IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 8, 2020, accepted July 16, 2020, date of publication July 20, 2020, date of current version July 31, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3010623

Dynamic Network Slice Scaling Assisted by Prediction in 5G Network

JINHE ZHOU, WENJUN ZHAO, AND SHUO CHEN, (Member, IEEE)
School of Information and Communication Engineering, Beijing Information Science and Technology University, Beijing 100101, China

Corresponding author: Jinhe Zhou (zhoujinhe@bistu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61901043 and Grant 61872044, in part by the research project 2020KYNH104 of Beijing Information Science and Technology University.

ABSTRACT Network slicing is a key technology in fifth-generation (5G) mobile networks. Slicing divides a physical network into multiple dedicated logical networks to meet the requirements of diverse use cases. Efficient slice deployment algorithms are critical in reducing network operators' costs and energy consumption and in providing users better service. Many researchers have focused on static deployment when investigating network slices, effectively ignoring network operators' requirements for the dynamic deployment and expansion of such slices. In this paper, we first construct a joint optimization problem of cost and energy consumption. Then, we propose a prediction-assisted adaptive network slice expansion algorithm to deploy network slices dynamically. The proposed algorithm consists of three parts. First, we devise a Holt-Winters (HW) prediction algorithm to determine traffic demand for network slices. This method is intended to avoid frequent changes in network topology. Second, we propose a virtual network function (VNF) adaptive scaling strategy to reasonably determine the number of VNFs and resources required for network slices to avoid resource wastage. Finally, we develop a proactive online algorithm to deploy network slices. This method deploys network slices reasonably via the VNF deployment algorithm and link-routing algorithm to ensure slices' service requirements. Resource capacity and delay requirements are also considered in our evaluation to ensure that network costs and energy consumption are minimized. We then perform a series of simulation experiments to compare the proposed method's performance to state-of-the-art dynamic network slicing technologies. Ultimately, our solution is deemed a suitable candidate for dynamic deployment of 5G network slices; the solution demonstrates advantages of high resource utilization, low deployment costs, and low energy consumption.

INDEX TERMS Adaptive scaling, dynamic deployment, energy consumption, link routing, network slice, virtual network function.

I. INTRODUCTION

Scholars have recognized 5G networks as crucial in enabling network operators to enter the vertical industry market. Dense and crowded environments, such as stadiums and subway stations, exemplify the disadvantages of excessive demand and traffic from the user side [1]. Network slicing can effectively address this problem. Specifically, network slicing technology can divide a physical network into multiple virtual network slices according to network slice providers' needs, thereby improving network resource utilization, reducing network operators' costs and energy consumption, and enhancing network users' experience quality [2]–[4]. Network function virtualization (NFV) and software-defined networking (SDN) are key technologies for constructing 5G network slices [5]. Network slicing technology divides the network into fine-grained network functions (NFs) and deploys them on the virtual platform through containers. Network slices are then further configured and managed through SDN and NFV to provide users an efficient, programmable, and scalable network service. While network slicing technology can ensure greater performance and more efficient networks, it also brings new obstacles: network slice deployment is a major challenge in network slicing technology [6].

The 5G core network is mainly composed of NFs such as the access and mobility management functions (AMF), session management function (SMF), authentication server function (AUSF), user plane functions (UPF), policy control function (PCF), network slice selection function (NSSF),

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

and others [7]. Network operators use NFV technology to build VNFs to meet network slice providers' needs and then deploy these NFs on cloud networks as needed to reduce network costs and energy consumption. Core network slices essentially constitute a virtual network composed of one or more ordered VNF service chains. The deployment of these slices thus represents a virtual network embedding problem [8], wherein virtual network nodes are mapped to physical servers and virtual links are mapped to physical ones.

A key issue in deploying network slices is how to reduce the costs and energy consumption of network operators while ensuring network slice service quality. Current network slicing is plagued by several difficulties [9]. First, most relevant research has focused on static deployment of network slices; in reality, however, the request process of network slices is dynamic, the slice user distribution is uneven, and static resource allocation leads to wasted resources and costs [10]. Second, VNF deployment in network slices affects slices' VNF routing path, thus influencing the delay quality of these slices. Third, the costs associated with network operators and network energy consumption cannot be neglected.

To address these obstacles, this paper investigates the dynamic deployment of network slices, establishes a network slicing management and orchestration architecture, and designs a 5G network-oriented prediction-assisted adaptive network slice expansion algorithm to dynamically create and deploy network slices. In our algorithm, we first consider a dynamic system in which a network slice request is dynamically generated. We then propose a HW prediction algorithm to predict the traffic rate of network slices. This algorithm can minimize prediction errors and simultaneously reduce the cost and delay of creating VNFs in network slices. Next, we construct a VNF adaptive scaling strategy to determine the number of VNFs and resources; this strategy can conserve network resources by (a) setting VNFs reasonably and (b) reducing the number of VNFs. We then discuss network operators' costs and energy consumption and construct a joint optimization problem. Finally, we propose a proactive online network slice deployment algorithm that can dynamically deploy network slices while meeting delay requirements to minimize network costs and energy consumption.

The rest of this paper is organized as follows. In Section II, we provide an overview of related work. Our research objective and main contributions are described in Section III. We introduce our system model and recommend addressing resource allocation via network architecture in Section IV. In Section V, we build a joint optimization problem to fully leverage physical servers while alleviating network operators' deployment costs and energy consumption. An adaptive network slice expansion algorithm is designed in Section VI, including the HW traffic prediction algorithm and proactive online algorithm. Our VNF deployment algorithm (VDA) and link-routing algorithm (LRA) are detailed in Section VII and evaluated via simulations in Section VIII.

II. RELATED WORK

In recent years, 5G network slicing has received extensive attention. Researchers are especially eager to examine network slice deployment. Most early studies focused on static deployment of network slices [11]–[13]. R. Wen *et al*. [14] presented the problem of re-mapping slice recovery with deterministic requirements and proposed two robust network slicing algorithms for slice recovery and reconfiguration under random requirements. The proposed algorithm allowed for adaptable traffic uncertainty tolerance. Balasubramanian *et al*. [15] put forth a unified service architecture that could switch seamlessly between 4G and 5G services via a network slicing paradigm; their evaluation considered signaling costs, service interruptions, and other resource reservation requirements. The proposed solution could improve bandwidth utilization and ensure profitability for mobile edge operators. Agarwal *et al*. [16] developed a queue-based model and adopted a network coordinator to best satisfy the needs of 5G network slices with physical resources. The authors also adopted a static method for network slice deployment while ignoring the network operator's need for dynamic slice deployment and expansion [17], which wasted physical resources to a certain extent. To rectify these problems, Sciancalepore *et al*. [18] devised a mobile slicing strategy for network slices that could adjust and allocate network slice resources based on prediction results to meet the needs of network slices and optimize network resource utilization. However, the authors did not consider the issue of network slice service performance degradation due to inaccurate prediction algorithms.

A few scholars have recently turned to VNF scaling in network slicing [19]–[22]. L. Ruiz *et al*. [23] analyzed VNF placement in 5G networks and estimated the maximum number of VNFs within a given period by using traffic prediction algorithms to estimate the maximum value of network slice traffic. Further, the authors proposed a VNF placement strategy to dynamically expand VNFs of network slices, thereby reducing the service blocking rate and the number of resources in operation. However, the authors only considered VNF placement in this work; they did not attend to the problem of service link routing.

Network-related costs and energy consumption are paramount when deploying network slices; as such, it is important to develop a feasible approach to network slice deployment that can reduce slice-related costs and energy consumption [24]. To address this problem, Yala *et al*. [25] addressed the trade-off between deployment costs and service availability by providing on-demand content delivery services. They further proposed a polynomial–time heuristic method to facilitate proper allocation of computing resources to VNF instances and VNF placement; however, they did not account for the delay problem in network slices. Accordingly, Balasubramanian *et al*. [26] developed an incentive-based model in which edge-based roadside units were adopted to form heterogeneous node resources and

generate available resources to fulfill user requests under minimal delay. Ridhawi *et al*. [27] envisioned a mobile edge computing solution that adjusted the proposed framework through service reliability and security to achieve effective smart city sustainability. This solution thus offered efficient service delivery under a reduced composition delay and enhanced service hit rate. Wang *et al*. [28] fully considered network costs and service performance by constructing a cost model and proposing a coordination method to optimize network slice deployment. Even so, they did not examine slices' energy consumption.

Compared with the extant literature, in this paper, we thoroughly consider dynamic network slice deployment and propose a prediction-assisted dynamic network slice deployment algorithm for 5G networks. Our method surmounts obstacles in network slice deployment and routing. This algorithm can also meet network slice performance requirements (e.g., delay) and reduce the network operators' costs and energy consumption.

