Deployment and Reconfiguration for Balanced 5G Core Network Slices

Xin LU‡, Xiang WANG†, Lin PANG†, Nonmembers, Jiayi LIU††, Member, Qinghai YANG††, Nonmembers, and Xingchen SONG††, Nonmembers

SUMMARY Network Slicing (NS) is recognized as a key technology for the 5G network in providing tailored network services towards various types of verticals over a shared physical infrastructure. It offers the flexibility of on-demand provisioning of diverse services based on tenants' requirements in a dynamic environment. In this work, we focus on two important issues related to 5G Core slices: the deployment and the reconfiguration of 5G Core NSs. Firstly, for slice deployment, balancing the workloads of the underlying network is beneficial in mitigating resource fragmentation for accommodating the future unknown network slice requests. In this vein, we formulate a load-balancing oriented 5G Core NS deployment problem through an Integer Linear Program (ILP) formulation. Further, for slice reconfiguration, we propose a reactive strategy to accommodate a rejected NS request by reorganizing the already-deployed NSs. Typically, the NS deployment algorithm is reutilized with slacked physical resources to find out the congested part of the network, due to which the NS is rejected. Then, these congested physical nodes and links are reconfigured by migrating virtual network functions and virtual links, to re-balance the utilization of the whole physical network. To evaluate the performance of deployment and reconfiguration algorithms we proposed, extensive simulations have been conducted. The results show that our deployment algorithm performs better in resource balancing, hence achieves higher acceptance ratio by comparing to existing works. Moreover, our reconfiguration algorithm improves resource utilization by accommodating more NSs in a dynamic environment.

key words: 5G network slicing, 5G core slices, load balancing, slice deployment, slice reconfiguration

1. Introduction

The Fifth Generation (5G) technology for cellular networks intends to provide tailored network services towards various types of vertical markets, such as Industry 4.0, wireless Virtual/Augmented Reality (VR/AR) streaming, and Internet of Things (IoT). To accommodate vertical applications with flexibility and scalability, the network-sharing-based concept of Network Slicing (NS) has been designed to establish multiple virtualized logical networks over a shared physical infrastructure, and to deliver in parallel different types of services based on the requirements and business needs of each vertical [1],[2]. The concept of NSs is expected to offer the flexibility of on-demand provisioning for multiple services, and to reduce both capital expenditure (CAPEX) and operating expense (OPEX) of the Infrastructure Provider (InP).

The concept of NS spans across the 5G network: from the Radio Access Network (RAN), go through the Transport Network (TN) and the Core Network (CN). In the scientific literature, studies related to 5G NS are classified into two categories [3]: those that focus on virtualizing and softwarizing the radio resources for RAN slicing [4]–[6] and those, referred to as Core slicing, that investigate the placement of Virtual Network Function (VNF) instances on the underlying physical infrastructure [7],[8]. The current paper belongs to the second category. For the implementation of the NSs, there are two fundamental issues: the deployment and the reconfiguration.

NS Deployment. The deployment of an NS is the first concern of an InP. When a tenant’s NS request (NSR) arrives, the InP should determine how to deploy the corresponding NS. A key concept of the 5G Core NS is the virtualized logical network, in which a set of VNFs are instantiated and connected via a virtual network. The set of VNFs forms the corresponding virtual network (VN), which allows each slice to be independently operated. The deployment of a 5G Core NS requires embedding the corresponding VN onto the physical infrastructure with respect to the allocation of physical resources. The mapping from the 5G Core slice instance towards the physical infrastructure is a virtual network embedding (VNE) optimization problem, which has been studied in the literature [9],[10]. For 5G Core NS embedding, the specific requirements of the 5G system should be considered to extend the traditional VNE problem and redesign the embedding algorithms.

We first investigate the deployment of Core NSs towards the 5G network. We emphasize load balancing of both physical nodes and links in order to avoid physical resource fragmentation and further enhance resource utilization ratio by improving the potential of accepting future unknown NS requests in practical dynamic environment [11],[12]. A load-balancing oriented 5G Core NS embedding problem (short for LBNE problem) is formulated based on max-min fairness through an Integer Linear Program (ILP) formulation. The complexity of the LBNE problem is NP-Hard, hence we design a LBNE algorithm which is a back-tracking coordinated virtual nodes and links mapping algorithm to effectively deploy the NS. In the literature, shortest path is frequently utilized to perform link mapping which can lead to the unbalanced usage of the physical links. Especially, we utilize the maximum residual capacity path for achieving a
balanced nodes and links workloads are balanced, and our solution outperforms the baseline work in the literature in regards to the NSR acceptance ratio.

