling as a dynamic programming mathematical model under dynamic network. And a genetic algorithm is proposed to obtain optimal deployment policy. The migration model on the SFC deployment is proposed in [18] to reduce the energy consumption. In this scheme, when the topology changes are completely observed, the migration policy is proposed to determine when and where to migrate the VNFs of SFCs. So the migration policy can dynamically be in response to changes of SFC request and network topology. However, it should be noted that these schemes in [13]–[18] consider that the whole real topology condition could be observed completely. In practical network, there may be topology observation errors in observation results. So it is unrealistic to directly consider the observation results as the real topology condition. In addition, all above SFC deployment schemes focus on solving the SFC deployment problem in EPC network. Compared with EPC, C-RAN has different structure, so there are some differences between the SFC deployment method in EPC and C-RAN. And these differences make these schemes unavailable for SFC deployment in C-RAN.

In this paper, we consider the SFC deployment based on realistic topology-aware in dynamic C-RAN, and propose a novel partial observation Markov decision process (POMDP) scheme for SFC deployment of RAN slices. The POMDP is a kind of stochastic learning, which is a powerful tool to process the decision-making problem under uncertain environment. The distinct features of this paper are as follows.

1) Heartbeat packet detection mechanism (HPDM) is used to observe servers' work state. Considering realistic topology-aware, the POMDP is applied in this paper to estimate the real topology condition by partially observing each server's state.
2) In order to achieve maximal utility associated with the total delay and server-repair cost, a POMDP scheme for SFC deployment in C-RAN is proposed in this paper. Joint decisions of repair decisions, selection decisions and resource allocation decisions are designed for SFC deployment in RAN slices. And two major influential factors which affect the decisions are considered in the POMDP scheme, including NTI and QSI.
3) When we consider the Bellman iteration for the POMDP scheme, the complexity of iteration is very high. Therefore, a point-based mingled heuristic value iteration (MHVI) algorithm is used in this paper to reduce the complexity.

The rest of this paper is organized as follows. Section II describes the system model. In section III, the problem of SFC deployment in RAN slice is formulated as a series of decisions. Section IV describes The POMDP scheme for SFC deployment. Section V describes MHVI algorithm. The simulation results are discussed in Section VI. Several conclusions are present in section VII.

II. SYSTEM MODEL

In this section, we describe the model of SFC deployment in C-RAN, based on realistic topology-aware.

With the NFV structure in C-RAN, the system model is shown in Fig.1. In the structure of NFV, each protocol function in C-RAN is virtualized to be a VNF by the software executing on the common servers. And these VNFs could share the common network infrastructure. Different from the EPC, C-RAN consists of virtual distribution units (DUs) and centralization units (CUs) for each SFC in RAN slices, so we consider that there are two physical networks in C-RAN. One physical network is in charge of providing virtual DUs with resources. The system could instantiate various virtual DUs for diverse services in application layer, so there is a DU pool on this physical network. Analogously, another physical network forms the CU poll. In general, a virtual DU or CU for a SFC contains several VNFs on the corresponding pool. For example, as shown in Fig.1, in NS 1 the virtual DU of SFC2 contains VNF-1 and its virtual CU contains VNF-2, VNF-3, VNF-4 and VNF-5. Besides, a virtual DU communicates with the corresponding virtual CU by fronthaul network.

In our system model, we consider U as the set of RAN slices which are in the condition of uplink, and let ζ_u be the set of SFCs in slice u. The physical networks of DU pool and CU pool are regarded as two undirected graphs, and they are defined as $G_1 = (N_1, L_1)$ and $G_2 = (N_2, L_2)$ respectively, where N_1 and N_2 are the sets of server nodes in DU pool and CU pool respectively, L_1 and L_2 are the sets of links in two physical network respectively. v_{n_k} is computational capacity of server n_k, while v_l is the bandwidth capacity of link l.

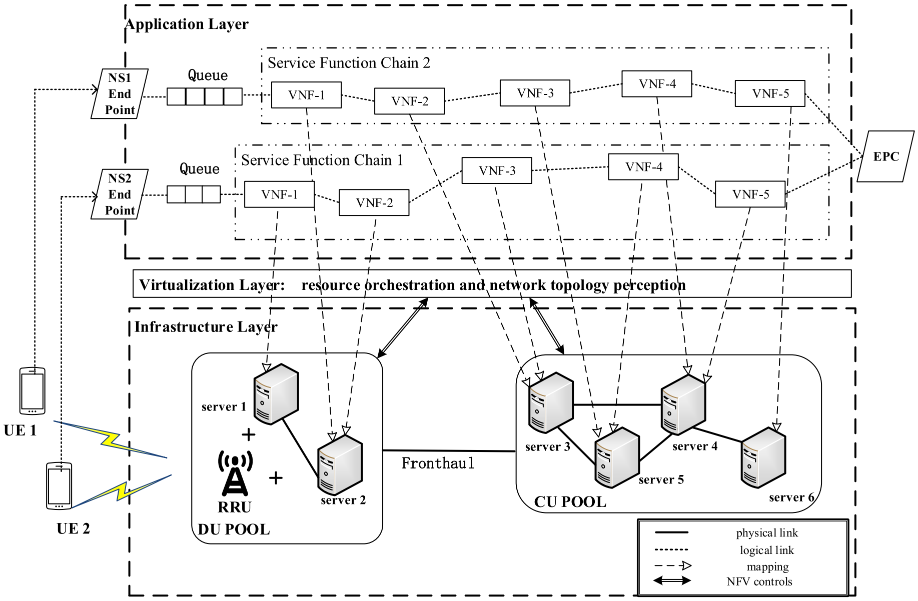


FIGURE 1. System model.

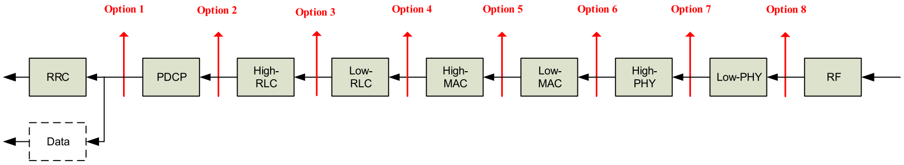


FIGURE 2. Eight methods of VNF placement in C-RAN.

In addition, we consider that data flows could be cached in DU pool, that is, each SFC has a queue. Let $q_{u,m}(t)$ be the queue length of SFC m ($m \in \zeta_u$) at time slot t and n_q be the allowable maximum queue length.

With the limit of fronthaul bandwidth resource, the RAN slices could place the VNFs of SFCs in DU pool and CU pool flexibly to satisfy quality of service (QoS). As shown in Fig.1, in the condition of uplink, NS 1 instantiates the VNF-1 in DU pool, while NS 2 instantiates the VNF-1 and VNF-2 in the DU pool, others are placed in CU pool. The two methods of VNF placement are used to satisfy their own different delay requirements or adapt to the changes of topology. In this paper, under the condition of up-link, we summarize 8 methods of VNF placement in RAN slices, as shown in Fig.2, and all the methods are in the discussion according to the 3GPP technical report. When a RAN slice selects the method of VNF placement from the option1 to option8 in turn, its consumption of fronthaul network resource gradually increases. Besides, some options' maximum allowed transmission latency in fronthaul network is different according to 3GPP technical report. The maximum allowed transmission latency of option1, option2 and option3 ranges form 1.5ms to 10ms. The maximum allowed transmission latency of option 4 and option 5 is about 500 us with the limitation of real-time performance of radio link control (RLC) protocol. Since the hybrid automatic repeat request (HARQ) function in the option 6, option 7 and option 8 is placed in CU, their maximum allowed transmission latency is equal to 250us. Of course, each option has its own shortcoming. The option 1, option 2 and option 3 easily cause high loads of DU pool, which may affect the reliability of SFCs. The option 4 and option 5 can increase the complexity of the relevant interfaces. Due to the strict requirement of latency in fronthaul network, the option 6, option 7 and option 8 may waste a lot of fronthaul resources and lead to poor flexibility of network.

On the one hand, the selection of method of VNF placement for each RAN slice decides the number of VNFs instantiated in DU pool and CU pool, which may affect the virtual computing and bandwidth resource allocation in two pools. On the other hand, different methods of VNF placement need to adjust allocation of fronthaul resource to satisfy the QoS such as throughput or transmission delay. Therefore, the selection of method of VNF placement is a significant factor for SFC deployment in RAN slices. The proposed scheme considers the selection when it makes deployment decisions for SFCs in RAN slices.

In fact, the physical networks of C-RAN tend to be in the outside with more complex environment. It is easy for these common servers in the physical networks to fail to work normally with a lot of stochastic environment factors.

In the system model, we consider that there is a server-observation function in virtualization layer. By partially observing each server's state, the system could obtain the present changes of network topology which is caused by broken server or servers. In this paper, we adopt the HPDM to observe whether the servers work well or not. Each server node in the physical networks sends a simple packet to the central monitor at regular intervals , and the central monitor checks if another new simple packet arrives within certain time interval for each node in real time. If $T_{now} - T_{last,i} > T_{o,i}$, the system considers that the server i fails to work at the present time, where T_{now} is the present time, $T_{last,i}$ is arrival time of last packet for server i, and $T_{o,i}$ is the maximum allowable arrival time interval.

Besides, for server i ($i \in N_1 \cup N_2$), the central monitor records the packet arrival times and time intervals between the arrival times of last packet and the latest packet when another new packet arrives. Apparently, every time the value of packet arrival time interval may change due to the dynamic network environment. These recorded arrival time intervals of server i make up a sample η_i with fixed size, and the sample is updated by adding newly recorded value of time interval and removing the oldest value. Let x_i be the time interval of packet arrival for server i.

Assumption 1: The value of x_i is subject to normal distribution based on the sample η_i. And the probability density function of x_i is defined as

$$f(x_i) = \frac{1}{\sqrt{2\pi}\sigma_i} e^{-\frac{(x_i-\mu_i)^2}{2\sigma_i^2}}$$

where μ_i and σ_i are the expectation and standard deviation of the sample η_i.