III. RESEARCH OBJECTIVE AND CONTRIBUTIONS

Our research objective is to deploy network slices with the lowest cost and energy while ensuring network slice performance. To achieve this aim, our work makes the following contributions:

- We construct a joint optimization problem of network slice cost and energy consumption. To solve this problem, we propose a proactive online algorithm that dynamically deploys network slices under the premise of meeting delay requirements.
- We propose a method to predict the traffic rate of the network slice service chain and reserve redundant resources appropriately using the 3-σ principle. Our corresponding aims are to (a) avoid insufficient network slice configuration caused by the prediction algorithm underestimating the network slice traffic rate and (b) improve network slice service performance.
- We propose a VNF adaptive scaling strategy, which reasonably determines the number of VNFs and resources in network slices based on the traffic rate obtained by the prediction algorithm. This strategy can dynamically enlarge or shrink VNF instances while reducing VNF creation costs and deployment delays.
- Finally, we compare the performance of our approach with state-of-the-art network slice dynamic deployment algorithms. Results show that our algorithm can reduce network operators' costs by 15% and energy consumption by 13%. Resource utilization was also found to increase by more than 6%. As such, the proposed solution is a candidate for dynamic deployment of 5G network slices.

IV. NETWORK ARCHITECTURE AND SYSTEM MODEL

This section first describes our network slicing architecture and system model. The VNF adaptive strategy, cost function, and energy consumption function are also presented in detail.

A. NETWORK ARCHITECTURE

The network slice management and orchestration (MO) architecture is illustrated in Fig. 1. This architecture supports dynamic deployment of 5G network slices on top of the ETSI NFV architecture [29]. The architecture includes five main modules: the communication service management function (CSMF), network slice descriptor (NSD), network slice management function (NSMF), network slice subnet management function (NSSMF), and cloud network.

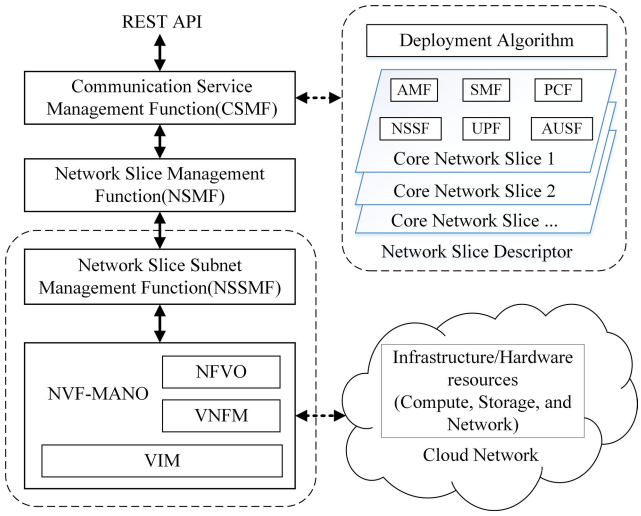

FIGURE 1. The network slice MO architecture.

Network slices are deployed as follows. In the first step, the network operator uses the northbound interface (e.g., REST API) to transmit a network service request to the CSMF module. In the second step, the CSMF transforms the network service request into a network slice request, determines the deployment positions of network slices using the proactive online algorithm in the NSD module, and transmits it to NSMF. In the third step, NSMF transforms the need for network slices into the need for separate subnets and then coordinates and manages network slices in separate subnets via NSSMF. The network-based resource allocation depends on NSSMF with NFV management and orchestration (NFV-MANO) [30]. NFV-MANO is responsible for running the algorithm proposed in this paper to determine the number of VNFs in network slices along with optimal deployment locations. NFV-MANO is mainly composed of NFV orchestration (NFVO) along with a VNF manager and virtualized infrastructure manager. Finally, network operators allocate resources from the cloud network to deploy network slices.

B. SYSTEM MODEL

The system model includes network operators, network slice providers, and network slice end users. Among these, network operators mainly provide infrastructure, which includes computing resources, storage resources, and communication resources. Network slice providers construct slices by renting network operators' resources and providing required services

to network slice end users. A network slice provider can create a network slice instance across multiple operators. In this paper, we only consider the case of a single operator; here, a network operator can simultaneously serve as a network slice provider. The undirected graph $G(N, L, R_n, B_l)$ denotes the physical resource provided by the network operator. In this case, N is the set of physical servers, L is the link resource in the network, R_n is the resource capacity of physical server n, and B_l is the bandwidth capacity of link l.

A network slice is a dedicated virtual network composed of one or more service chains; different service chains consist of single or multiple VNF instances. VNFs are constructed by consuming a certain number of physical server resources. In this paper, the physical server resource is abstracted as a type of virtual resource. VNF instances of the same type can be deployed on multiple physical servers, and multiple types of VNFs can be deployed on one physical server. We assume there are M network slices in the network. For specific network slice m, it is necessary to deploy and operate F kinds of VNFs for network slice users in multiple time slots, and the time of each time slot is roughly several minutes or several seconds. In total, there are I service chain requirements arriving at time slot $t = [1, 2, \cdot x, T]$. At time slot $t_i \le t \le t_i + \Delta t_i$, the traffic rate of service chain i for network slice m is $r_i^m(t)$. The demand for traffic rates continues to change over time. The demand for the traffic rate of a given network slice service chain is predicted by the HW algorithm, which will be introduced in Section V.

C. VNF ADAPTIVE STRATEGY

The traffic demand of the service chain for network slices also changes with time, as does the number of required VNFs. We assume that at initial time $t = 0$, the VNFs previously required for network slices have been placed in the physical server node. The initial state information of network slices at this moment is known. In service chain i of network slice m, the number of resources required by the type-f VNF is $R_i^{m,f}(0), f \in \{1, 2, \ldots, F\}$. To efficiently use server resources, we propose an adaptive scaling strategy to update VNF instances and determine the number of VNFs in network slices as well as corresponding resources according to the service chain's traffic demands. Therefore, newly created VNF instances are adjusted to an appropriate size to meet the needs of network slice users. Once the VNF instances are constructed, network slices cannot immediately change the sizes of VNFs but are determined in advance. We can therefore avoid a system restart that results in service interruption.

The traffic rate of service chain i of network slices m at time t is $r_i^m(t)$. The resource number $R_i^{m,f}(t)$ required by the type-f VNF in service chain i of network slice m is defined as the number of data packets processed per unit time. The calculation formula is as follows:

$$R_i^{m,f}(t) = \sum_{i \in I : t \in (t_i, t_i + \Delta t_i)} \varphi_i^f r_i^m(t) \sigma(f) / L_p, \tag{1}$$

where $\sigma(f)$ is the processing time of the packet of the type-f VNF; L_p is the length of the packet in the stream; and $\varphi_i^f = 1$ indicates that the type-f VNF is used in service chain i (otherwise, the type-f VNF is not used in service chain i). In time slot t, the number of newly added type-f VNFs in network slice m is

$$\kappa^{m,f}(t) = \begin{cases} 0, & R_i^{m,f}(t) \le R_i^{m,f}(t-1), \\ \left\lceil \dfrac{R_i^{m,f}(t) - R_i^{m,f}(t-1)}{R_{\max}^{m,f}} \right\rceil, & \text{otherwise}, \end{cases} \tag{2}$$

where $R_{\max}^{m,f}$ represents the maximum number of resources required by a single type-f VNF instance in network slice m, defined as the maximum service traffic that the VNF instances can handle. Specifically, the number of newly added VNFs in network slices and the corresponding number of required resources include two cases: (1) when the number of newly added resources required at time t is less than or equal to $R_{\max}^{m,f}$ (i.e., at $0 \le R_i^{m,f}(t) - R_i^{m,f}(t-1) \le R_{\max}^{m,f}$, the network slice must add one VNF), the VNF capacity is $C(t) = R_i^{m,f}(t) - R_i^{m,f}(t-1)$, and the capacity is expressed as $C_o^{m,f}(t)$; (2) when the number of newly added resources required at time t is greater than $R_{\max}^{m,f}$ (that is, at $R_{\max}^{m,f} \le R_i^{m,f}(t) - R_i^{m,f}(t-1)$, the network slice must add $\kappa^{m,f}(t) - 1$ capacity $R_{\max}^{m,f}$ and one capacity $R_i^{m,f}(t) - R_i^{m,f}(t-1) - (\kappa^{m,f}(t) - 1)R_{\max}^{m,f}$ type-f VNF), the capacity is expressed as $C_p^{m,f}(t)$. In summary, the number of type-f VNFs required in network slice m at time t is

$$\pi^{m,f}(t) = \begin{cases} \pi^{m,f}(0), & t = 0, \\ \pi^{m,f}(t-1) + \kappa^{m,f}(t), & 1 \le t \le T. \end{cases} \tag{3}$$

In terms of data flow routing in the service chain of network slices, once VNFs are created, we can divide data flow requirements into multiple VNF instances and aggregate data flow at the node's next-hop VNF. We define the variable $h_i^m(l)$ as the proportion of data flow demand i along link l. Notably, the bandwidth used by the link along the path should be less than the link's remaining total bandwidth.

D. COST FUNCTION

Network operators deploying network slices will incur costs and delays, mostly attributed to VNF deployment and data stream transmission. Because the cost and delay required to delete a VNF instance are small, they can be ignored. Therefore, we regard the creation of VNF instances and the transmission of service chain data streams as the costs of a network operator. The cost function of the network operator includes the costs of deploying network slices and of service execution.