**NS Reconfiguration.** The reconfiguration of the NSs is the second concern of an InP dealing with the dynamic nature of the NS management in 5G, where tenants’ NSRs arrive and depart stochastically and network parameters are time-varying. The InP should continuously reconfigure NSs to embed rejected NSRs and to improve resource utilization ratio. NS reconfiguration refers to the reallocation of the already-deployed NSs by migrating and reallocating the corresponding virtual nodes and links. In the literature, NS reconfiguration is normally considered independently from the NS deployment process. However, we propose to consider these two important aspects together for the following reasons: (1) the reconfiguration process is triggered by an NSR deployment rejection event; (2) the reconfiguration could get necessary information from the failed deployment to figure out why a rejection is produced. Hence, we design a NS deployment and reconfiguration process as shown in Fig. 1. Especially, the LBNE algorithm is re-utilized with slacked physical resources to figure out the congested part of the physical network to facilitate the reconfiguration process. Different from existing works, which tries to re-balance the underlying network [13], [14], our idea is to dredge the congested parts of the network which are identified by the LBNE deployment algorithm. Then, we design reconfiguration algorithm to effectively migrate NS virtual nodes and remap virtual links. The evaluation results show that the reconfiguration improves NSR acceptance ratio in a dynamic system environment.

Related work is summarized in Sec. 2. Then, Sec. 3 describes the system model. The LBNE problem formulation and NS deployment algorithm is presented in Sec. 4. Then, we detail the proposed reactive reconfiguration strategy for embedding a rejected NSR in Sec. 5. In Sec. 6 we evaluate the performance gain of our proposed algorithms. Finally, we conclude this work in Sec. 7.

### 2. Related Works

#### 2.1 5G Network Slicing

5G NS can be roughly classified into two categories: (1) RAN slicing includes dynamic RAN composition and slice-oriented radio resource virtualization and scheduling mechanisms; (2) CN slicing refers to the embedding of the verticals’ virtual networks towards the physical infrastructure [3], [15]. For RAN slicing, the main issue is to allocate radio resources among multiple slices according to the channel conditions and Service-Level Agreement (SLA) requirements of tenants, to achieve slice isolation without compromising multiplexing gain over the shared radio resources. On the other hand, the CN slicing also stands for an important part for 5G NS. In CN slicing, Virtual Network Function (VNF) instances form virtual networks (VNs) which represent diverse kinds of applications. And the essential problem is the embedding of the VN towards the 5G physical infrastructure, which is formulated as a virtual network embedding (VNE) problem [16], [17]. In [16], the authors propose model and algorithm for 5G NSs embedding by considering the topological information of NS virtual networks. In [17], a two-stage approach is proposed to firstly form the 5G NS by chaining of the VNFs, and then embedding the composed NSs towards the 5G physical network.

There are also works to design the interfaces and protocols between RAN slices and CN slices [18], to facilitate the interoperation between the RAN slice and CN slices. In [7], the authors propose to deploy traffic aggregation points in RANs to group and deliver RAN traffic to appropriate CN slices. In the current paper, we focus on Core 5G NSs on deploying and reconfiguring the dynamic arriving NS requests.

#### 2.2 NS deployment and Virtual Network Embedding

5G CN NS deployment is essentially a VNE problem by considering the specific requirements of the 5G system. The VNE problem has been intensively studied in the literature. VNE problem can be formulated as an ILP, and the complexity of the problem is known as NP-hard. In the literature, a large amount of efforts is devoted in designing algorithms to achieve a practical solution with efficiency. Some works adopt the two-step embedding method in which the virtual nodes are firstly placed, then based on the position of virtual nodes, virtual links are mapped as Multi-Commodity Flow problem [9]. Another class of methods try to place virtual nodes and links coordinately within one stage to benefit from exploring the relationship between nodes and links. In [19], the authors utilize the one-stage method to map nodes and links at the same time and show the benefits of this method.
Various objectives are considered for the VNE problem, such as minimizing the embedding cost and maximizing the total network revenue [17]. In [16], the authors consider multiple objectives for different types of slices, such as maximizing the total remaining resources of physical nodes and maximizing the total remaining bandwidth on physical links. However, these strategies, which are designed for the deployment of one VN, may lead to unbalanced usage of the underlying physical resources. In [13], the authors found that most VN rejections are caused by bottlenecked substrate links, and balanced resource utilization is beneficial by avoiding resource fragmentation and increasing the possibility of accepting future unknown stochastic NSRs. Hence, in the current work, we reformulate the VNE optimization problem by considering the specific topology constraints of the 5G system, and we propose load-balancing oriented 5G Core NS embedding algorithm to balance the workload of both physical nodes and links. We summarize VNE and NS related works on balancing the resource utilization of the underlying physical network (especially balancing link utilization) in the following paragraph and in Table 1.