In HPDM, the maximum allowable time interval of packet arrival is set manually for each server. If another new packet of server i does not reach the central monitor within $T_{o,i}$, server i is considered to fail to work. In fact, in the sample η_i, some values may be larger than $T_{o,i}$ but the server actually is well-working at that time. Therefore, there may be some observation errors with HPDM when the system observes these actually well-working servers. Let $P_{e,i}$ be the probability of occurring observation error for server i. Especially, the $P_{e,i}$ also is the probability of $x_i > T_{o,i}$. With the $f(x_i)$, the $P_{e,i}$ is expressed as,

$$P_{e,i} = Pr[x_i > T_{o,i}] = \int_{T_{o,i}}^{+\infty} \frac{1}{\sqrt{2\pi}\sigma_i} e^{-\frac{(x_i-\mu_i)^2}{2\sigma_i^2}} dx \quad (1)$$

On the one hand, to improve the accuracy of observation, $T_{o,i}$ should be as large as possible and $P_{e,i}$ should be as small as possible. On the other hand, every time the central monitor needs to allocate the CPU resource to watch if another new packet arrives from server i in real time, especially, the allocated CPU resource for server i is occupied until another new packet arrives within the time of $T_{o,i}$ or occupied during the time of $T_{o,i}$. Let Φ_i be the maximum resource overhead of observing the server i and it is defined as $\Phi_i = T_{o,i} \cdot v_c^i$, where v_c^i represents the central monitor's CPU resource allocated to observe server i. To improve the resource utilization of central monitor, Φ_i should be as small as possible. Therefore, the tradeoff between the performance and the maximum overhead of observation should be considered. We define the tradeoff Θ_i for server i as $\Theta_i = \max\limits_{0<T_{o,i}\leq\Delta} \frac{1-P_{e,i}}{\Phi_i}$, where Δ is the interval between two adjacent time slot. The optimal maximum allowable time interval for server i is defined as

$$\overline{T}_{o,i} = \arg\max_{0<T_{o,i}\leq\Delta} \frac{1-P_{e,i}}{\Phi_i} \quad (2)$$

And the $\overline{P}_{e,i}$ is equal to $Pr[x_i > \overline{T}_{o,i}]$.

In virtualization layer, there also is a resource orchestration function. In the proposed scheme, we only consider the joint allocation of virtual computing resource, virtual link resource and fronthaul resource for SFCs in RAN slices, and other resources' allocation will continue in future research. In addition, the amount of each kind of resource is discrete in this paper.

Let $\chi_{u,m}^c(t)$ be the set of computing resource allocation policy for VNFs which belong to the SFC $m(m \in \zeta_u)$ at time slot t,

$$\chi_{u,m}^c(t) = \{v_{f_{u,m}^j,n_k}^c(t) \cdot x_{f_{u,m}^j,n_k}^p | j \in F_{u,m}, n_k \in G_{f_{u,m}^j}\}$$

where $F_{u,m}$ is the set of VNFs which belong to the SFC m, $G_{f_{u,m}^j}$ is the set of server nodes which can instantiate VNF j, $v_{f_{u,m}^j,n_k}^c(t)$ represents the amount of computing resource allocated for VNF j in server n_k, and the binary variable $x_{f_{u,m}^j,n_k}^p = 1$ if and only if VNF j is instantiated on server n_k.

Let $\chi_{u,m}^b(t)$ be the set of bandwidth resource allocation policy for VNFs of SFC $m(m \in \zeta_u)$. Under the condition of up-link, the data flows go through a virtual DU and a virtual CU along the SFC m. It is noticed that after the data flows go through $f_{u,m}^{DU}$ which donates the last VNF of the virtual DU, fronthaul network provides the bandwidth resource for the data flows, so the system only allocates fronthaul resource for $f_{u,m}^{DU}$. And we consider that the system does not allocate bandwidth resource for the VNF $f_{u,m}^{CU}$ which sends the data flows to EPC under the condition of up-link. Therefore, the bandwidth allocation at time slot t consists of two parts, namely,

$$\chi_{u,m}^b(t) = \{v_{f_{u,m}^j,l}^b(t) \cdot y_{f_{u,m}^j,l}^p | j \in F_{u,m} - \{f_{u,m}^{DU}, f_{u,m}^{CU}\}, \\ l \in L_1 \cup L_2\} \cup \{v_{f_{u,m}^{DU}}^b(t)\}$$

where $v_{f_{u,m}^j,l}^b(t)$ denotes the amount of bandwidth resource allocated for VNF j on link l, the binary variable $y_{f_{u,m}^j,l}^p = 1$ if only if VNF j transmits the data flows through the link l, and $v_{f_{u,m}^{DU}}^b(t)$ is defined as the fronthaul resource allocated for the VNF $f_{u,m}^{DU}$.

III. PROBLEM FORMULATION FOR THE SFC DEPLOYMENT

Considering the particularity of SFC deployment in dynamic C-RAN, the problem of SFC deployment in RAN slices is defined as three decisions in order. The first decision is repair decision. In the decision, the system decides whether the broken servers which are observed by HPDM need to be repaired or not. Let $a_\alpha(t)$ be the repair decision at time slot t. If $a_\alpha(t) = 1$, all the broken servers would be repaired, which leads to the server-repair cost. If $a_\alpha(t) = 0$, the broken servers keep stopping working. The second decision is selection decision. In this decision, each RAN slice chooses a method of VNF placement to adapt to changes of topology. $a_\beta(t)$ denotes the set of selection decision at time slot t, that is, $a_\beta(t) = \{\beta_u(t)|u \in U\}$, where the $\beta_u(t)$ represents the option of VNF placement for RAN slice u, and let A_β be the set of all possible selection decisions. The third decision is resource allocation decision, and $A_\chi(t) = (\chi^c_{u,m}(t), \chi^b_{u,m}(t)|m \in \zeta_u, u \in U)$ is defined as the decision on resource allocation at time slot t. Besides, let A_χ be the set of the whole possible resource allocation decisions.

We consider the relationship between the repair decision and resource allocation decision. When the system decides to repair the broken server or servers, the more available computing and bandwidth resources can be allocated to all SFCs than not repairing, which may reduce the total delay. However, the action of repairing broken server or servers also leads to the server-repair cost. For operators, the total delay should be as low as possible while the enough low server-repair cost also must be realized. Therefore, the tradeoff between the total delay and server-repair cost is the objective in this paper.

On the one hand, we consider the server-repair cost which repair decision results in. Let $R_1(t)$ be the server-repair cost at time slot t. And it is defined as $R_1(t) = c(\sum_{i \in N_1} \varrho_i(t) + \sum_{j \in N_2} \varrho_j(t))$, where binary variable $\varrho_i(t) = 1$ if and only if server i which fails to work is repaired at time slot t, c denotes the repair cost of each server, and $|N_1|$ represents the number of elements in set N_1.

On the other hand, the system total delay includes all the SFCs' queuing delay and scheduling delay from DU to CU.

Assumption 2: The amount of data packets arrival in SFC m ($m \in \zeta_u$) is subject to non-homogeneous Poisson distribution. The average arrival rate is the function of time slot t, and it is defined as $\lambda_{u,m}(t)$ at time slot t. Let $w_{u,m}(t)$ be the number of data packets which reach SFC m from UEs at time slot t. Besides, let $Pr[w_{u,m}(t) = n]$ be the probability of $w_{u,m}(t) = n$, and it is expressed as

$$Pr[w_{u,m}(t) = n] = e^{-\overline{\lambda}_{u,m}} \frac{(\overline{\lambda_{u,m}})^n}{n!} \tag{3}$$

where $\overline{\lambda}_{u,m} = \int_t^{t+1} \lambda_{u,m}(t')dt'$.

Applying the Little law in queuing theory, the queuing delay of SFC $m (m \in \zeta_u)$ at time slot t is defined as $q_d^{u,m}(t) = q_{u,m}(t)/\lambda_{u,m}(t)$. The scheduling delay of SFC m at time slot t includes processing delay $p_d^{u,m}(t)$ and transmission delay $l_d^{u,m}(t)$. The processing delay $p_d^{u,m}(t)$ is defined as

$$p_d^{u,m}(t) = \sum_{j \in F_{u,m}} (\sum_{n_k \in G_{f^j_{u,m}}} x^p_{f^j_{u,m}, n_k} \cdot p^{n_k}_{f^j_{u,m}}(t)) \tag{4}$$

where $p^{n_k}_{f^j_{u,m}}(t)$ is the processing delay of VNF j on the server n_k at time slot t. And the transmission delay $l_d^{u,m}(t)$ is formulated as,

$$l_d^{u,m}(t) = \frac{D'_{u,m}(t)}{v^b_{f^{DU}_{u,m}}(t)} + \sum_{j \in F'_{u,m}} \frac{D_{f^j_{u,m}}(t)}{v^b_{f^j_{u,m}, l}(t)} \tag{5}$$

where $D_{f^j_{u,m}}(t)$ is the amount of data sending out from VNF j, especially, $D'_{u,m}(t)$ is the amount of data which need to transmit through fronthaul network for SFC m, and $F'_{u,m} = F_{u,m} - \{f^{DU}_{u,m}, f^{CU}_{u,m}\}$. Let $R_2^{u,m}(t)$ be the delay of SFC m ($m \in \zeta_u$) at time slot t, $R_2^{u,m}(t) = q_d^{u,m}(t) + p_d^{u,m}(t) + l_d^{u,m}(t)$. And the system total delay function $R_2(t)$ is defined as

$$R_2(t) = \sum_{u \in U} \gamma_u \sum_{m \in \zeta_u} R_2^{u,m}(t) \tag{6}$$

where γ_u is the priority of slice u, and ω is a normalizing parameter which is larger than any maximum delay.

Considering the tradeoff between the total delay and server-repair cost, we define utility as the weight sum of the total delay and server-repair cost. Due to the two metrics have different units, we consider normalization of the total delay and server-repair cost respectively. The normalization methods are as follows:

The normalized server-repair cost $R'_1(t)$ is expressed as $R'_1(t) = \frac{R_1(t)}{(|N_1|+|N_2|)c}$, where $(|N_1| + |N_2|)c$ represents the cost of repairing all the servers and it is a positive cost constant. The normalized total delay $R'_2(t)$ is defined as $R'_2(t) = \frac{R_2(t)}{\omega}$, where ω is a positive delay constant which is enough large.