Suppose $\delta_n^{m,f}(t)$ represents the number of type-f VNFs deployed on physical server n by network slice m at time t ($\delta_n^{m,f} \ge 1$ indicates that multiple type-f VNFs are deployed on physical server n). $U_{dep}^m(t)$ represents the deployment costs

of VNFs for network slice m at time t, expressed as

$$U_{dep}^{m}(t) = \sum_{n\in N}\sum_{f\in F} \tau_{n}^{m,f}\delta_{n}^{m,f}(t), \tag{4}$$

where $\tau_{n}^{m,f}$ represents the cost of network slice m deploying a single type-f VNF on physical server n. Assuming that $d_{n}^{m,f}$ represents the delay required for network slice m to deploy a single type-f VNF on physical server n, the total delay of the VNF deployment of network slice m at time t is expressed as $D_{dep}^{m}(t)$, and the specific expression is

$$D_{dep}^{m}(t) = \sum_{n\in N}\sum_{f\in F} d_{n}^{m,f}\delta_{n}^{m,f}(t). \tag{5}$$

The transmission of data streams in network slices also requires time and cost. Due to the low cost and low delay of data transmission between VNFs on the same node, they can be ignored; we only consider the cost and delay of VNF data stream transmission between nodes. $r_{f,f',n,n'}^{m,i}(t)$ represents the inflow rate of service chain i of network slice m at time t from VNF f in physical node n to VNF f' in physical node n'. Suppose $\chi_{n,n'}^{m,i}$ is the cost of transmitting a unit data stream from physical node n to the physical node n' for service chain i of network slice m. The data transmission cost of network slice m at time slot t is written as $U_{tra}^{m}(t)$, and its expression is

$$U_{tra}^{m}(t) = \sum_{i\in I}\sum_{n\in N}\sum_{n'\in N}\sum_{f\in F}\sum_{f'\in F} \chi_{n,n'}^{m,i} r_{f,f',n,n'}^{m,i}(t). \tag{6}$$

Similarly, only the transmission delay between physical nodes is considered in a network slice's data stream transmission. $d_{n,n'}^{m,i}$ denotes service chain i of network slice m transmitting a unit data stream transmission delay from physical node n to physical node n'. The data transmission delay of network slice m at time t is $D_{tra}^{m}(t)$, expressed as

$$D_{tra}^{m}(t) = \sum_{i\in I}\sum_{n\in N}\sum_{n'\in N}\sum_{f\in F}\sum_{f'\in F} d_{n,n'}^{m,i} r_{f,f',n,n'}^{m,i}(t). \tag{7}$$

In summary, the total cost of network operators includes the costs of deploying VNFs and executing services, which can be expressed as

$$U(t) = \sum_{m\in M}\left[U_{\text{dep}}^{m}(t) + U_{tra}^{m}(t)\right]. \tag{8}$$

The total delay of network slice m at time t includes the total deployment delay of VNFs and the data transmission delay, which can be expressed as

$$D^{m}(t) = D_{dep}^{m}(t) + D_{tra}^{m}(t). \tag{9}$$

E. ENERGY CONSUMPTION FUNCTION

The problem of energy consumption in the network cannot be ignored; it is necessary to increase the resource utilization rate as much as possible to reasonably reduce network energy consumption. The energy consumption of the network server is related to the server's resource usage. Therefore, the energy consumption of the physical server in the network at time t is defined as

$$E(t) = \sum_{n\in N}\left[E_{idel}^{n} + \left(E_{\max}^{n} - E_{idel}^{n}\right)\vartheta^{n}(t) + E_{open}^{n}\right], \tag{10}$$

where E_{idel}^{n} is the energy consumption of physical server n when idle; $E_{\max}^{n}$ is the maximum energy consumption of physical server n (i.e., the resources of physical server n are fully used); E_{open}^{n} is the energy consumption required to open the nth physical server; and $\vartheta^{n}(t)$ is the resource utilization rate of physical server n at time t ($0 \le \vartheta^{n}(t) \le 1$).

The resource utilization rate of physical server n at time t can be expressed as

$$\vartheta^{n}(t) = \frac{\sum_{m\in M}\sum_{f\in F}\left\{C_{o}^{m,f}(t), C_{p}^{m,f}(t) + R_{\max}^{m,f}\right\}\delta_{n}^{m,f}}{R_{n}}, \tag{11}$$

where R_n is the maximum resource capacity of physical server n. The numerator is expressed as the consumed resource capacity for physical server n at time t. If the newly added VNF instances of network slice m at time t are equal to 1, then $\left\{C_{o}^{m,f}(t), C_{p}^{m,f}(t) + R_{\max}^{m,f}\right\}\delta_{n}^{m,f} = C_{o}^{m,f}(t)\delta_{n}^{m,f}$; otherwise, $\left\{C_{o}^{m,f}(t), C_{p}^{m,f}(t) + R_{\max}^{m,f}\right\}\delta_{n}^{m,f} = C_{p}^{m,f}(t) + \delta_{n}^{m,f}R_{\max}^{m,f}$. Because there are two sizes of VNFs of $C_{p}^{m,f}(t)$ and $R_{\max}^{m,f}$ on the physical server, the number of $R_{\max}^{m,f}$ instances is determined by $\delta_{n}^{m,f}$.

V. JOINT OPTIMIZATION PROBLEM OF COST AND ENERGY CONSUMPTION

In this paper, a prediction algorithm is first proposed to determine the traffic rate demand $r_{i}^{m}(t)$ of service chain i in network slice m; the number of VNFs and resources is respectively determined according to the prediction results. Further, we develop an optimal VNF placement and link allocation using a proactive online algorithm to minimize network energy consumption and total cost under the premise of meeting the network delay. Therefore, we construct a joint optimization problem of network cost and energy consumption as shown in Formula (12).

$$\min \sum_{t\in T}[\varphi E(t) + (1-\varphi)U(t)], \tag{12}$$

$$D_{dep}^{m}(t) + D_{tra}^{m}(t) \le D_{thr}^{m}(t), \tag{12.1}$$

$$\sum_{m\in M}\sum_{f\in F}\left\{C_{o}^{m,f}(t), C_{p}^{m,f}(t) + R_{\max}^{m,f}\right\}\delta_{n}^{m,f} \le R_{n}(t), \tag{12.2}$$

$$\sum_{i\in I, t\in t_i+\Delta_i} r_{i}^{m}(t)h_{i}^{m}(l) \le B_{l}(t), \tag{12.3}$$

$$\sum_{n\in N}\delta_{n}^{m,f} = \pi^{m,f}(t), \tag{12.4}$$

$$\delta_{n}^{m,f} \in \{0,1\}, \varphi \in [0,1], r_{i}^{m}(t), h_{i}^{m}(l) \ge 0 \quad \forall t \in T, m \in M, f \in F, n \in N, i \in I, \tag{12.5}$$

Constraint (12.1) meets the delay requirements of network slices, where $D_{thr}^{m}(t)$ is the delay threshold of network slice

m at time t. Constraint (12.2) ensures that the number of resources used to deploy VNFs on physical server n is less than this server's available resources. Constraint (12.3) is satisfied such that the total bandwidth carried by link l is less than the remaining capacity of link B_l. Constraint (12.4) ensures that all VNFs of a network slice can be deployed on physical servers.

Collectively, the problem of the joint optimization of network slice costs and energy consumption is NP-hard. The joint optimization problem can be transformed into a special case of the delay-constrained shortest path problem. It is assumed that the deployment costs and delay of network slices and energy consumption are not considered; that is, $D^m_{dep}(t) = 0, U^m_{tra}(t) = 0, E(t) = 0$. The system cost is transformed into the cost of data transmission. According to Constraint (12.1), this problem is akin to identifying the path with the lowest cost under the delay constraint. The shortest path problem with a delay limitation is also NP-hard [31]. Taken together, the joint optimization problem of deployment costs and energy consumption of network slices derived from the above is NP-hard.

We put forth a proactive online algorithm to solve this problem. The solution to the deployment problem of network slices involves three steps. First, we propose an HW algorithm to predict whether new demand exists for network slices. If there is new demand, then we can calculate the number of newly added VNF instances in the network slice and use the VNF deployment algorithm to map these added VNFs. Second, we map the links of network slices through routing algorithms. The deployment of VNF instances determines the routing path to some extent, which influences routing and transmission costs in the network slice service chain. It is therefore crucial to manage VNF deployment properly. To fully use physical server resources in the network while reducing network slices' deployment costs and energy consumption, we construct the objective function as an optimization problem of deployment costs and network energy consumption as shown in Formula (13).

$$\begin{aligned}\min & \sum_{t\in T}\left[\varphi E(t)+(1-\varphi)\sum_{m\in M}U^m_{dep}(t)\right]\\ &=\sum_{t\in T}\left[\varphi\sum_{n\in N}\left[\frac{\sum_{m\in M}\sum_{f\in F}\left\{C^{m,f}_o(t),C^{m,f}_p(t)+R^{m,f}_{\max}\right\}\delta^{m,f}_n}{R_n}\right.\right.\\ &\left.\left.\left(E^n_{\max}-E^n_{idel}\right)+E^n_{idel}+E^n_{open}\right]+(1-\varphi)\right.\\ &\left.\sum_{m\in M}\sum_{n\in N}\sum_{f\in F}\tau^{m,f}_n\delta^{m,f}_n(t)\right].\\ &\text{subject to Constraints}(12.1-12.5).\end{aligned}\tag{13}$$

VI. PROACTIVE ONLINE ALGORITHM DESIGN

A network slice can be deployed in two ways. In one case, reactive deployment responds passively to what has occurred in the network and cannot actively predict the network status, leading to wasted network resources. Alternatively, proactive deployment can promptly adjust the VNF instance allocation strategy and link-routing strategy based on actual network circumstances. The traffic requirements of each network slice evolve constantly; therefore, we developed a proactive method to predict the network slice traffic rate in advance. In this way, network slices can be deployed in a timely and effective manner to improve their service quality and minimize network resource wastage.