In [20], a congestion-aware VNE problem is studied for virtual data center. A 2-step heuristic algorithm is proposed. The authors formulate the link mapping problem through linear programming to minimize the maximum link utilization, and a K-widest load-balancing splitting paths method is utilized for link balancing. In [21], a multiple-objective VNE problem is studied to save cost, energy and avoids network congestion. A coordinated heuristic algorithm is proposed to map nodes and links. Especially, the non-congested shortest path is utilized for link mapping. In [22], the authors incorporate energy consumption and resource utilization for embedding VNs. For node mapping, a node ranking method is proposed to balance node utilization. For link mapping, a weighted shortest path method is utilized by assigning weights to links according to their resource utilization ratio. In [23], the authors propose a load balancing algorithm for solving a multi-objective VNE. The algorithm incorporates a VNE algorithm by measuring physical node load value for node mapping process. In [24], the authors study VNE in the IoT networks. Especially, VNE is solved through a hybrid genetic algorithm, links load balancing is achieved by updating links weight according to the link bandwidth occupancy. Finally, in the current work, we formulate a load-balancing oriented VNE problem for the 5G network to achieve balanced workloads for both physical nodes and links. For node mapping, node remaining capacity is considered to prioritize nodes with lower work loads. For link mapping, we utilized the maximum residual capacity path to avoid exclusively utilizing the shortest paths.

2.3 NS Reconfiguration and Virtual Network Reconfiguration

One of the most significant benefits of network slicing is providing InPs the flexibility to create and manage slices in a dynamic environment depending on the time-varying network parameters [25]. The dynamic reorganization of the already embedded virtual networks is referred as virtual network reconfiguration (VNR). There are two types of different reconfiguration policies: the proactive strategy reconfigures the system periodically, whereas the reactive strategy is triggered by some events (such as VN rejection). Periodic network reconfiguration may incur more costs and lead to unnecessary service performance degradation due to VNF migrations. In this work, we adopt the reactive reconfiguration strategy and propose algorithm to re-embed a rejected NSR.

For proactive reconfiguration, the objective is to balance the workloads of the underlying physical network by migrating congested nodes and links to prevent performance degradation. In [26], the authors study the VNE problem for distributed cloud by analyzing the periodical resource demands of VNs. In [27], the authors apply forecasting techniques to adjust the allocated slice resources so as to optimize the network utilization for the 5G network. In [28] a proactive reconfiguration mechanism for 5G network slices is proposed for finding a sequence of feasible configurations towards the final one. For reactive reconfiguration, it is triggered by network change events. In [29], re-embedding schemes for evolving VN requests with time varying characteristics are studied to minimize physical node migration cost. In [30], the authors tackle the problem of reconfigure VNs affected by underlying physical network failure. In [31], the authors propose a hybrid strategy to take the advantages of both reconfiguration policies. Especially, they utilize a resource reservation mechanism to reserves partial resources for future NSRs.

There are also VNR works related on balancing the workloads of the underlying physical network. In [14], the authors develop a VN reconfiguration scheme to balance the load on the physical network. However, they focuses on the selection and remapping and balancing the workloads of physical nodes. In [33], a proactive reconfiguration algorithm for VN re-embedding is proposed to minimize the number of over utilized substrate links and total bandwidth cost on the SN. In [32], a reconfiguration cost aware VNE problem is studied. The authors design a migration policy in response to known and periodic cycle-stationary traffic pattern based on Markov Decision Process. As we sum up in Table 1, comparing to the aforementioned related works. We especially focus both on the deployment and reconfiguration strategies for the 5G CN NSs. Our contributions are as follows:

- Firstly, we consider both the deployment and reconfiguration. Reactive VN reconfiguration is triggered by a NSR failure event. We believe that the reconfiguration could benefit from figuring out the reason of the rejection. As shown in Fig. 1, the deployment algorithm is reutilized to find out the congested part of the physical network to facilitate the reconfiguration process.
- Secondly, we design a load balancing oriented 5G CN
NS deployment mechanism. Our objective is to balance the workload of the underlying network for both physical nodes and links. For node mapping, physical nodes are selected based on their remaining resource and topology importance. For link mapping, we especially utilize the maximum residual capacity path, rather than the conventionally utilized shortest path to balance the workloads of physical links.