After normalization, the normalized total delay and normalized server-repair cost both have no units. And the weight sum of $R'_1(t)$ and $R'_2(t)$ could be realized reasonably. Besides, our objective is to maximize the utility. Therefore, the utility is expressed as equation (7),

$$R(t) = -(e_1 R'_1(t) + e_2 R'_2(t)) \tag{7}$$

where e_1 and e_2 are the positive weights and satisfy $e_1 + e_2 = 1$. Moreover, in order to ensure SFC deployment is effective, some constraints are necessary to be satisfied:

$$x^p_{f^j_{u,m}, n_k} = \sum_{n_k = l.head} y^p_{f^j_{u,m}, l} \quad \forall j \in F_{u,m}, u \in U, m \in \zeta_u \tag{8}$$

$$x^p_{f^j_{u,m}, n_k} = \sum_{n_k = l.tail} y^p_{f^{j-1}_{u,m}, l} \quad \forall j \in F_{u,m}, u \in U, m \in \zeta_u \tag{9}$$

where $l.head$ and $l.tail$ denote the start and end point of link l. Constraint (8) and (9) indicate that in a SFC two adjacent VNFs which are not on the same server must be instanced on two adjacent server nodes separately.

$$\sum_{n_k \in G_{f^j_{u,m}}} x^p_{f^j_{u,m}, n_k} = 1 \quad \forall j \in F_{u,m},\ u \in U, m \in \zeta_u \tag{10}$$

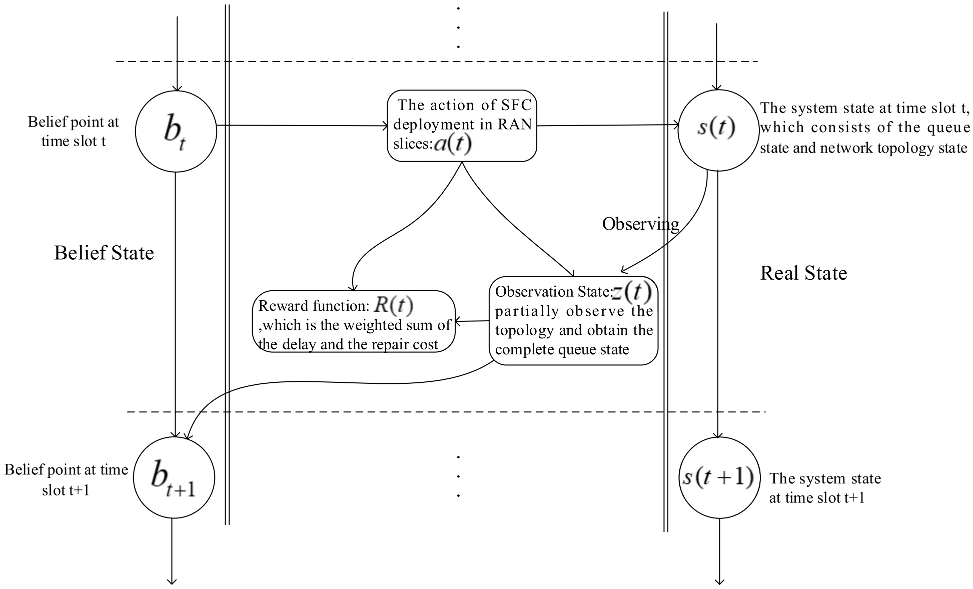

Belief point at time slot t
b_t
The action of SFC deployment in RAN slices: $a(t)$
$s(t)$
The system state at time slot t, which consists of the queue state and network topology state
Observing
Belief State
Real State
Reward function: $R(t)$,which is the weighted sum of the delay and the repair cost
Observation State: $\Xi(t)$ partially observe the topology and obtain the complete queue state
Belief point at time slot t+1
b_{t+1}
$s(t+1)$
The system state at time slot t+1

FIGURE 3. POMDP structure of SFC deployment in RAN slices.

$$\sum_{l\in L_1\cup L_2} y^p_{f^j_{u,m},l} \le 1 \quad \forall j\in F_{u,m},\ u\in U, m\in\zeta_u \tag{11}$$

Constraint (10) ensures that any VNF j only chooses one server to be instantiated, and (11) ensures that at most one link is selected for any VNF j.

$$\sum_{u\in U}\sum_{m\in\chi_u}\sum_{j\in F_{u,m}} v^c_{f^j_{u,m},n_k}\cdot x^p_{f^j_{u,m},n_k} \le v_{n_k} \quad \forall n_k\in N_1\cup N_2 \tag{12}$$

$$\sum_{u\in U}\sum_{m\in\chi_u}\sum_{j\in F_{u,m}} v^c_{f^j_{u,m},l}\cdot y^p_{f^j_{u,m},l} \le v_l \quad \forall l\in L \tag{13}$$

Constraint (12) ensures that the amount of computing resource occupied by VNFs must be less than the capacity of computing resource on any server, and (13) ensures that the amount of bandwidth resource which is allocated to the VNFs is less than capacity of link l.

IV. POMDP-BASED SFC DEPLOYMENT SCHEME

In this section, the POMDP framework is adopted to formulate the problem of SFC deployment in RAN slices. The system obtains the observable NTI by observing each server in physical network. With HPDM, the system knows the work state of server i by observing whether another simple packet from server i reaches the central monitor within the threshold of arrival time interval. If the simple packet does not arrive within the threshold, the server i is considered to be broken at the present time. Therefore, the server i may be considered to be broken with the probability $P_{e,i}$ while the server actually works well. Since not all servers' real states in physical network are obtained due to the observation errors, the system only obtains partial NTI. Therefore, the POMDP is used to estimate the whole real network topology condition. Then a POMDP scheme for SFC deployment in RAN slices is proposed to determine the deployment policy to maximize the utility according to the QSI and partial NTI. The diagram of POMDP framework for SFC deployment in RAN slices is shown in Fig.3. Specifically, the POMDP formulation of the SFC deployment includes the following components:

A. STATE SPACE

The state space is defined as $S = \{(q,\varphi)|q \in Q, \varphi \in \varphi'\}$, q is the queue state of system and $Q = \{q_{1,1}, q_{1,2}, \cdots, q_{|U|,|\zeta_u|}|q_{u,m} \in \{0,1,2,\cdots,n_q\}\}$ denotes the queue state space, where $q_{u,m}$ is the queue length of SFC $m(m \in \zeta_u)$; The topology state φ includes all servers' working states, so the topology state space is defined as $\varphi' = \{(\varphi_1, \varphi_2, \cdots \varphi_{|N_1|}, \varphi_{|N_1|+1}, \cdots, \varphi_{|N_1|+|N_2|})|\varphi_i \in \{0,1\}\}$, where $\varphi_i = 1$ represents the server i works well and $\varphi_i = 0$ indicates the server i fails to work.

B. ACTION SPACE

Let $A = \{(a_\alpha, a_\beta, a_\chi)|a_\alpha \in \{0,1\}, a_\beta \in A_\beta, a_\chi \in A_\chi\}$ be the action space, where a_α represents the repair decision, a_β is selection decision, and a_χ is the resource allocation decision.

C. TRANSITION PROBABILITY

With the action $a(t)$, the system transition probability from $s(t)$ to $s(t+1)$ is defined as $\tau(s(t), a(t), s(t+1))$. Because the system states include queue states and topology states, the system transition probability includes queue state transition probability $\tau(q(t), a(t), q(t+1))$ and topology state transition probability $\tau(\varphi(t), a(t), \varphi(t+1))$. Besides, the queue state and topology state are independent of each other.

Therefore, the system transition probability is expressed as

$$\tau(s(t), a(t), s(t+1)) = \tau(q(t), a(t), q(t+1)) \cdot \tau(\varphi(t), a(t), \varphi(t+1)) \quad (14)$$

In this paper, the queue state consists of all SFCs' queue states which are independent of each other. The relationship between $q_{u,m}(t+1)$ and $q_{u,m}(t)$ is expressed as $q_{u,m}(t+1) = w_{u,m}(t) + q_{u,m}(t) - v_{u,m}(t)\Delta$, where $w_{u,m}(t)$ is the number of data packets arrival in SFC m at time slot t, and $v_{u,m}(t)$ is the data service rate of SFC m. Especially, the $v_{u,m}(t)$ is defined as $v_{u,m}(t) = min(d_{f^{\bar{j}}_{u,m}}, l_{f^{\bar{l}}_{u,m}})$, where $d_{f^{\bar{j}}_{u,m}}$ is the capacity of the data processing on VNF $f^{\bar{j}}_{u,m}$ which has the least computing resource among all the VNFs in SFC m, and $l_{f^{\bar{l}}_{u,m}}$ denotes the capacity of the data transmission on link $f^{\bar{l}}_{u,m}$ which has the least bandwidth resource among all the links in SFC m. With the action $a(t)$, the data service rate could be known, but the amount of data arrival is uncertain and it is subject to non-homogeneous Poisson distribution. Therefore, the transition probability from $q_{u,m}(t)$ to $q_{u,m}(t+1)$ is related to the probability of the number of data arrival. And $\tau(q_{u,m}(t), a(t), q_{u,m}(t+1))$ is defines as

$$\tau(q_{u,m}(t), a(t), q_{u,m}(t+1)) = Pr[w_{u,m}(t)] \quad (15)$$

where $w_{u,m}(t) = q_{u,m}(t+1) - q_{u,m}(t) + v_{u,m}(t)\Delta$. If $w_{u,m}(t) < 0$, $Pr[w_{u,m}(t)] = 0$. When $w_{u,m}(t) \geq 0$ the $Pr[w_{u,m}(t)]$ is calculate by equation (3). In this way, $Pr[w_{u,m}(t)]$ is expressed as

$$Pr[w_{u,m}(t)] = \begin{cases} 0 & \text{others} \\ e^{-\overline{\lambda}_{u,m}} \dfrac{(\overline{\lambda}_{u,m})^{w_{u,m}(t)}}{w_{u,m}(t)!} & \text{if } w_{u,m}(t) \geq 0 \end{cases} \quad (16)$$

where $\overline{\lambda}_{u,m} = \int_t^{t+1} \lambda_{u,m}(t')dt'$. Therefore, the queue state transition probability is expressed as

$$\tau(q(t), a(t), q(t+1)) = \prod_{u \in U} \prod_{m \in \chi_u} Pr[w_{u,m}(t)] \quad (17)$$