In this section, we present a proactive online algorithm for network slice deployment, called network slice deployment for cost and energy consumption minimization (NSD-CECM). First, the HW prediction algorithm is introduced. Each network slice is configured by predicting the traffic demand of the network slice service chain to avoid frequent updates to network topology. Then, we further elaborate on the NSD-CECM algorithm to efficiently handle fluctuations in the network slice service chain's traffic rate in each time slot t with the help of the HW prediction algorithm. Network operators can then apply this algorithm to deploy network slices dynamically.

A. TRAFFIC DEMAND PREDICTION USING HW ALGORITHM

Users' service requests for network slices change dynamically with time but exhibit definite trends and periodicity. Therefore, an HW prediction algorithm is proposed to predict slices' future traffic rate. The predicted value will be used to set the number of VNFs and resources to facilitate dynamic network slice deployment, thereby enhancing resource utilization and network operator revenue. The accuracy of the prediction algorithm will directly affect the resource allocation algorithm's performance: the higher the accuracy, the higher the corresponding resource utilization and the lower the costs of network slices. This section presents a detailed analysis of the traffic prediction HW algorithm.

Traffic prediction is primarily responsible for predicting the traffic rate of a network slice service chain and for configuring network requests according to network slices' varying needs. Assuming that users' requests for network slices are evenly distributed throughout the network, the traffic rate requirements of network slice m are implemented as a point process Γ_m:

$$\Gamma_m=\sum_{t=0}^{W}\xi_t r^m_i(t),\tag{14}$$

where ξ_t is the Dirac parameter of sample t. A user's demand for the traffic rate of service chain i of network slice m is expressed as $r^m_i(t)$. We first provide the requested data for service chain i of network slice m in the previous cycle T_c, which is represented as a vector: $r^m_i = \left[r^m_i(t-T_c), r^m_i(t-T_c+1), \ldots, r^m_i(t)\right]$. We can then predict the demand value of the traffic rate of service chain i of network slice m in a time window T_{WIN} using the HW

prediction algorithm, which is also expressed as a vector: $r_i^m = [r_i^m(t+1), r_i^m(t+2), \ldots, r_i^m(t+T_{WIN})]$. It is assumed that the process Γ_m of the user's service request is smooth and traversal within one cycle. The HW prediction algorithm can forecast the future traffic demand $r_i^m(t)$ of slice m. Because the periodic effect (added through the level and trend) can influence prediction values, the cumulative form of the HW algorithm is chosen. The level $l_{i,t}^m$, trend $s_{i,t}^m$, and period $p_{i,t}^m$ are used to predict future traffic demand.

However, this prediction algorithm may underestimate network slices' actual traffic rate requirements. An underestimated traffic rate will directly cause network slices to fail to handle all user requests, then compromising network slices' service availability. To avoid this scenario, we reserve some redundant resources based on the estimated value of the HW algorithm to ensure that actual demand for network slices exceeds the predicted traffic rate. Network slices can configure sufficient resources to meet user requests. We choose the 3-σ principle to guarantee that network slices will be configured with sufficient resources, where σ is the standard deviation of the estimated value of the training set and the actual demand value in the prediction algorithm. Based on the 3-σ principle and the HW algorithm, the traffic demand for service chain i of network slice m at time $t+h$ can be predicted using the following formula:

$$\begin{aligned} r_{i,t+h}^m &= l_{i,t}^m + h s_{i,t}^m + p_{i,t-k+h}^m + 3\delta, \text{ where }, \\ l_{i,t}^m &= \alpha_1 \left(r_{i,t}^m - p_{t-k}\right) + (1-\alpha_1)\left(l_{i,t-1}^m + s_{i,t-1}^m\right), \\ s_{i,t}^m &= \alpha_2 \left(l_{i,t}^m - l_{i,t-1}^m\right) + (1-\alpha_2)\, s_{i,t-1}^m, \\ p_{i,t}^m &= \alpha_3 \left(r_{i,t}^m - l_{i,t}^m\right) + (1-\alpha_3)\, p_{i,t-k}^m, \end{aligned} \tag{15}$$

where k is the periodic length, $l_{i,t}^m$ is the smoothed value in time slot t, and $r_{i,t}^m$ is the actual data in time slot t. The initial value expressions of $l_{i,t}^m$ and $s_{i,t}^m$ are

$$\begin{aligned} l_{i,0}^m &= x_0, \\ s_{i,0}^m &= \frac{1}{K}\left(\frac{x_{K+1}-x_1}{K} + \frac{x_{K+2}-x_2}{K} + \ldots + \frac{x_{K+K}-x_K}{K}\right). \end{aligned} \tag{16}$$

The HW algorithm has three adjustable parameters α_1, α_2 and α_3: α_1 is the data smoothing factor, α_2 is the trend smoothing factor, and α_3 is the cyclically changing smoothing factor, where $0 < \alpha_1, \alpha_2, \alpha_3 < 1$. When solving for these parameters, we adopt the least squares method to minimize error.

B. PROACTIVE ONLINE ALGORITHM

The network slice deployment algorithm NSD-CECM is shown in Alg. 1. By using the HW prediction algorithm to determine the traffic rate $r_i^m(t)$ of service chain i of network slice m, the predicted value is further used to calculate the number $\pi^{m,f}(t)$ and capacity of newly added type-f VNF instances in service chain i of network slice m. Then, by calling the VDA and LRA, the newly added VNF instances of network slices are deployed.

Algorithm 1 Proactive Online Network Slice Deployment for Cost and Energy Consumption Minimizing— NSD-CECM

Input: $M, N, L, R_n, B_l, T, \varphi_i^f, L, F, R_{max}^{m,f}, \sigma(f)$
Output: $C_p^f(t), C_o^f(t), \pi^{m,f}(t), h_i^m(l), \delta_n^{m,f}(t)$

State initialize $\pi^{m,f}=0, h_i^m(l)=0, \delta_f^n=0, r_i^m(t)=0$;
for $t = 1, 2, \ldots, T$ **do**
  for $m = 1, 2, \ldots, M$ **do**
    for $f = 1, 2, \ldots, F$ **do**
      Compute the total resources capacity $R_i^{m,f}(t)$ requested by VNF f in slice m and acc.to (1);
      if $R_i^m(t) < R_i^m(t-1)$ **then**
        Update the instance of the buffer area,and put the extra VNF in the last time slot into the buffer; release spare instances last for ϱ time slots;
        Update the remaining space capacity $R_n(t)$ and the remaining bandwidth capacity $B_l(t)$;
      else
        Compute $\pi^{m,f}(t)$ requested by VNF f in slice m acc.to (2), and using $R_i^m(t)$ obtain $C_p^f(t), C_o^f(t)$ by acc.to (3);
        Call VDA to deploy new type n VNF instances;
        Update the remaining space capacity $R_n(t)$;
      end if
    end for
  end for
end for
Call LRA to set routing path for sevice chain;
Update the remaining bandwidth capacity $B_l(t)$;
return $C_p^f(t), C_o^f(t), \pi^{m,f}(t), h_i^m(l), \delta_n^{m,f}(t)$.

The life cycle of VNFs in network slices includes a preparation phase, instantiation, configuration and activation phase, run-time phase, and decommissioning phase. Each stage involves network costs and energy consumption. The cost of maintaining redundant VNFs is much smaller than that associated with newly added VNFs [32], [33]. Therefore, we use redundant VNFs to reduce unnecessary stages of creating VNFs. This method can avoid frequent creation or release of VNF instances in the event of traffic fluctuations, thus reducing the total network cost.

During the service chain deployment of network slices, the traffic rate requirement of time slot t may be greater than the traffic rate requirement in $t+1$; that is, the VNF instance requirement of time slot t is greater than that of time slot $t+1$. Redundant VNF instances in t time slots will not be deleted directly in time slot t at $t+1$. This circumstance can mitigate frequent addition or deletion of VNF instances under a changing traffic rate. A buffer queue is therefore constructed to mark redundant VNF instances of network slices in the $t+1$ time slot. In short, redundant VNF instances will be in a dormant state at t. T is used to

mark redundant VNFs in $t+1$ and stored in the buffer queue. In the next time slot, if a new VNF instance needs to be added, the VNF instances will first be popped from the buffer queue to build a network slice service chain and hence reduce deployment costs. The life cycle ϱ of unused VNF instances in the buffer queue will last for a time slot and will be deleted later.