- Finally, we design a comprehensive NSs reconfiguration process to remanage the congested part of the physical network by migrating both virtual nodes and links to re-balance the underlying physical network to accept the rejected NSR.

3. System Model

As shown in Fig. 2, the deployment of the 5G CN NS is essentially the mapping of the 5G VNFs towards the physical network. In the figure, these are two types of slices to be deployed. The ultra Reliability and Low Latency Communication (uRLLC) slice targets for applications with ultra low delay requirements. The enhanced Mobile Broadband (eMBB) slice targets for mobile broadband use cases, such as AR/VR media and applications. Then the mapping strategy of CN NSs can be different. For uRLLC NS, the related 5G VNFs need to be mapped towards the edge Data Center (DC) to meet the low delay requirements. For eMBB NS, the related MEC platform need to be mapped towards the edge DC, whereas the control plane VNFs can be mapped towards the core DC. Essentially, the mapping of the 5G CN NS’s VN is a VNE problem by considering the specific topology and service requirements of the 5G network.

The overall NS deployment and reconfiguration process is shown in Fig. 1. When a tenant’s NSR arrives, the InP firstly determine the VN based on the tenant’s service and SLA requirements. Then, the InP tries to implement the NS by the LBNE algorithm. If the NS is deployed successfully, the process ends. When a first rejection is returned by the LBNE algorithm, the InP further tries to rearrange the network to still embed the rejected NSR. The rearrangement starts by re-run the LBNE algorithm with slacked physical resources to find out the congested part of the physical network. The reconfiguration algorithm is launched to migrate the already deployed NSs and re-embed the rejected NSR. If the NSR is finally re-embedded, the process ends with success, otherwise the reconfiguration returns a rejection and the NSR is finally rejected without deployment.

For a practical dynamic system, the life-cycle management of a NS is performed by the NFV Management and Orchestration (MANO) entity to orchestrate the individual VNFs in the NS instance. After the NS’s life time expires, the NS is deleted and physical network resources are released. In the current paper, we do not treat the NSs leaving events. We focus on treating the NSR arriving events by the deployment and reconfiguration strategy to improve physical resource utilization ratio. In the following, we describe the underlying physical infrastructure model and the NSR model.

3.1 Physical Infrastructure Model

The physical infrastructure for 5G Core NS starts from the mobile edge, continues through the transport (fronthaul and backhaul), and ends until the core network. The physical infrastructure is modeled by a graph $G_I(N_I, E_I)$ where $N_I$ denotes the physical network nodes and $E_I$ stands for physical communication links. One physical network node is denoted by $n^I_i \in N_I$. Hence, we distinguish three kinds of physical nodes: (1) the access nodes set $N^A_i$, such as edge DCs that are capable for accommodating virtual Mobile Edge Computing (MEC) servers; (2) the transport network nodes set $N^T_i$ which denotes TN optical switches; (3) the core network nodes set $N^C_i$, which includes core DCs for CN data plane and control plane VNFs virtualization. Hence, we have $N_I = N^A_i \cup N^T_i \cup N^C_i$. Without loss of generality, we assume that $N^A_i$ and $N^C_i$ nodes are suitable for accommodating multiple types of virtualized VNFs for the flexible deployment of NSs. Moreover, each physical node $n^I_u \in N^A_i \cup N^C_i$ is characterized by its capacity $C^I_u$ which represents the available computing resources of the node, the total computing capacity of node $n^I_u$ is represented by $C^I_u$ by adding the allocated resource towards the already deployed network slices.

Physical communication links are represented by $E_I$, with $e^I_{uw} \in E_I$ denote the physical link connecting two physical nodes $n^I_u$ and $n^I_v$. For each link $e^I_{uw} \in E_I$, we denote by $B^I_{uw}$ the available bandwidth of the link, and $L^I_{uw}$ the total bandwidth capacity. $L^I_{uw}$ denotes the transmission delay of the link.