Analogously, the transition probability of topology state includes the transition probability of each server's working state. Let $\tau(\varphi_i(t), a(t), \varphi_i(t+1))$ be the transition probability from $\varphi_i(t)$ to $\varphi_i(t+1)$ for server i. When the action $a(t)$ includes the action of repairing the broken server or servers, server i must work normally at $t+1$ time slot, and the transition probability is expressed as

$$\tau(\varphi_i(t), a(t), \varphi_i(t+1)) = \begin{cases} 1 & \text{if } \varphi_i(t+1) = 1 \\ 0 & \text{others} \end{cases} \quad (18)$$

When the action $a(t)$ includes the action of not repairing the broken server or servers, the broken server i_1 still can not work at time slot $t+1$. And the transition probability of server i_1 is expressed as

$$\tau(\varphi_{i_1}(t), a(t), \varphi_{i_1}(t+1)) = \begin{cases} 1 & \text{if } \varphi_{i_1}(t+1) = 0 \\ 0 & \text{others} \end{cases} \quad (19)$$

However, the normal server i_2 may break down at the next time slot with a certain probability. In this paper, p_{i_2} denotes the probability of failing to work for normal server i_2 and it is set according to its environment condition. Thus, the transition probability of server i_2 is expressed as

$$\tau(\varphi_{i_2}(t), a(t), \varphi_{i_2}(t+1)) = \begin{cases} p_{i_2} & \text{if } \varphi_{i_1}(t+1) = 0 \\ 1 - p_{i_2} & \text{others} \end{cases} \quad (20)$$

Therefore, the transition probability of topology state is defined as

$$\tau(\varphi(t), a(t), \varphi(t+1)) = \prod_{i \in N_1 \cap N_2} \tau(\varphi_i(t), a(t), \varphi_i(t+1)) \quad (21)$$

Finally, the system transition probability $\tau(s(t), a(t), s(t+1))$ is expressed as

$$\tau(s(t), a(t), s(t+1)) = \prod_{u \in U} \prod_{m \in \chi_u} Pr[w_{u,m}(t)] \cdot \prod_{i \in N_1 \cap N_2} \tau(\varphi_i(t), a(t), \varphi_i(t+1)) \quad (22)$$

D. REWARD

Since we concentrate on maximizing the utility, the immediate reward with the action $a(t)$ and system state $s(t)$ is defined as

$$R(s(t), a(t)) = -(e_1 R'_1(t) + e_2 R'_2(t)) \quad (23)$$

E. OBSERVATION SPACE

The observation space is expressed as $Z = \{(z_q, z_\varphi) | z_q \in Z_Q, z_\varphi \in Z_\varphi\}$, where z_q is the observable queue state, Z_Q is observable queue state space, z_φ is the observable topology state, and Z_φ is observable topology state space.

F. OBSERVATION FUNCTION

With the observation result $z(t)$ after the action $a(t)$, the observation function $o(z(t), a(t), s(t+1))$ is described as

$$o(z(t), a(t), s(t+1)) = o(z_q(t), a(t), q(t)) \cdot o(z_\varphi(t), a(t), \varphi(t+1)) \quad (24)$$

Because the whole queue states could be completely observed, $o(z_q(t), a(t), q(t)) = 1$ if and only if $z_q(t) = q(t)$, otherwise, $o(z_q(t), a(t), q(t)) = 0$.

$$o(z_\varphi(t), a(t), \varphi(t+1)) = \prod_{i \in N_1} pr[z_{\varphi_i}(t) | a(t), \varphi_i(t+1)] \cdot \prod_{j \in N_2} pr[z_{\varphi_j}(t) | a(t), \varphi_j(t+1)] \quad (25)$$

where $z_{\varphi_i}(t)$ denotes the observation result for server i at time slot t. If all broken servers are repaired at time slot t, $o(z_\varphi(t), a(t), \varphi(t+1)) = 1$. Conversely, with the condition that the system decide not repair them, $Pr[z_{\varphi_i}(t)|a(t),$

$\varphi_i(t+1)] = 1$ if and only if $z_{\varphi_i}(t) = 0$ when $\varphi_i(t+1) = 0$; When $\varphi_i(t+1) = 1$, $Pr[z_{\varphi_i}(t)|a(t), \varphi_i(t+1)]$ is expressed as

$$Pr[z_{\varphi_i}(t)|a(t), \varphi_i(t+1)] = \begin{cases} 1 - \overline{P}_{e,i} & \text{if } z_{\varphi_i}(t) = 1 \\ \overline{P}_{e,i} & \text{if } z_{\varphi_i}(t) = 0 \end{cases} \quad (26)$$

G. BELIEF STATE

POMDP makes the present decisions based on the historical informations including the whole actions and observations in the past. And these historical informations are defined as a belief state. The belief state vector b_t contains the statistical informations before the time slot t. Besides, b_t is also equivalent to a statistic distribution over the state space S at time slot t. Let $b_t(s)$ be one element of vector b_t, and it corresponds to the probability of state s at time slot t. All elements in b_t must satisfy $\sum_{s \in S} b_t(s) = 1$. As explained in Fig 3, the updated belief state could be obtained as the outcome of state estimator, which consists of the inputs of action, observation and previous belief state. Therefore, the $b_{t+1}(s)$ could be updated by the Bayesian rule as

$$b_{t+1}(s) = \frac{o(z(t), a(t), s) \cdot \sum_{s \in S} \tau(s, a(t), s) \cdot b_t(s)}{Pr[z(t)|b_t, a(t)]} \quad (27)$$

where $Pr[z(t)|b_t, a(t)]$ is the normalizing factor

$$Pr[z(t)|b_t, a(t)] = \sum_{s_2 \in S} o(z(t), a(t), s_2) \cdot \sum_{s_1 \in S} \tau(s_1, a(t), s_2) b_t(s_1) \quad (28)$$

H. OBJECTIVE

Give a deployment policy π which consists of a series of actions in order, the average utility $\rho(\pi)$ in POMDP is defined as

$$\rho(\pi) = E[\sum_{t=1}^{\infty} \gamma^t \sum_{s(t) \in S} R(s, \pi(t)) \cdot b_t(s)] \quad (29)$$

where $\pi(t)$ represents the action of deployment policy π at time slot t. Since the induced Markov chain is ergodic, there must be a unique steady SFC deployment policy to maximize the average utility. Therefore, the objective of SFC deployment is formulated as

$$\max_{\pi' \in \Omega} \rho(\pi') = E[\sum_{t=1}^{\infty} \gamma^t \sum_{s(t) \in S} R(s, \pi'(t)) \cdot b_t(s)] \quad (30)$$

where the Ω represents all possible SFC deployment policies.

The optimal policy is obtained by Bellman value-function iterations. Let the $V_k^*(b)$ be the optimal value function with belief point b in k^{th} iteration,

$$V_k^*(b) = \max_{a \in A}\{\gamma \sum_{z \in Z} Pr[z|b, a] \cdot V_{k-1}^*(b_a^z) + \sum_{s \in S} R(s, a) \cdot b(s)\} \quad (31)$$

where $b_a^z = \{b_a^z(s)|s \in S\}$,

$$b_a^z(s) = \frac{o(z, a, s) \cdot \sum_{s' \in S} \tau(s, a, s) \cdot b(s')}{Pr[z|b, a]} \quad (32)$$

And the corresponding optimal action $\pi_k^*(b)$ is described as

$$\pi_k^*(b) = \arg\max_{a \in A}\{\gamma \sum_{z \in Z} pr[z|b_t, a] \cdot V_{k-1}^*(b_a^z) + \sum_{s \in S} R(s, a) \cdot b(s)\} \quad (33)$$

V. MINGLED HEURISTIC VALUE ITERATION ALGORITHM

In Bellman iteration, each value function could be regarded as a hyper-plane with $|S| - 1$ dimensions in belief space. So it could be defined as a value function vector with $|S| - 1$ dimensions. Since the value functions have the nature of piecewise linear convexity in belief space, the optimal value function $V^*(b)$ of belief point b is given by $V^*(b) = \max_{\alpha \in \Gamma} \alpha \cdot b$, where Γ is the set of value-function vectors. Therefore, the Bellman iteration actually constantly updates the Γ until convergence.

To solve the Bellman iteration, complete value iteration (CVI) and point-based value iteration are two important methods. Since the CVI algorithm updates the Γ based on the whole belief space, it could obtain the optimal solution. And the procedures of updating the Γ in $(k+1)^{th}$ iteration are as follows:

Step 1, calculate Γ_0 which is the initial Γ. Let $\Lambda_a = (R(s_1, a), \cdots, R(s_{|S|}, a))$ be the one step reward vector with action a, and the Γ_0 is expressed as

$$\Gamma_0 = \{\Lambda_a | a \in A\} \quad (34)$$

Step 2, calculate the parameter $\Gamma_{k+1}^{a,z}$. The $\alpha^{a,z}$ is defined as $\alpha^{a,z} = (\alpha^{a,z}(s_1), \cdots, \alpha^{a,z}(s_{|S|}))$. In the $\alpha^{a,z}$, the i^{th} element $\alpha^{a,z}(s_i)$ is defined as $\alpha^{a,z}(s_i) = \gamma \sum_{s' \in S} \tau(s_i, a, s') o(z, a, s') \alpha_j(s')$, where $\alpha_j(s')$ is a vector in Γ_k which is the updated set of value-function vectors after k iterations. And the $\Gamma_{k+1}^{a,z}$ is expressed as

$$\Gamma_{k+1}^{a,z} = \{\alpha^{a,z} | \alpha_j(s') \in \Gamma_k\} \quad (35)$$

Step 3, calculate the parameter Γ_{k+1}^{a}. It is defined as $\Gamma_{k+1}^{a} = \Lambda_a \bigoplus \Gamma_{k+1}^{a,z_1} \bigoplus \cdots \bigoplus \Gamma_{k+1}^{a,z_{|Z|}}$, where $A \bigoplus B = \{a + b | a \in A, b \in B\}$.