As mentioned above, changes in the requirements of VNF instances will determine the VNF configuration. As such, proceeding from time slot $t=2$, the size of the VNF instance requirement of the t slot and that of the $t-1$ slot is compared. Specifically, in our algorithm, the number of resources required for the type-f VNF in service chain i of network slice m is first estimated based on traffic obtained by the HW prediction algorithm. It is further compared with the number of resources required by VNFs in the last time slot to determine whether the traffic rate requirement increases or declines. If the traffic rate requirement is less than that in the previous time slot, then excess VNF instances in the previous time slot are placed in the buffer queue corresponding to different types of VNFs. The VNF instances whose life cycle reaches a time slot will be deleted. Then we calculate the remaining capacity of each physical server and the remaining bandwidth capacity of the link. If the traffic rate requirement is greater than the traffic rate requirement of the previous time slot, the number of newly added VNFs and resources corresponding to those VNFs are determined using the formula. Instances in the buffer queue are next preferentially configured to reduce deployment costs, after which the VDA is called to deploy newly added VNF instances of the network slice service chain. Finally, the LRA is called to perform link routing for these VNF instances. The specific VDA and LRA are introduced in the next section.

VII. VNF DEPLOYMENT ALGORITHM AND LINK-ROUTING ALGORITHM

A. VNF DEPLOYMENT ALGORITHM

VNF instance deployment in network slices uses the discrete particle swarm optimization (DPSO) algorithm, which is based on simulated annealing (SA). The DPSO algorithm is an intelligence optimization algorithm that studies the behavioral characteristics of birds when finding food and uses information sharing between birds to determine the optimal solution. We consider the mapping process of the network operator's physical server and the VNFs of the network slice analogous to birds' search for food. Each particle represents a possible solution to the problem, with the optimal solution representing food the birds ultimately find. The VNF deployment algorithm is presented in Alg. 2.

When deploying network slices, to fully use the network's physical servers and reduce slices' deployment costs and energy consumption, we use the objective function as the fitness function of the VNF deployment algorithm as shown in Formula (13). First, suppose the population consists of N

Algorithm 2 VNF Deployment Algorithm for Cost and Energy Consumption Minimizing

Input: $G, C_p^f(t), C_o^f(t), \pi^{m,f}(t), R_{max}^{m,f}, R_n(t)$
Output: $\delta_n^{m,f}$

```
State initialize δ_n^{m,f} = 0;
for t = 1, 2, ..., T do
    Set the population size N,the maximum number of iterations ϑ_max, π_1 and π_2;
    Initialization position size v;
    Compute the fitness value of all particles acc.to (13), and find the individual optimal value s_la, the global optimal value s_ga;
    Set annealing temperature T = −fit(ng)/log(0.2);
    if 1 < ϑ < ϑ_max and 1 < scope < N then
        Use SA algorithm to search for the local optimal solution of the particle, and to determine whether to replaces s_ga with s_la acc.to (20);
        Update particle velocity vector acc.to (21) and position vector acc.to (19), and update individual optimal solution s_la and global optimal solution s_ga;
    end if
end for
return VNF best mapping position δ_n^{m,f}.
```

particles and the vector $\vec{\delta_n} = (\delta_{n1}, \delta_{n2}, \ldots, \delta_{n\kappa})$ is used to represent the nth particle, where $n \in 1, 2, \ldots, N$. N is the population size, κ is the total number of newly added VNF instances at the current moment, and $\delta_{n\kappa}$ is the current serial number of physical server n to which κ VNFs are mapped. Next, we substitute $\vec{\delta_n}$ for the objective function to find the objective function value of the current mapping scheme. Once the fitness of a particle is determined, the algorithm updates the local optimal solution $s_{l\kappa}$ and the global optimal solution $s_{g\kappa}$ to the current individual. For particles to efficiently find the optimal solution, each particle corresponds to a flight speed, and the flight speed of the nth particle is denoted by vector $\vec{v} = (v_{n1}, v_{n2}, \ldots, v_{n\kappa})$. We use the DPSO algorithm with a compression factor; the speed update appears in Formula (17). The compression factor ρ is solved using Formula (18). The learning factors π_1 and π_2 represent the experience gained by the particle itself and by other particles. The learning factor is adjusted to balance particles to find the optimal solution globally and locally. The location update is shown in Formula (19).

$$v_{na}^{m+1} = \rho\left[v_{na}^{m} + \pi_1 \,\text{random}(0,1)\left(s_{la}^{m} - \varepsilon_{na}^{m}\right) + \pi_2 random(0,1)\left(s_{ga}^{m} - \varepsilon_{na}^{m}\right)\right]. \quad (17)$$

$$\rho = \frac{2}{|2-\pi-\sqrt{\pi^2-4\pi}|}, \pi = \pi_1+\pi_2, \pi > 4. \quad (18)$$

$$\varepsilon_{na}^{m+1} = \varepsilon_{na}^{m} + v_{na}^{m+1}, \quad a \in [1, 2, \ldots N]. \quad (19)$$

The DPSO algorithm has a strong global search ability but can easily fall into the local optimal solution. To prevent this scenario, we choose the DPSO algorithm based

on SA. The SA algorithm regards the energy of system cooling as the objective function and simulates the optimization process as the system cooling process. Further, the SA algorithm is used to search for the individual optimal solution, and a certain probability is used to determine whether to replace the individual optimal solution with the global optimal solution. The state transition probability is illustrated in Formula (20); the speed update is optimized to Formula (21).

$$P = \begin{cases} 1, fit_{na}^{m} < fit_{ga}^{m} \\ -EXP\left(fit_{na}^{m} - fit_{ga}^{m}\right)/T, \quad fit_{na}^{m} \geq fit_{ga}^{m}. \end{cases} \tag{20}$$

$$v_{na}^{m+1} = \rho\left[v_{na}^{m} + \pi_1 \,\text{random}(0,1)\left(s_{la}^{m} - \varepsilon_{na}^{m}\right) + \pi_2 random(0,1)\left(s_{ga}^{\prime m} - \varepsilon_{na}^{m}\right)\right]. \tag{21}$$

B. ONLINE ALGORITHM FOR SERVICE CHAIN ROUTING WITH DELAY CONSTRAINTS

After completing the mapping between network slice VNF instances and the physical server, link routing must be performed on VNF instances. To reduce complex state changes, assuming that the number of previous packets in the stream remains unchanged, we only need to route newly added packets in the stream to newly added VNF instances based on link state information provided by the network. The newly added flow requirement set at t is I, and the deployment order of physical servers is determined by the VNF mapping result. The VNF sequence added for each flow requirement i is $o_i, n_i^1, n_i^2, \ldots, n_i^g, s_i$, and the flow i path sequence of network slice m is

$$\psi_m = \left\{o_i, n_i^1, n_i^2, \ldots, n_i^g, s_i\right\}, \quad \forall n_i^g \in N, i \in I. \tag{22}$$

Service chain i is connected through the path $p_{o_i,n_i^1}^m, p_{n_i^1,n_i^2}^m, \ldots, p_{n_i^g,s_i}^m$. For each path p, h_{ip}^m is the ratio of a new flow to path p. H_{lp}^m indicates that link l of network slice m is mapped onto path p. Then the bandwidth constraint condition is transformed into Formula (23). The delay constraint condition is Formula (24), and Formula (25) ensures that the traffic rate cannot exceed 1.

$$\sum_{i \in I, t \in t_i + \Delta_i} r_i^m(t) H_{lp}^m h_{ip}^m \leq B_l(t), \tag{23}$$

$$D_{dep}^m(t) + D_{tra}^m(t) \leq D_{thr}^m(t), \tag{24}$$

$$r_i^m(t) \leq 1, \; H_{lp}^m \in \{0,1\}, \quad \forall t \in T, \; \forall p \in P, \forall l \in L, \quad \forall i \in I, \; \forall m \in M. \tag{25}$$

The purpose of link-routing is to find the path with the lowest cost under delay conditions. The above problem involves linear programming. Therefore, we choose a simple dual method to solve the routing problem of network slicing. This algorithm has the advantages of a fast operating speed and strong optimization ability. The routing problem is constructed as shown in Formula (26):

$$\min \gamma(l) \sum_{i \in I} \sum_{n \in N} \sum_{n' \in N} \sum_{f \in F} \sum_{f' \in F} \chi_{n,n'}^{m,i} r_{f,f',n,n'}^{m,i}(t). \tag{26}$$

where $\gamma(l)$ is the dual variable. The specific algorithm is presented in Alg. 3.

Algorithm 3 Proactive Online Link Routing Algorithm

Input: $r_i^m(t), R_l(t)$

State initialize $h_i^m = 0$;
Derive $r_{i,new}^m$ based on $r_i^m(t)$;
for $i = 1, 2, \ldots, I$ **do**
  $p^* = argmin_{p \in P_i^m} \gamma(l) U_{tra}^m(t)$;
  if $\sum_{i \in I} \sum_{l \in P} H_{lp}^m h_{ip}^m \leq 1$ **then**
    if $D_{p^*} \leq D^m(t) - D_{dep}^m(t)$ **then**
      Route i through p^*;
      for each $l \in p^*$ **do**
        $\gamma(l) = \gamma(l)[\frac{h_i^m(l) r_{i,new}^m}{B_l(t)} + 1]$;
      end for
    end if
  end if
end for
return Best link routing outcome.