3.2 Network Slice Request Model

As shown in Fig. 2, the 5G core network architecture includes data plan VNFs (user plane functions (UPF)) and control plan VNFs (such as mobility management functions (AMF), session management function (SMF) and policy control function (PCF), etc.). Deploying such a cloud-native network means finding the appropriate positions for each 5G VNFs. Considering the interfaces connecting these VNFs, we formulate the 5G CN NS by a directed graph $G_S(N_S, E_S)$, where $N_S$ denotes for the 5G VNFs (virtual network nodes) and $E_S$ stands for virtual links connecting the virtual nodes to form the VN. After the deployment of the 5G CN NS, the end-to-end slices are formed by connecting the UPF towards the 5G AN through the N3 interface, and the AMF towards the AN and UE through the N1 and N2 interfaces. In this paper, we typically focus on the deployment of the 5G core network slices.

One virtual network node is denoted by $n^S_i \in N_S$. Each virtual network node represents a virtual VNF instance, such as virtual MEC server, virtual AMF, and virtual gateway, etc. We denote by $C^S_i$ the physical resource requirement of the virtual node $n^S_i$. A virtual link is represented by $e^S_{ij} \in E_S$, where...
**Table 1** Comparison of VNE and VNR works

<table>
<thead>
<tr>
<th>Paper</th>
<th>System</th>
<th>VNE</th>
<th>VNR</th>
<th>Findings</th>
</tr>
</thead>
<tbody>
<tr>
<td>et al.[16]</td>
<td>5G NS</td>
<td>yes</td>
<td>no</td>
<td>Coordinated mapping algorithm for 3 scenarios with different objectives.</td>
</tr>
<tr>
<td>et al.[17]</td>
<td>5G NS</td>
<td>yes</td>
<td>no</td>
<td>2-stage algorithm to minimize embedding cost.</td>
</tr>
<tr>
<td>et al.[20]</td>
<td>Data Center</td>
<td>yes</td>
<td>no</td>
<td>Load balancing Oriented VNE, 2-step mapping, minimize maximum link utilization, use splitting K-widest path for link balancing.</td>
</tr>
<tr>
<td>et al.[21]</td>
<td>General</td>
<td>yes</td>
<td>no</td>
<td>Load balancing Oriented VNE, coordinated node and link mapping, use non-congested shortest path for link mapping.</td>
</tr>
<tr>
<td>et al.[22]</td>
<td>General</td>
<td>yes</td>
<td>no</td>
<td>Load balancing Oriented VNE, 2-step mapping, use weighted shortest path for link mapping.</td>
</tr>
<tr>
<td>et al.[23]</td>
<td>General</td>
<td>yes</td>
<td>no</td>
<td>Load balancing Oriented VNE, 2-step mapping, use node load value for node mapping.</td>
</tr>
<tr>
<td>et al.[24]</td>
<td>IoT network</td>
<td>yes</td>
<td>no</td>
<td>Load balancing Oriented VNE, genetic algorithm, use link weight update mechanism for link mapping.</td>
</tr>
<tr>
<td>et al.[26]</td>
<td>Cloud</td>
<td>no</td>
<td>Proactive</td>
<td>Proactive reconfiguration by analyzing the periodical resource demands of VNs.</td>
</tr>
<tr>
<td>et al.[27]</td>
<td>5G NS</td>
<td>no</td>
<td>Proactive</td>
<td>Proactive reconfiguration for the 5G NSs through traffic analysis and prediction.</td>
</tr>
<tr>
<td>et al.[28]</td>
<td>5G NS</td>
<td>no</td>
<td>Proactive</td>
<td>Proactive reconfiguration for the 5G network through configuration changes.</td>
</tr>
<tr>
<td>et al.[29]</td>
<td>General</td>
<td>no</td>
<td>Reactive</td>
<td>Reactive reconfiguration to reduce node migration cost.</td>
</tr>
<tr>
<td>et al.[31]</td>
<td>5G NS</td>
<td>no</td>
<td>Hybrid</td>
<td>Proactive and reactive reconfiguration for 5G NS, resource reserve mechanism for future NSRs.</td>
</tr>
<tr>
<td>et al.[33]</td>
<td>General</td>
<td>no</td>
<td>Proactive</td>
<td>Load balancing reconfiguration for minimizing physical bandwidth consumption and overloaded links.</td>
</tr>
<tr>
<td>et al.[14]</td>
<td>General</td>
<td>no</td>
<td>Proactive</td>
<td>Load balancing reconfiguration for balancing workload of physical nodes.</td>
</tr>
<tr>
<td>Current work</td>
<td>5G NS</td>
<td>yes</td>
<td>Reactive</td>
<td>Load balancing Oriented VNE and VNR, coordinated VNE algorithm, node workload for node selection, maximum residual capacity path for link mapping, node and link load-balancing oriented reconfiguration.</td>
</tr>
</tbody>
</table>
with $B_{ij}^S$ denoting its bandwidth requirement, and $L_{ij}^S$ the link delay requirement.