Step 4, calculate the new set of value-function vectors Γ_{k+1}. The Γ_{k+1} is equal to $\bigcup_{a \in A} \Gamma_{k+1}^{a}$.

Step 5, when the set of value-function vectors is not convergence, back to step 1.

When Γ_{k+1} is obtained in $(k+1)^{th}$ iteration, in step 2 the complexity is equal to $O(|\Gamma_k||Z||S|^2|A|)$, and in step 3 the complexity is equal to $O(|\Gamma_k|^{|Z|}|S||A|)$. Therefore, the complexity of CVI algorithm is approximated as $O(|\Gamma_k||Z||S|^2|A| + |\Gamma_k|^{|Z|}|S||A|)$. Due to the exponential complexity, CVI algorithm has a very high complexity when the scale of POMDP is large.

Algorithm 1 Updating the Set of Reachable Belief Points

```
for each $b \in B^{pre}$ do
    Calculate the $suc(b)$ by equation (37)
    Calculate the furthest reachable belief point $b'$ by equation (38)
    $B_{sub} \leftarrow B_{sub} \cup \{b'\}$
end for
Set $v' \leftarrow \emptyset$
for each $b \in B_{sub}$ do
    Calculate the lower bound vector $\alpha_b$ by equation (39), (40)
    $v' \leftarrow v' \cup \{\alpha_b\}$
end for
Set $\underline{V} \leftarrow v'$
for each $b \in B_{sub}$ do
    $V_{cos} \leftarrow \{b | \exists s \in S, b(s) = 1\}$
    $v_b^0 \leftarrow \sum_{b' \in V_{cos}} v(b') \cdot b$
    for each $< b_i, \overline{v}_i > \in B_{sub} - V_{cos}$ do
        $c(b_i) \leftarrow \min_{s \in S} \frac{b(s)}{b_i(s)}$
        $f(b_i) \leftarrow \overline{v}_i - \sum b' \in V_{cos} v(b') \cdot b_i(s)$
    end for
    $\overline{v} \leftarrow v_b^0 + \min_i c(b_i) f(b_i)$
    Add new pair $< b, \overline{v} >$ to the set $\overline{V}$
end for
```

In this paper, the point-based mingled heuristic value iteration (MHVI) algorithm is used to deal with the Bellman iteration approximately due to the low complexity. The algorithm does not depend on the whole belief space, and updates Γ based on B_{sub} which is the set of partial reachable belief points. Thus, MHVI algorithm comprises of two processes, one is the process of updating B_{sub}, the other is the process of updating Γ based on updated B_{sub}. Specifically, two processes are executed circularly until convergence criterion is satisfied.

A. UPDATING THE SET OF REACHABLE BELIEF POINTS

The process of updating the set of reachable belief points is shown in algorithm 1.

Let B_{sub} be the set of reachable belief points, and it is initialized to be $\{b_0\}$, where b_0 is the initial belief point. The process of updating the set of reachable belief points as follows:

First, these belief points whose difference between upper bound and lower bound of value functions is over threshold compose the set B^{pre}

$$B^{pre} = \{b | \overline{V}(b) - \underline{V}(b) \geq \frac{\xi}{\gamma^{h_b}}, b \in B_{sub}\} \tag{36}$$

where γ^{h_b} is the level number of b in the belief tree; $\underline{V}(b)$ is the is the lower bound of b. And it is defined as $\underline{V}(b) = \max_{\alpha \in \underline{V}} b \cdot \alpha$, where $\underline{V}$ is composed of several lower bound vectors; $\overline{V}(b)$ is the upper bound of b, where $\overline{V}$ is the set of upper bound vectors.

Next, the set $suc(b)$ is composed of several reachable points from b ($b \in B^{pre}$),

$$suc(b) = \{b' | b' = b_a^z, \overline{V}(b') - \underline{V}(b') \geq \frac{\xi}{\gamma^{h'_b}}, \forall a \in A, z \in Z\} \tag{37}$$

The distance between the point b_1 in $suc(b)$ and the set B_{sub} is defined as

$$||b_1 - B_{sub}|| = \min_{b^* \in B_{sub}} |\sum_{s \in S} |b_1(s) - b^*(s)|| \tag{38}$$

And the point which is farthest from B_{sub} would be added to B_{sub}.

Then, new lower bound vectors are added to the set $\underline{V}$. The lower bound vector α_b with another new point $b(b \in B_{sub})$ is defined as

$$\alpha_b(s) = \max_{a \in A}\{\sum_{s \in S} R(s, a) b(s) + \gamma \sum_{z \in Z} \max_{\alpha \in \underline{V}}\{\sum_{s \in S} \tau(s, a, s) O(z, a, s) b_a^z(s) \alpha\}\} \tag{39}$$

$$\alpha_b = (\alpha_b(s_1), \alpha_b(s_1), \cdots, \alpha_b(s_{|S|})) \tag{40}$$

Finally, $\overline{V}$ also needs to be updated. A belief point b_i and corresponding $\overline{v}_i$ make a pair defined as $(b_i, \overline{v}_i)$, where $\overline{v}_i$ is the upper bound of value function of b_i. And $\overline{V}$ is composed of several such pairs. To update the $\overline{V}$, new pairs need to be calculated. The upper bound $\overline{v}_i$ of another new point b_i is expressed as

$$\overline{v}_i = \max_{a \in A}\{\sum_{s \in S} R(s, a) b(s) + \gamma \sum_{z \in Z} Pr[z|b, a] \overline{V}(b_a^z)\} \tag{41}$$

B. UPDATING THE SET OF VALUE-FUNCTION VECTORS

Let Γ_k be the set of value-function vectors after k iterations. After updating the B_{sub} in $(k+1)^{th}$ iteration, the set Γ_k needs to update, as shown in algorithm 2.

First, calculate $\Gamma_0 =$ by equation (34).

Then, calculate $\Gamma_{k+1}^{a,z}$ by equation (35), and the parameter $\Gamma_{k+1,b}^{a}$ for b ($b \in B_{sub}$) is given by

$$\Gamma_{k+1,b}^{a} = \Lambda_a + \sum_{z \in Z} \arg\max_{\alpha \in \Gamma_{k+1}^{a,z}} b \cdot \alpha \tag{42}$$

All the actions in action space compose of the set $\Gamma_{k+1,b}$ for b ($b \in B_{sub}$), and the set is defined as $\Gamma_{k+1,b} = \cup_{a \in A} \Gamma_{k+1}^{a,b}$.

Next, the *update*(b) for belief point b is defined as $update(b) = \arg\max_{\alpha \in \Gamma_{k+1,b}} b \cdot \alpha$.

Finally, the new set of value-function vectors after $k+1$ iterations is defined as $\Gamma_{k+1} = \cup_{b \in B_{sub}} update(b)$

C. MINGLED HEURISTIC VALUE ITERATION ALGORITHM

Noticing that the sets of upper bound and lower bound demand to be initialized, the MHVI algorithm first considers the two initializations. Let $\underline{v}(s, a)$ be the lower bound parameter with action a and system state s,

$$\underline{v}(s, a) = \min_{a', s'} \frac{R(s', a')}{1 - \gamma} \tag{43}$$

Algorithm 2 Updating the Set of Value-Function Vectors

```
for each $b \in B_{sub}$ do
    Set $\Gamma_{k+1} \leftarrow \emptyset$
    for each $a \in A$ do
        Set $\Gamma_{k+1,b} \leftarrow \emptyset$
        for each $z \in Z$ do
            Calculate the set $\Gamma_{k+1}^{a,z}$ by equation (35)
            Calculate $\Gamma_{k+1,b}^{a}$ by equation (42)
        end for
        $\Gamma_{k+1,b} \leftarrow \Gamma_{k+1,b} \cup \{\Gamma_{k+1,b}^{a}\}$
        Set $update(b) = \arg\max_{\alpha \in \Gamma_{k+1,b}} b \cdot \alpha$
    end for
    $\Gamma_{k+1} \leftarrow \Gamma_{k+1} \cup \{update(b)\}$
end for
```

And $\underline{v}(s,a)$ is constantly updated until it converges, and the update equation is defined as

$$\underline{v}(s,a) = R(s,a) + \gamma \sum_{s' \in S} \tau(s,a,s')\underline{v}(s',a) \tag{44}$$

When the update is completed, these convergent values form the vector $\underline{v}(a) = (\underline{v}(s_1,a), \underline{v}(s_2,a), \cdots, \underline{v}(s_{|S|},a))$, and the initial lower bound set is expressed as

$$\underline{V} = \{\underline{v}(a) | a \in A\} \tag{45}$$

Similarly, the $\overline{v}(s,a)$ is the upper bound parameter,

$$\overline{v}(s,a) = \max_{a',s'} \frac{R(s',a')}{1-\gamma} \tag{46}$$

and the update equation for $\overline{v}(s,a)$ is defined as

$$\overline{v}(s,a) = R(s,a) + \gamma \sum_{z \in Z} \max_{a' \in A} \{\sum_{s' \in S} \tau(s,a,s') \\ o(z,a.s')\overline{v}(s',a')\} \tag{47}$$

With these convergent values, the initial $\overline{V}$ eventually is obtained.

When the sets of upper and lower bound are initialized, MHVI algorithm starts to iterate to meet the convergence criterion, as the algorithm 3 shown. The convergence criterion is defined as $\overline{V}(b_0) - \underline{V}(b_0) \leq \eta$, where the η is the established threshold.

When the algorithm 1 updates the set of reachable belief points, its complexity is approximated as $O(|B_{sub}||A||Z|)$. Then the algorithm 2 updates the set of value-function vectors based on updated B_{sub}, its complexity is calculated as $O(|S|^2|A||Z||B_{sub}|^2)$. Therefore, the complexity of MHVI algorithm in an iteration is estimated as $O(|B_{sub}||A||Z| + |S|^2|A||Z||B_{sub}|^2)$. Comparing the complete value iteration, MHVI algorithm significantly reduces the complexity, especially, it avoids the exponential complexity.