VIII. PERFORMANCE EVALUATION

A. SIMULATION SETTINGS

For simulations, we selected NS2 and GT-ITM tools to generate the physical network and network slice instance requests based on the network slicing standard in 3GPP and analyzed the results in Python. We evaluated the process using a temporary emulator with dual Intel® Xeon CPU 2.40GHz 4 cores and 16GB RAM. The simulated physical network was composed of 12 physical servers with a capacity of 4800Mbps and 20 physical servers with a capacity of 2400Mbps, which we used to place VNF instances. The link bandwidth between physical servers was 1000Mbps. We regarded traffic from the same source node to the target node (falling into time slot t) as a flow. In this case, traffic in time slot t denotes the total traffic size divided by the time slot interval, with an average packet length of 1024 bytes. We considered five types of VNFs that complied with 5G network standards; specific parameters are listed in Table 1 [34]. The network slice randomly selected VNFs to form a service chain. We further compared the energy consumption and cost with VNF provisioning for cost minimization (VPCM) [35], elastic service function chains algorithm (ESFC) [36], and hybrid slice reconfiguration with resource reservation mechanism (HSR-RSV) [37]. Other simulation parameters were as follows: $N = 32$, $\pi_1 = 2$, $\pi_2 = 2$, $\vartheta_{max} = 80$, $M = 10$.

B. EVALUATION OF TRAFFIC FORECAST

We evaluated regression loss in the prediction algorithm based on R-squared. R^2 is a statistical measure of how close a

TABLE 1. VNF parameters.

VNF types	NAT	IDS	EV	LB	FW
Processing time(us)	200	150	100	80	50
Maximum capacity(mbps)	200	260	450	500	600

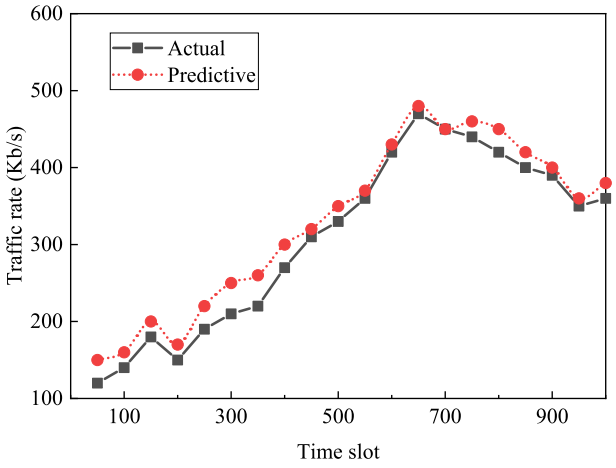

FIGURE 2. Comparison between predicted and actual.

predicted value is to an actual value. We defined the error rate of the prediction algorithm as R^2. The calculation formula R^2 is shown in Formula (27).

$$R^2 = 1 - \frac{\sum_{i\in I}(r_i^{real} - r_i)^2}{\sum_{i\in I}(r_i^{real} - \bar{R})^2}, \tag{27}$$

where $\bar{R}$ is the mean of the data in the training set, r_i is the predicted value of the traffic prediction algorithm, and r_i^{real} is the actual traffic rate of the service chain in the network slice.

Fig. 2 illustrates the relationship between the predicted and actual values of the traditional HW algorithm for network slices. Some errors remained between these values; therefore, we revised the traditional HW algorithm. Under different prediction algorithms, the relationship between the error of the prediction algorithm when the size of the training set changes is shown in Fig. 3. As indicated, within a certain range, the prediction error rate gradually declined as the size of the training set increased. The prediction algorithm performed optimally when the size of the training set was 5. If the training set was too large, the prediction error increased. When the training set was too large, historical traffic information became overly complex, leading to adverse consequences. Compared with other prediction algorithms, our proposed algorithm performed better. In particular, we introduced the 3-δ principle based on the predicted value to modify the predicted flow value and enhance our algorithm's robustness.

Fig. 4 depicts the relationship between the number of network slices and the algorithm's running time. As the number of network slices increased, the running time increased gradually. The running time of HSR-RSV and ESFC was respectively better than other algorithms, namely because these two strategies do not include a prediction component.

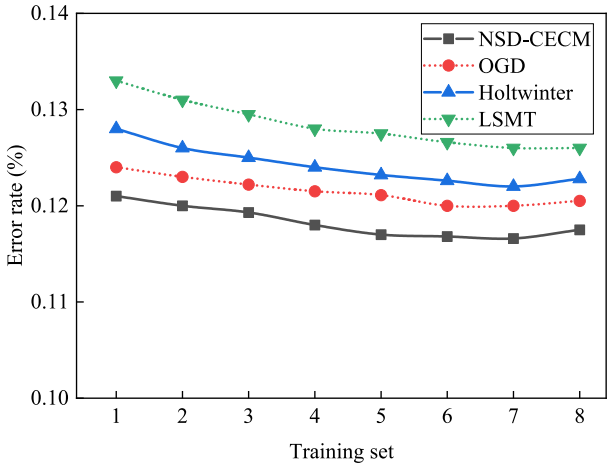

FIGURE 3. Effect of training set size.

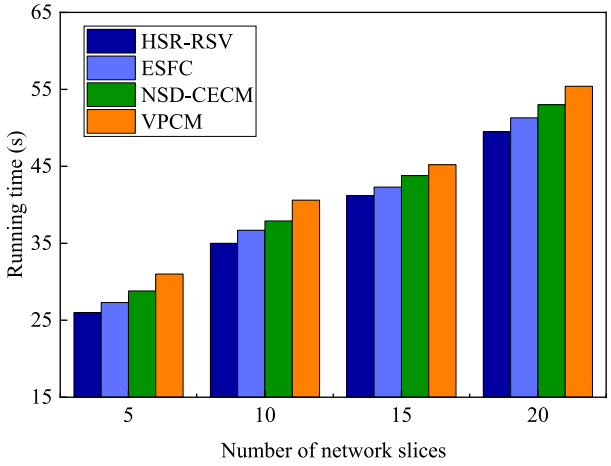

FIGURE 4. Influence of the number of network slice on running time.

HSR-RSV was superior to the ESFC algorithm due to flexible, workload-based resource allocation. When the utilization rate was low, the ESFC algorithm reduced the amount of resources accordingly, eliciting a slight increase in response time. The running time of our algorithm was preferable to that of the VPCM algorithm. We chose the original dual algorithm for link routing, which reduced the whole algorithm's running time to ensure good performance.

C. COST EVALUATION

Fig. 5 shows the relationship between the costs of different network slice deployment algorithms. We constructed eight network slices, of which the delay constraint of one network slice is 50 ms, two are 100 ms, three are 150 ms and the others are 200 ms. When $t < 260$, HSR-RSV carried the lowest cost of all algorithms, mainly because HSR-RSV set a reasonable profit function for the network slice. The energy consumption associated with network slice deployment was also not considered; as such, the initial algorithm cost was low. The cost gap then widened gradually after $t > 260$. Compared with the other algorithms, the performance of our algorithm was

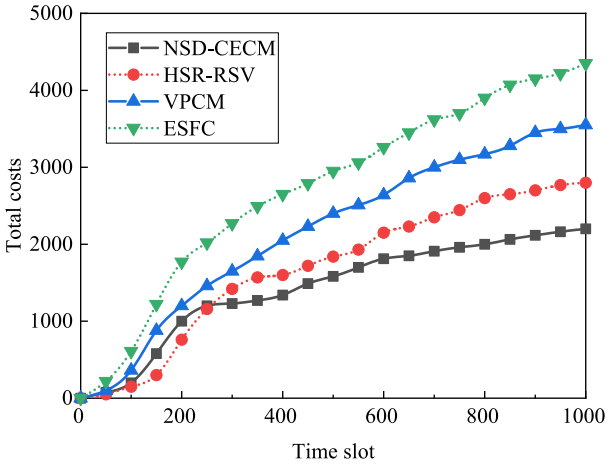

FIGURE 5. Evolution of the total costs.

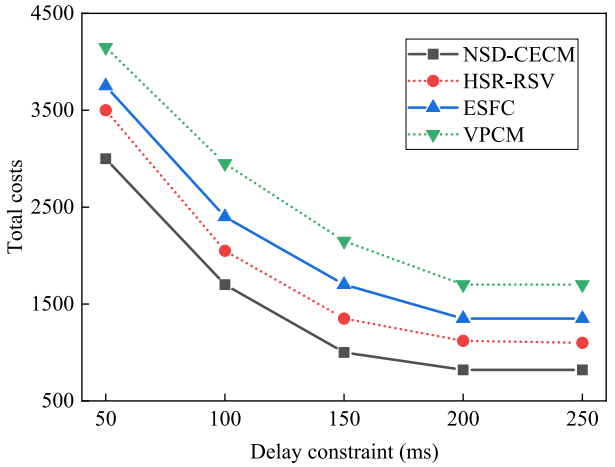

FIGURE 6. Effect of delay constraints on cost.