VI. SIMULATION RESULTS AND DISCUSSION

In this section, the simulation results are presented and some discussions on these results are described.

Algorithm 3 Mingled Heuristic Value Iteration Algorithm

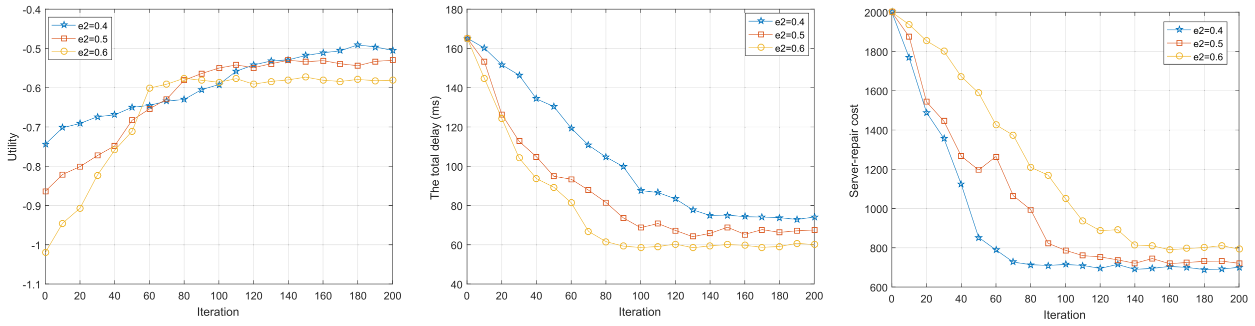
```
Input: $b_0$, $U$
Output: optimal policy $\pi^*$
Build the physical network topology $G = (N, L)$
Set $B_{sub} \leftarrow \{b_0\}$
Set $k = 0$
Calculate $\underline{v}(s,a)$ by equation (43)
Update $\underline{v}(s,a) \forall s \in S, a \in A$ by equation (44)
Initialize $\underline{V}$ by equation (45)
Calculate $\overline{v}(s,a)$ by equation (46)
Update $\overline{v}(s,a) \forall s \in S, a \in A$ by equation (47)
Initialize $\overline{V}$
while dissatisfy convergence criterion do
    Update $B_{sub}$ by algorithm 1
    Update $\Gamma_{k+1}$ by algorithm 2
    $k++$
end while
Obtain the approximately optimal deployment policy $\pi^*$
    by equation (33)
```

A. SIMULATION SETUP

In order to evaluate the performance of proposed POMDP scheme and compare with present SFC deployment schemes, several simulations for SFC deployment are necessary. Two physical networks for DU pool and CU pool respectively are simulated to be two independent undirected graphs whose topology both are initialized randomly. In the simulations, the undirected graph corresponding to DU pool has 16 nodes and the other has 23 nodes. The number of CPU cores on a node represents the node's computing resource. In the simulation there are two kinds of servers. One kind of servers have 4 CPU cores, the other kind of servers have 8 CPU cores. The bandwidth capacity of each link is set as 10M in the simulation. Each node's allowable maximum packet arrival time interval is set by equation (2). The maximum allowed queue length is set to be 20 packets. Besides, the capacity of fronthaul bandwidth resource is 260M.

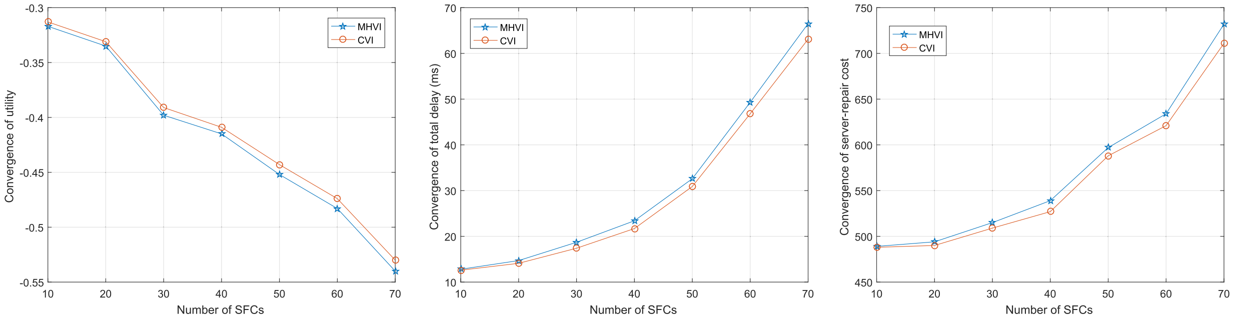
In the simulation scene, there are three RAN slices, and the total of SFCs in three RAN slices ranges form 10 to 70. The proportion of SFCs in slice 1 is 50%, the proportion of SFCs in slice 2 is 30%, and the proportion of SFCs in slice 3 is 20%. For example, when the total of SFCs in three slices is 70, the slice 1 contains 35 SFCs, the slice 2 has 21 SFCs, and slice 3 consists of 14 SFCs. The amount of each SFC's arrival data is subject to non-homogeneous Poisson distribution with different time-varying arrival rate. Specifically, each SFC's arrival rate in slice 1 is higher than it in slice 2, and each SFC's arrival rate in slice 2 is higher than it in slice 3. In addition, we define that each SFC has 9 VNFs in slice 1. In the slice 2 and slice 3, each SFC is set to have 10 VNFs in order.

In order to improve the accuracy of simulation, the number of iterations is set to be 200 and the final simulation results are the expectation of several simulation results after several repeated simulations. Moreover, all the simulations have been



(a) (b) (c)

FIGURE 4. Performance comparison of different weight values. (a) Utility. (b) Total delay. (c) Server-repair cost.



(a) (b) (c)

FIGURE 5. Performance comparison between CVI and MHVI. (a) Utility. (b) Total delay. (c) Server-repair cost.

executed on a machine characterized by 3.40 GHz Intel i7-6700 processor and by an 8 GB memory.

B. COMPARISON OF DIFFERENT WEIGHT VALUES

We change the weight e_2 as $e_2 = 0.4$, $e_2 = 0.5$ and $e_2 = 0.6$ respectively, and make the simulations with different weight values. The total number of iteration is set as 200, and the number of SFCs is equal to 70. The simulation results are shown in Fig.4.

In Fig.4, the convergence value of utility gradually becomes larger when the weight e_2 increases from 0.4 to 0.6. Besides, the growth of e_2 could accelerate the convergence speed of utility. When e_2 is equal to 0.6, the convergence speed is the fastest among the three cases, which indicates that the total delay has more effect on the utility than server-repair cost.

With the weight e_2 rising from 0.4 to 0.6, convergence speed of the total delay becomes faster, and the convergence value becomes smaller. However, the convergence speed of the server-repair cost is slowed down when the weight e_2 increases, and its convergence value becomes larger. These simulation results indicate that different weight values indeed affect the convergence performance.

C. COMPARISON WITH CVI ALGORITHM

CVI and MHVI algorithm are simulated respectively to deal with the Bellman iteration. The total of SFCs ranges from 10 to 70, and the weight $e_2 = 0.5$. The performance comparison between CVI and MHVI is shown in Fig.5 and Fig.6.

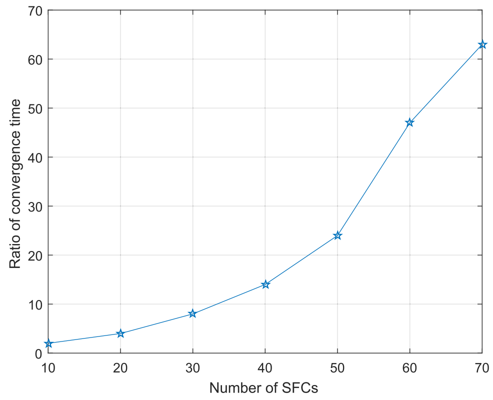
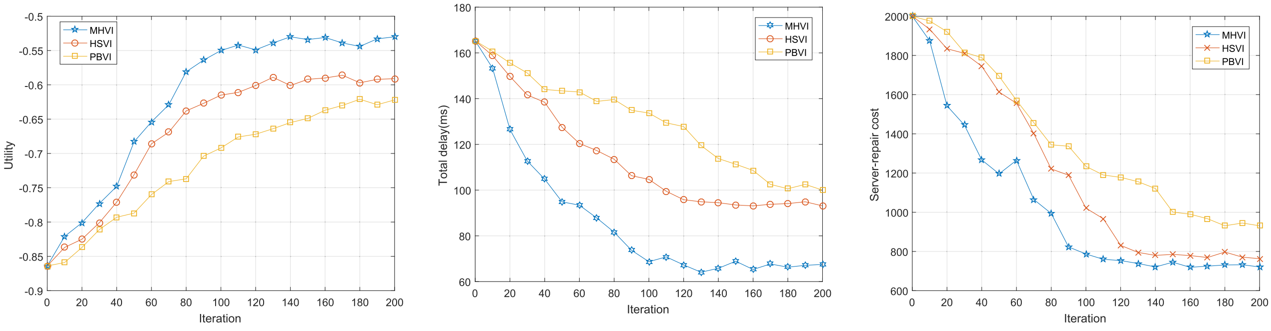


FIGURE 6. Ratio of T_e^1 to T_e^2.

In Fig.5, CVI algorithm is used to obtain the optimal solution while MHVI algorithm is applied to get the approximate solution. The solution includes the convergence values of utility, the total delay and server-repair cost. When the scale of SFCs is small, the approximate convergence values of MHVI algorithm are very close to the optimal convergence values of CVI. With the number of SFCs increasing, convergence values of MHVI algorithm gradually deviate slightly from the optimal convergence, but the differences between two



(a) (b) (c)

FIGURE 7. **Performance comparison of three point-based algorithms. (a) Utility. (b) Total delay. (c) Server-repair cost.**

solutions are small enough to have a little effect on practical application.

In this simulation, we also record the convergence time of two algorithms with certain number of SFCs. Let T_e^1 be the convergence time of CVI algorithm, and T_e^2 be the convergence time of MHVI algorithm. The ratios of T_e^1 to T_e^2 have been calculated with different number of SFCs, as Fig.6 shown. In Fig.6, the ratio is always larger than 1, which indicates that the complexity of CVI algorithm is indeed larger than it of MHVI. Besides, with the growth of the scale of SFCs, the ratio rapidly increases. When the scale of SFC becomes larger, the convergence time of CVI algorithm rises very fast due to its exponential complexity. However, the convergence time of MHVI algorithm grows slowly. Therefore, the ratio rapidly increases with the number of SFCs rising.

D. COMPARISON WITH TRADITIONAL POINT-BASED ALGORITHM

We make the simulations about the three point-based algorithms for POMDP, including MHVI, HSVI (heuristic search value iteration) and PBVI (point-based value iteration). The total of SFCs in three RAN slices is equal to be 70 and the e_2 is set as 0.5. The performance of three solutions is shown in Fig.7.

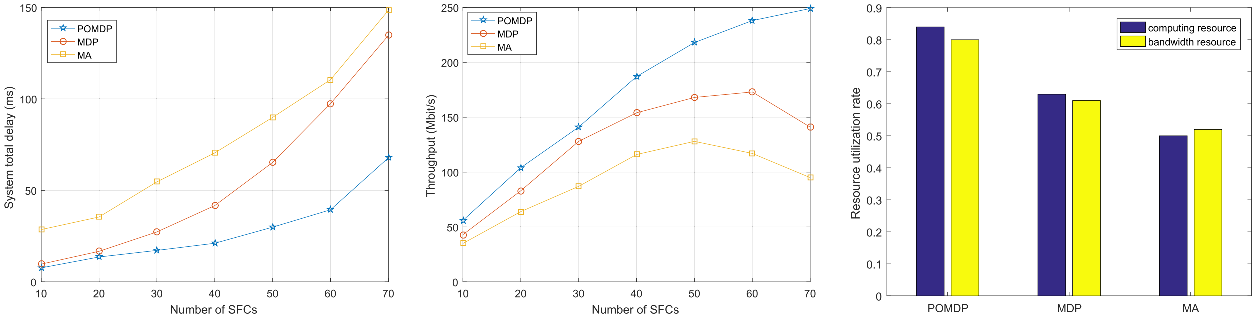
The simulation results indicate that the utility gradually increases to the convergence value with iterations, while total delay and server-repair cost both decrease until convergence. The convergence speed of PBVI is slowest among the three solutions, and the solution easily traps in local optimum. The reason of poor performance for PBVI is that it searches the reachable belief points based on the points density, which is an extremely inefficient search policy. Compared with PBVI, the HSVI accelerates the convergence speed and searches the reachable points according to the value functions, so it could improve the global searching ability and have a better convergence value. Especially, the MHVI searches effective points according to the distribution of belief points and value functions, which further improve global searching ability so as to it can searches the reachable points deeply and widely. Besides, the searching policy is efficient and it could guarantee enough convergence speed. Therefore, the performance of MHVI is best among the three solutions.

E. COMPARISON WITH EXISTING SCHEMES

We compare the performance of POMDP scheme, MA scheme introduced in [15], and MDP scheme introduced in [16]. The total of SFCs ranges from 10 to 70. Every time the number of SFCs changes, POMDP, MDP and MA scheme are simulated respectively. Specifically, we evaluate the performance of the three schemes in three aspects, including total delay, throughput and resource utilization.

1) The system delay of three schemes is shown in Fig.8 (a). Considering the topology is completely observed, the MDP scheme and MA scheme could not make use of some available network resources on some servers which are directly considered to be broken according to the observation results. But the POMDP scheme estimates the real topology condition according to the partial observation results, and could find more available network resources than other two scheme. Thus the POMDP scheme has the lowest system delay among the three schemes. Besides, due to the difference of decision-making mechanism between MDP and MA, the MDP has lower total delay than MA scheme. However, when the number of SFCs is over 50, the MDP scheme's system delay rapidly increases and seems to be close to the latency of MA scheme.

2) The variation of throughput is shown in Fig.8 (b). Based on realistic topology, the POMDP scheme leads to the highest throughput among the three schemes. When the number of SFCs is relatively small, the throughput increases quickly. The throughput growth has slowed down when the total of SFCs is over 50. With the policy of MDP, the network throughput is lower than it in POMDP scheme. Especially, when the total of SFC is over 70, the throughput rapidly drops off. The MA scheme has the lowest throughput, and it rises slowly until the total of SFCs is over 50. Therefore, the simulation result indicates that the POMDP scheme could improve the throughput in dynamic network.



(a)

(b)

(c)

FIGURE 8. Performance comparison of three SFC deployment schemes. (a) Total delay. (b) Throughput. (c) Resource utilization.

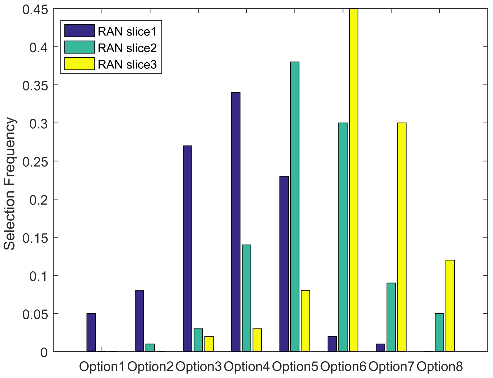


FIGURE 9. Selection comparison of three RAN slices.

3) The resource utilization of three schemes are described in Fig.8 (c). Apparently, the utilization ratios of computing and bandwidth resource for POMDP scheme both are the highest among the three schemes, which indicates that POMDP scheme could take advantage of the network resources by estimating the real topology. Disable to deal with the observation errors, the MDP scheme has a lower resource utilization ratio. The MA scheme easily wastes resources, so its utilization ratio is lowest. In short, the POMDP scheme indeed improves the resource utilization.

F. SELECTION DECISION

Finally, we discuss the selection decisions of three RAN slices. The total of SFCs is set equal to 70, and the weight e_2 is set as 0.5. The MHVI algorithm is simulated to obtain the approximately optimal policy for SFC deployment in three RAN slices. In the approximately optional deployment policy, the frequency of each method of VNF placement chose by each slice is calculated.

As shown in Fig.9, the slice 1 with massive arrival data and strict delay requirement prefers choosing the optional 3 and 4, which means a lot of VNFs of SFCs placed in DU to reduce resource consumption of fronthaul network. Since the slice 2 has less arriving data than slice 1, which results that the requirement of frontal network resource tails off, it prefer choosing option 5 and 6. In slice 3, there are the least traffics through the slice among the three RAN slices. So the slice 3 tends to choose option 6 to reduce the resource consumption of DU pool whose computing resource and bandwidth resource are less than CU pool.

VII. CONCLUSION

In this paper, the problem of SFC deployment in RAN slices has been defined as a series of deployment decisions. Partially observing the NTI by HPDM, the POMDP-based scheme was formulated to maximize the utility associated with the total delay and server-repair cost. And the MHVI algorithm was proposed to obtain approximately optimal SFC deployment policy with lower complexity. Simulation results indicated that the POMDP-based SFC deployment scheme could improve the system delay, throughput and resource utilization.

REFERENCES

[1] S. Sharma, R. Miller, and A. Francini, "A cloud-native approach to 5G network slicing," *IEEE Commun. Mag.*, vol. 55, no. 8, pp. 120–127, Aug. 2017, doi: 10.1109/MCOM.2017.1600942.

[2] H. Zhang, N. Liu, X. Chu, K. Long, A. Aghvami, and V. C. M. Leung, "Network slicing based 5G and future mobile networks: Mobility, resource management, and challenges," *IEEE Commun. Mag.*, vol. 55, no. 8, pp. 138–145, Aug. 2017, doi: 10.1109/MCOM.2017.1600940.

[3] K. Katsalis, N. Nikaein, E. Schiller, A. Ksentini, and T. Braun, "Network slices toward 5G communications: Slicing the LTE network," *IEEE Commun. Mag.*, vol. 55, no. 8, pp. 146–154, Aug. 2017, doi: 10.1109/MCOM.2017.1600936.

[4] X. Foukas, G. Patounas, A. Elmokashfi, and M. K. Marina, "Network slicing in 5G: Survey and challenges," *IEEE Commun. Mag.*, vol. 55, no. 5, pp. 94–100, May 2017, doi: 10.1109/MCOM.2017.1600951.

[5] X. Li *et al.*, "Network slicing for 5G: Challenges and opportunities," *IEEE Internet Comput.*, vol. 21, no. 5, pp. 20–27, Sep. 2017, doi: 10.1109/MIC.2017.3481355.

[6] R. Mijumbi *et al.*, "Network function virtualization: State-of-the-art and research challenges," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 236–262, 1st Quart., 2016, doi: 10.1109/COMST.2015.2477041.

[7] J. G. Herrera and J. F. Botero, "Resource allocation in NFV: A comprehensive survey," *IEEE Trans. Netw. Service Manage.*, vol. 13, no. 3, pp. 518–532, Sep. 2016, doi: 10.1109/TNSM.2016.2598420.

[8] O. Rottenstreich, I. Keslassy, Y. Revah, and A. Kadosh, "Minimizing delay in network function virtualization with shared pipelines," *IEEE Trans. Parallel Distrib. Syst.*, vol. 28, no. 1, pp. 156–169, Jan. 2017, doi: 10.1109/TPDS.2016.2556670.

[9] L. Qu, C. Assi, and K. Shaban, ''Network function virtualization scheduling with transmission delay optimization,'' in *Proc. IEEE/IFIP Netw. Oper. Manage. Symp. (NOMS)*, Istanbul, Turkey, Apr. 2016, pp. 638–644, doi: 10.1109/NOMS.2016.7502870.

[10] H. A. Alameddine, S. Sebbah, and C. Assi, ''On the interplay between network function mapping and scheduling in VNF-based networks: A column generation approach,'' *IEEE Trans. Netw. Service Manag.*, vol. 14, no. 4, pp. 860–874, Dec. 2017, doi: 10.1109/TNSM.2017.2757266.