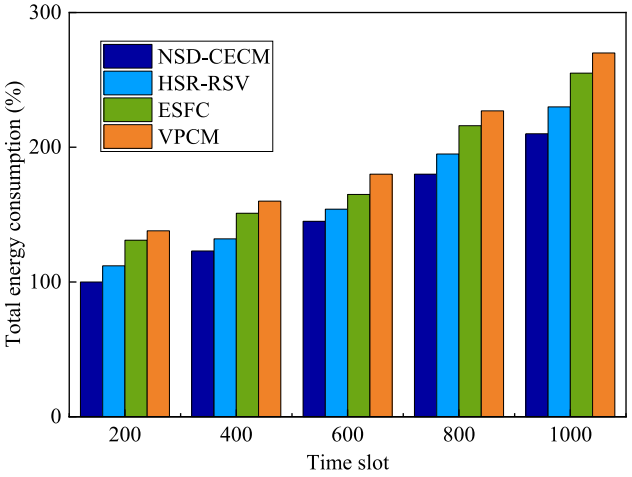

FIGURE 7. Evolution in energy consumption in different time slots.

superior. We adopted the user VNF adaptive scaling strategy in which the number and size of VNFs were set reasonably to avoid frequent updates to network topology, thus minimizing wasted network resources and further reducing the cost of VNF deployment. In exchange for service continuity, HSR-RSV limited some reconfiguration; it was therefore impossible to realize profit optimization fully. At the same time, HSR-RSV sacrificed some profit to improve service quality. The cost of network slices was thus higher than in the proposed algorithm. For VPCM, the cost of VNF over- and under-configuration was fully considered but VNF deployment costs were not; as such, the cost was higher compared to HSR-RSV. The total cost of ESFC also surpassed other algorithms. ESFC first allocated network resources according to the network slice delay and then deployed network slices. However, allocating resources based on delay increased network costs.

Fig. 6 displays the relationship between network slice delay constraints and the costs of deploying network slices under different algorithms. The delay setting ranged from 50ms to 250ms. As constraints were relaxed, the cost of deploying network slices gradually declined. However, once the delay constraint exceeded 200ms, costs tended to be stable. Because this delay requirement was too high, some high-cost paths were chosen to meet delay requirements, resulting in higher costs. As the delay requirements lessened, low-cost paths could be selected to reduce costs. When the delay requirement exceeded the minimum cost path requirement, deployment costs remained unchanged. Compared with other algorithms, our algorithm demonstrated obvious advantages in lowering the costs of network slices. HSR-RSV significantly reduced the number of network slice reconfigurations while maintaining a high slice profit; therefore, HSR-RSV performed better than the ESFC and VPCM algorithms. ESFC allocated remaining network resources based on the network slice delay. When conditions were relaxed, ESFC could identify more low-cost but high-latency paths. Yet resource allocation methods in ESFC may not allocate resources effectively. Meanwhile, VPCM did not consider the problem of network slice delay, resulting in worse delay performance than the other algorithms.

D. ASSESSMENT OF ENERGY CONSUMPTION

The total energy consumption of different algorithms was found to change as shown in Fig. 7. We took the network energy consumption of our algorithm at time slot $t = 200$ as the reference value. Other values were compared with the reference value. As the time slot increased, network energy consumption gradually increased as well, and the energy consumption of our algorithm was lowest. Specifically, our algorithm predicted network slice traffic in advance: we reasonably determined the number of VNFs and resources in each network slice and deployed slices via the proactive online algorithm, fully considering energy consumption in the network to avoid energy wastage. Other algorithms did not analyze energy consumption in such detail, hence their comparatively larger energy consumption.

E. EVALUATION OF LINK UTILIZATION

The relationship between link utilization and slot increases in different algorithms appears in Fig. 8. As the time slot

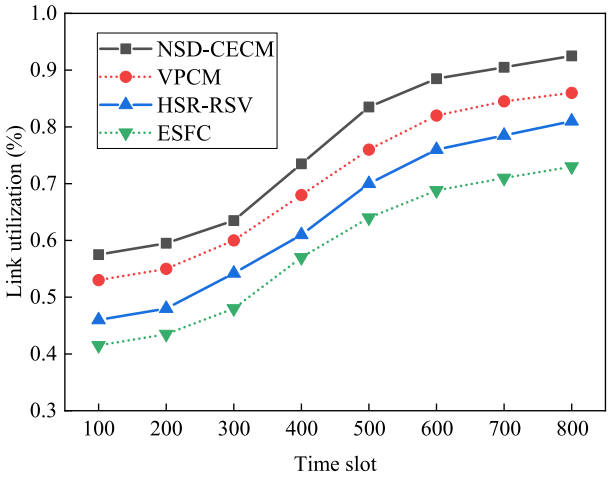

FIGURE 8. Link utilization under different time slots.

increased, network link utilization increased gradually as well. More specifically, network link utilization rose because the network topology was a stochastic model in which each physical server could serve as the flow source and destination, leading to many non-repeating routing paths. Compared with other algorithms, link utilization in ours was significantly better. The VPCM algorithm fully considered remaining link capacity to improve the network's link utilization, causing such utilization to be relatively high. The HSR-RSV resource utilization rate was lower than that of VPCM and our algorithm; because slice dimensioning generally occurs at a large time scale, slice resources may not be fully utilized due to mismatched demand and resources. HSR-RSV reserved partial resources for future traffic, resulting in low resource utilization. ESFC exhibited the worst performance due to limitations in VNF deployment that led streams to be processed in only one data center.

IX. CONCLUSION AND FUTURE WORK

In this paper, we considered the dynamic nature of a network when investigating network slice deployment. First, we built the network slice management and orchestration architecture. We then constructed a joint optimization of cost and energy consumption. Network slice service requests changed dynamically over time and demonstrated certain trends and periodicity. We proposed a prediction algorithm to forecast the network slice traffic rate to avoid frequent updates to network topology. Then we used the VNF adaptive scaling strategy based on our prediction results to determine the number of VNFs and resources to conserve resources and reduce network slice deployment costs. In addition, we proposed a proactive online algorithm to deploy network slices and elaborated on the VDA and LRA for VNF instance configuration and link routing. Finally, we compared our proposed solution with three competing solutions. Our solution demonstrated significant improvements in energy consumption, network costs, and resource utilization.

For future diversified application scenarios, such as in eMBB, uRLLC, and mMTC, 5G network slicing can meet the diverse requirements of various use cases. Dynamic deployment of low-energy network slices is inevitable alongside the development of future networks. Our proposed solution offers the advantages of low delay, low energy consumption, and high resource utilization. This solution can reduce operators' costs while meeting network slice requirements for low energy consumption. Our work represents a step forward in dynamic network slice deployment, which is pivotal to the construction of future green network slices. In the future, we plan to jointly deploy network slices with the information center network. Deployment in such scenarios is complicated and thus presents another potential avenue for subsequent research.

REFERENCES

[1] M. Aloqaily, I. A. Ridhawi, H. B. Salameh, and Y. Jararweh, "Data and service management in densely crowded environments: Challenges, opportunities, and recent developments," *IEEE Commun. Mag.*, vol. 57, no. 4, pp. 81–87, Apr. 2019.

[2] X. Li, C. Guo, L. Gupta, and R. Jain, "Efficient and secure 5G core network slice provisioning based on VIKOR approach," *IEEE Access*, vol. 7, pp. 150517–150529, 2019.

[3] H. Baba, T. Tojo, S. Yasukawa, and Y. Okazaki, "Soft-isolated network slicing evaluation for 5G low-latency services with real application microburst," in *Proc. IEEE 2nd 5G World Forum (5GWF)*, Dresden, Germany, Sep. 2019, pp. 528–531.

[4] A. Farrel, "Recent developments in service function chaining (SFC) and network slicing in backhaul and metro networks in support of 5G," in *Proc. 20th Int. Conf. Transparent Opt. Netw. (ICTON)*, Bucharest, Romania, Jul. 2018, pp. 1–4.

[5] J. Nightingale, P. Salva-Garcia, J. M. A. Calero, and Q. Wang, "5G-QoE: QoE modelling for ultra-HD video streaming in 5G networks," *IEEE Trans. Broadcast.*, vol. 64, no. 2, pp. 621–634, Jun. 2018.

[6] C. Song, M. Zhang, Y. Zhan, D. Wang, L. Guan, W. Liu, L. Zhang, and S. Xu, "Hierarchical edge cloud enabling network slicing for 5G optical fronthaul," *J. Opt. Commun. Netw.*, vol. 11, no. 4, pp. B60–B70, Apr. 2019.

[7] *System Architecture for the 5G System*, document TS 23.501 v15.5.0, 3GPP, Mar. 2019.

[8] S. Taeb, N. Shahriar, S. R. Chowdhury, M. Tornatore, R. Boutaba, J. Mitra, and M. Hemmati, "Virtual network embedding with path-based latency guarantees in elastic optical networks," in *Proc. IEEE 27th Int. Conf. Netw. Protocols (ICNP)*, Chicago, IL, USA, Oct. 2019, pp. 1–12.