[11] D. B. Oljira, K.-J. Grinnemo, J. Taheri, and A. Brunstrom, ''A model for QoS-aware VNF placement and provisioning,'' in *Proc. IEEE Conf. Netw. Function Virtualization Softw. Defined Netw. (NFV-SDN)*, Berlin, Germany, Nov. 2017, pp. 1–7, doi: 10.1109/NFV-SDN.2017.8169829.

[12] N. Zhang, Y.-F. Liu, H. Farmanbar, T.-H. Chang, M. Hong, and Z.-Q. Luo, ''Network slicing for service-oriented networks under resource constraints,'' *IEEE J. Sel. Areas Commun.*, vol. 35, no. 11, pp. 2512–2521, Nov. 2017, doi: 10.1109/JSAC.2017.2760147.

[13] L. Wang, Z. Lu, X. Wen, R. Knopp, and R. Gupta, ''Joint optimization of service function chaining and resource allocation in network function virtualization,'' *IEEE Access*, vol. 4, pp. 8084–8094, 2016, doi: 10.1109/ACCESS.2016.2629278.

[14] T.-H. Lei, Y.-T. Hsu, I.-C. Wang, and C. H.-P. Wen, ''Deploying QoS-assured service function chains with stochastic prediction models on VNF latency,'' in *Proc. IEEE Conf. Netw. Function Virtualization Softw. Defined Netw. (NFV-SDN)*, Berlin, Germany, Nov. 2017, pp. 1–6, doi: 10.1109/NFV-SDN.2017.8169837.

[15] H. Huang, S. Guo, J. Wu, and J. Li, ''Service chaining for hybrid network function,'' *IEEE Trans. Cloud Comput.*, to be published, doi: 10.1109/TCC.2017.2721401.

[16] A. A. Haghighi, S. S. Heydari, and S. ShahbazPanahi, ''MDP modeling of resource provisioning in virtualized content-delivery networks,'' in *Proc. IEEE 25th Int. Conf. Netw. Protocols (ICNP)*, Toronto, ON, Canada, Oct. 2017, pp. 1–6, doi: 10.1109/ICNP.2017.8117600.

[17] L. Qu, C. Assi, and K. Shaban, ''Delay-aware scheduling and resource optimization with network function virtualization,'' *IEEE Trans. Commun.*, vol. 64, no. 9, pp. 3746–3758, Sep. 2016, doi: 10.1109/TCOMM.2016.2580150.

[18] V. Eramo, M. Ammar, and F. G. Lavacca, ''Migration energy aware reconfigurations of virtual network function instances in NFV architectures,'' *IEEE Access*, vol. 5, pp. 4927–4938, 2017, doi: 10.1109/ACCESS.2017.2685437.

[19] J. Guo, F. Liu, J. C. S. Lui, and H. Jin, ''Fair network bandwidth allocation in IaaS datacenters via a cooperative game approach,'' *IEEE/ACM Trans. Netw.*, vol. 24, no. 2, pp. 873–886, Apr. 2016, doi: 10.1109/TNET.2015.2389270.

[20] D. Bhamare, R. Jain, M. Samaka, G. Vaszkun, and A. Erbad, ''Multi-cloud distribution of virtual functions and dynamic service deployment: Open ADN perspective,'' in *Proc. IEEE Int. Conf. Cloud Eng.*, Tempe, AZ, USA, Mar. 2015, pp. 299–304, doi: 10.1109/IC2E.2015.49.

[21] N. M. M. K. Chowdhury, M. R. Rahman, and R. Boutaba, ''Virtual network embedding with coordinated node and link mapping,'' in *Proc. IEEE INFOCOM*, Rio de Janeiro, Brazil, Apr. 2009, pp. 783–791, doi: 10.1109/INFCOM.2009.5061987.

[22] J. Kang, O. Simeone, and J. Kang, ''On the trade-off between computational load and reliability for network function virtualization,'' *IEEE Commun. Lett.*, vol. 21, no. 8, pp. 1767–1770, Aug. 2017, doi: 10.1109/LCOMM.2017.2698040.

[23] V. Eramo, E. Miucci, M. Ammar, and F. G. Lavacca, ''An approach for service function chain routing and virtual function network instance migration in network function virtualization architectures,'' *IEEE/ACM Trans. Netw.*, vol. 25, no. 4, pp. 2008–2025, Aug. 2017, doi: 10.1109/TNET.2017.2668470.

[24] C. Wang, O. Spatscheck, V. Gopalakrishnan, Y. Xu, and D. Applegate, ''Toward high-performance and scalable network functions virtualization,'' *IEEE Internet Comput.*, vol. 20, no. 6, pp. 10–20, Nov./Dec. 2016, doi: 10.1109/MIC.2016.111.

[25] M. C. Luizelli, L. R. Bays, L. S. Buriol, M. P. Barcellos, and L. P. Gaspary, ''Piecing together the NFV provisioning puzzle: Efficient placement and chaining of virtual network functions,'' in *Proc. IFIP/IEEE Int. Symp. Integr. Netw. Manage. (IM)*, Ottawa, ON, Canada, May 2015, pp. 98–106, doi: 10.1109/INM.2015.7140281.

[26] P.-C. Lin, Y.-D. Lin, C.-Y. Wu, Y.-C. Lai, and Y.-C. Kao, ''Balanced service chaining in software-defined networks with network function virtualization,'' *Computer*, vol. 49, no. 11, pp. 68–76, Nov. 2016, doi: 10.1109/MC.2016.349.

[27] Q. Zheng, K. Zheng, H. Zhang, and V. C. M. Leung, ''Delay-optimal virtualized radio resource scheduling in software-defined vehicular networks via stochastic learning,'' *IEEE Trans. Veh. Technol.*, vol. 65, no. 10, pp. 7857–7867, Oct. 2016, doi: 10.1109/TVT.2016.2538461.

[28] H. Ren *et al.*, ''Low-latency C-RAN: An next-generation wireless approach,'' *IEEE Veh. Technol. Mag.*, vol. 13, no. 2, pp. 48–56, Jun. 2018, doi: 10.1109/MVT.2018.2811244.

[29] Y. Li, H. Xia, J. Shi, and S. Wu, ''Joint optimization of computing and radio resource for cooperative transmission in C-RAN,'' in *Proc. IEEE/CIC Int. Conf. Commun. China (ICCC)*, Qingdao, China, Oct. 2017, pp. 1–6, doi: 10.1109/ICCChina.2017.8330508.

[30] N. Hayashibara, X. Defago, R. Yared, and T. Katayama, ''The φ accrual failure detector,'' in *Proc. 23rd IEEE Int. Symp. Rel. Distrib. Syst.*, Oct. 2004, pp. 66–78, doi: 10.1109/RELDIS.2004.1353004.

[31] S. Costanzo, I. Fajjari, N. Aitsaadi, and R. Langar, ''DEMO: SDN-based network slicing in C-RAN,'' in *Proc. 15th IEEE Annu. Consum. Commun. Netw. Conf. (CCNC)*, Las Vegas, NV, USA, Jan. 2018, pp. 1–2, doi: 10.1109/CCNC.2018.8319321.

[32] V. Eramo, E. Miucci, and M. Ammar, ''Study of migration policies in energy-aware virtual router networks,'' *IEEE Commun. Lett.*, vol. 18, no. 11, pp. 1919–1922, Nov. 2014, doi: 10.1109/LCOMM.2014.2360190.

[33] V. Eramo, E. Miucci, and M. Ammar, ''Study of reconfiguration cost and energy aware VNE policies in cycle-stationary traffic scenarios,'' *IEEE J. Sel. Areas Commun.*, vol. 34, no. 5, pp. 1281–1297, Apr. 2016, doi: 10.1109/JSAC.2016.2520179.

[34] K. Guo, M. Sheng, J. Tang, T. Q. S. Quek, and Z. Qiu, ''On the interplay between communication and computation in green C-RAN with limited fronthaul and computation capacity,'' *IEEE Trans. Commun.*, vol. 66, no. 7, pp. 3201–3216, Jul. 2018, doi: 10.1109/TCOMM.2018.2811479.

[35] Y. Wang, M. Haenggi, and Z. Tan, ''The meta distribution of the sir for cellular networks with power control,'' *IEEE Trans. Commun.*, vol. 66, no. 4, pp. 1745–1757, Apr. 2018, doi: 10.1109/TCOMM.2017.2780218.

[36] M. S. Rahman, M. Y. S. Uddin, T. Hasan, M. S. Rahman, and M. Kaykobad, ''Using adaptive heartbeat rate on long-lived TCP connections,'' *IEEE/ACM Trans. Netw.*, vol. 26, no. 1, pp. 203–216, Feb. 2018, doi: 10.1109/TNET.2017.2774275.

YOUCHAO YANG received the B.Sc. degree from Sichuan University, Chengdu, China, in 2016.

He is currently pursuing the M.S. degree in communication and information systems with the Chongqing University of Posts and Telecommunications. His current research interests include resource allocation and 5G network slicing.

QIANBIN CHEN (M'03–SM'14) received the Ph.D. degree in communication and information system from the University of Electronic Science and Technology of China, Chengdu, China, in 2002.

He is currently a Professor with the School of Communication and Information Engineering, Chongqing University of Posts and Telecommunications, and the Director of the Chongqing Key Laboratory of Mobile Communication Technology. He has authored or co-authored over 100 papers in journals and peer-reviewed conference proceedings and has coauthored seven books. He holds 47 granted national patents.

GUOFAN ZHAO received the B.Sc. degree from the Chongqing University of Posts and Telecommunications, Chongqing, China, in 2016.

She is currently pursuing the M.S. degree in communication and information systems with the Chongqing University of Posts and Telecommunications. Her current research interests include radio resource allocation, network reliability, and 5G network slicing.

PEIPEI ZHAO received the B.Sc. degree from the Chongqing University of Posts and Telecommunications, Chongqing, China, in 2016.

She is currently pursuing the M.S. degree in communication and information systems with the Chongqing University of Posts and Telecommunications. Her current research interests include service function chain, network virtualization, and 5G network slicing.

LUN TANG received the Ph.D. degree in communication and information system from Chongqing University, Chongqing, China.

He is currently a Professor with the School of Communication and Information Engineering, Chongqing University of Posts and Telecommunications. His current research interests include 5G cellular networks, interference management, and small-cell networks.