[9] S. Dawaliby, A. Bradai, and Y. Pousset, "Distributed network slicing in large scale IoT based on coalitional multi-game theory," *IEEE Trans. Netw. Service Manage.*, vol. 16, no. 4, pp. 1567–1580, Dec. 2019.

[10] H. Halabian, "Distributed resource allocation optimization in 5G virtualized networks," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 3, pp. 627–642, Mar. 2019.

[11] J. Ordonez-Lucena, O. Adamuz-Hinojosa, P. Ameigeiras, P. Munoz, J. J. Ramos-Munoz, J. F. Chavarria, and D. Lopez, "The creation phase in network slicing: From a service order to an operative network slice," in *Proc. Eur. Conf. Netw. Commun. (EuCNC)*, Ljubljana, Slovenia, Jun. 2018, pp. 1–36.

[12] W. Lee, T. Na, and J. Kim, "How to create a network slice?—A 5G core network perspective," in *Proc. 21st Int. Conf. Adv. Commun. Technol. (ICACT)*, PyeongChang Kwangwoon_Do, South Korea, Feb. 2019, pp. 616–619.

[13] V. Q. Rodriguez, F. Guillemin, and A. Boubendir, "Automating the deployment of 5G network slices using ONAP," in *Proc. 10th Int. Conf. Netw. Future (NoF)*, Rome, Italy, Oct. 2019, pp. 32–39.

[14] R. Wen, G. Feng, J. Tang, T. Q. S. Quek, G. Wang, W. Tan, and S. Qin, "On robustness of network slicing for next-generation mobile networks," *IEEE Trans. Commun.*, vol. 67, no. 1, pp. 430–444, Jan. 2019.

[15] V. Balasubramanian, F. Zaman, M. Aloqaily, I. A. Ridhawi, Y. Jararweh, and H. B. Salameh, "A mobility management architecture for seamless delivery of 5G-IoT services," in *Proc. ICC-IEEE Int. Conf. Commun. (ICC)*, May 2019, pp. 1–7.

[16] S. Agarwal, F. Malandrino, C. F. Chiasserini, and S. De, "VNF placement and resource allocation for the support of vertical services in 5G networks," *IEEE/ACM Trans. Netw.*, vol. 27, no. 1, pp. 433–446, Feb. 2019.
[17] Q. Xu, J. Wang, and K. Wu, "Learning-based dynamic resource provisioning for network slicing with ensured end-to-end performance bound," *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 1, pp. 28–41, Jan. 2020.
[18] V. Sciancalepore, K. Samdanis, X. Costa-Perez, D. Bega, M. Gramaglia, and A. Banchs, "Mobile traffic forecasting for maximizing 5G network slicing resource utilization," in *Proc. IEEE INFOCOM-IEEE Conf. Comput. Commun.*, Atlanta, GA, USA, May 2017, pp. 1–9.
[19] B. Yi, X. Wang, M. Huang, and K. Li, "Design and implementation of network-aware VNF migration mechanism," *IEEE Access*, vol. 8, pp. 44346–44358, 2020.
[20] J. Pei, P. Hong, M. Pan, J. Liu, and J. Zhou, "Optimal VNF placement via deep reinforcement learning in SDN/NFV-enabled networks," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 2, pp. 263–278, Feb. 2020.
[21] P. T. A. Quang, A. Bradai, K. D. Singh, G. Picard, and R. Riggio, "Single and multi-domain adaptive allocation algorithms for VNF forwarding graph embedding," *IEEE Trans. Netw. Service Manage.*, vol. 16, no. 1, pp. 98–112, Mar. 2019.
[22] Q. Ye, W. Zhuang, X. Li, and J. Rao, "End-to-end delay modeling for embedded VNF chains in 5G core networks," *IEEE Internet Things J.*, vol. 6, no. 1, pp. 692–704, Feb. 2019.
[23] L. Ruiz, R. J. Duran, I. de Miguel, N. Merayo, J. C. Aguado, P. Fernandez, R. M. Lorenzo, and E. J. Abril, "Joint VNF-provisioning and virtual topology design in 5G optical metro networks," in *Proc. 21st Int. Conf. Transparent Opt. Netw. (ICTON)*, Angers, France, Jul. 2019, pp. 1–4.
[24] A. N. Al-Quzweeni, A. Q. Lawey, T. E. H. Elgorashi, and J. M. H. Elmirghani, "Optimized energy aware 5G network function virtualization," *IEEE Access*, vol. 7, pp. 44939–44958, 2019.
[25] L. Yala, P. A. Frangoudis, G. Lucarelli, and A. Ksentini, "Cost and availability aware resource allocation and virtual function placement for CDNaaS provision," *IEEE Trans. Netw. Service Manage.*, vol. 15, no. 4, pp. 1334–1348, Dec. 2018.
[26] V. Balasubramanian, S. Otoum, M. Aloqaily, I. Al Ridhawi, and Y. Jararweh, "Low-latency vehicular edge: A vehicular infrastructure model for 5G," *Simul. Model. Pract. Theory*, vol. 98, Jan. 2020, Art. no. 101968.
[27] I. A. Ridhawi, S. Otoum, M. Aloqaily, Y. Jararweh, and T. Bakere, "Providing secure and reliable communication for next generation networks in smart cities," *Sustain. Cities Soc.*, vol. 56, May 2020, Art. no. 102080.
[28] 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.
[29] M. Yannuzzi, R. Irons-Mclean, F. van Lingen, S. Raghav, A. Somaraju, C. Byers, T. Zhang, A. Jain, J. Curado, D. Carrera, O. Trullols, and S. Alonso, "Toward a converged OpenFog and ETSI MANO architecture," in *Proc. IEEE Fog World Congr. (FWC)*, Santa Clara, CA, USA, Oct. 2017, pp. 1–6.
[30] J. Chen, H. Cao, and L. Yang, "NFV MANO based network slicing framework description," in *Proc. IEEE Int. Conf. Consum. Electron. Taiwan (ICCE-TW)*, Yilan, Taiwan, May 2019, pp. 1–2.
[31] C.-C. Sun, G. E. Jan, S.-W. Leu, K.-C. Yang, and Y.-C. Chen, "Near-Shortest path planning on a quadratic surface with $O(n \log n)$ time," *IEEE Sensors J.*, vol. 15, no. 11, pp. 6079–6080, Nov. 2015.
[32] G. Venâncio, V. F. Garcia, L. C. Marcuzzo, T. N. Tavares, M. F. Franco, L. Bondan, A. E. Schaeffer-Filho, C. R. P. dos Santos, L. Z. Granville, and E. P. Duarte, "Beyond VNFM: Filling the gaps of the ETSI VNF manager to fully support VNF life cycle operations," *Int. J. Netw. Manage.*, p. e2068, 2019, doi: 10.1002/nem.2068.
[33] *Study on Management and Orchestration of Network Slicing For Next Generation Network*, document TR 28.801 v15.1.0, 3GPP, Jan. 2018.
[34] 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.
[35] X. Fei, F. Liu, H. Xu, and H. Jin, "Adaptive VNF scaling and flow routing with proactive demand prediction," in *Proc. IEEE INFOCOM-IEEE Conf. Comput. Commun.*, Honolulu, HI, USA, Apr. 2018, pp. 486–494.
[36] A. N. Toosi, J. Son, Q. Chi, and R. Buyya, "ElasticSFC: Auto-scaling techniques for elastic service function chaining in network functions virtualization-based clouds," *J. Syst. Softw.*, vol. 152, pp. 108–119, Jun. 2019.
[37] G. Wang, G. Feng, T. Q. S. Quek, S. Qin, R. Wen, and W. Tan, "Reconfiguration in network Slicing—Optimizing the profit and performance," *IEEE Trans. Netw. Service Manage.*, vol. 16, no. 2, pp. 591–605, Jun. 2019.

JINHE ZHOU received the B.S. and M.S. degrees in radio physics from Wuhan University, Hubei, China, in 1988 and 1991, respectively. He is currently a Professor with the School of Information and Communication Engineering, Beijing Information Science and Technology University. He has been the author of more than 50 articles published. He hosted and participated in several scientific research projects, including the National Key Project of Hi-Tech Research and Development Program of China (973 Program) and the National Natural Science Foundation of China. He awarded Beijing famous teacher award. His research topics span a large spectrum, including the 5G networks, edge computing, game theory, and green information-centric networks.

WENJUN ZHAO received the B.S. degree in electronic information engineering from the Taiyuan Institute of Technology, Taiyuan, China, in 2018. He is currently pursuing the M.S. degree in communications engineering with the Beijing Information Science and Technology University, Beijing, China. His research interests include network slicing, network cache, and game theory.

SHUO CHEN (Member, IEEE) received the Ph.D. degree in information and communication engineering from the Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 2018. She is currently a Lecturer with the School of Information and Communication Engineering, Beijing Information Science and Technology University. Her current research interests are in the areas of wireless communications and networks, with an emphasis on cognitive radios and spectrum sharing.