4. Load Balancing NS Deployment

4.1 LBNE Problem Formulation

In this section, we formulate the problem of allocating physical network node and link resources to implement the NSR as an Integer Linear Program. Deploying a NS corresponds to mapping the VN towards the physical infrastructure. We define two sets of binary variables:

- The node mapping decision variable $x_{ui}^n$ takes the value 1 if the virtual node $n_i^S$ is placed on the physical node $n_u^I$, and takes the value 0 otherwise.
- The link mapping decision variable $y_{uv}^{ij}$ takes the value 1 if the virtual link $e_{ij}^S$ is deployed through the physical link $e_{uv}^I$, and takes the value 0 otherwise.

We first formulate the 5G network topology related constraints. The 5G Core network utilizes a cloud native architecture as shown in Fig. 2, which facilitate the formulation of 5G VNFs mapping. In the 5G network, VNFs are position specific. The MEC VNFs should be mapped towards the edge DCs. The 5G control plan VNFs can be deployed on either the edge or the core DCs. To describe such location specification, we define a binary location parameter $\text{loc}_{ni}^v$ which takes value 1 if the corresponding virtual node $n_i^S$ can be placed on the corresponding physical node $n_u^I$, and 0 otherwise. Hence, we introduce a location constraints in (1).

$$x_{ui}^n \leq \text{loc}_{ni}^v, \quad \forall n_u^I \in N_I, \; n_i^S \in N_S. \quad (1)$$

Especially, the location parameter $\text{loc}_{ni}^v$ should be properly defined. For edge VNFs, $\text{loc}_{ni}^v = 0, \forall n_u^I \in N_I^E \cup N_I^C$; for core VNFs, $\text{loc}_{ni}^v = 0, \forall n_u^I \in N_I^A \cup N_I^F$.

Then, we consider the following VNE related constraints. Firstly, each virtual nodes must be mapped to one physical node.

$$\sum_{n_u^I \in N_I} x_{ui}^n = 1, \quad \forall n_i^S \in N_S. \quad (2)$$

The physical node and link resource constraint should be respected.

$$\sum_{n_i^S \in N_S} C_{ij}^S x_{ui}^n \leq C_{ui}^I, \quad \forall n_u^I \in N_I. \quad (3)$$

$$\sum_{e_{ij}^S \in E_S} B_{ij}^{uv} y_{uv}^{ij} \leq B_{uv}^I, \quad \forall e_{uv}^I \in E_I. \quad (4)$$

The mapped physical links should be bounded by the virtual link delay requirement to fulfill the delay SLA requirements of the NS.

$$\sum_{e_{ij}^S \in E_S} L_{ij}^{uv} y_{uv}^{ij} \leq L_{ij}^S, \quad \forall e_{ij}^S \in E_S. \quad (5)$$

One virtual link is mapped as a flow from the physical source to sink. Hence, we impose the following flow related constraints.

$$\sum_{n_i^S \in N(n_u^I)} y_{uv}^{ij} \leq 1, \quad \forall n_u^I \in N_I, \; e_{ij}^S \in E_S. \quad (6)$$

$$\sum_{n_i^S \in N(n_u^I)} y_{uv}^{ij} - y_{ij}^c = x_{ui}^n - x_{uj}^n, \quad \forall n_u^I \in N_I, \; e_{ij}^S \in E_S. \quad (7)$$

in which $N(n_u^I)$ stands for the neighbor set for the physical node $n_u^I$. Constraint (6) ensures that there is no loop along the physical path mapped for the virtual link. Constraint (7) satisfies the flow conservation constraints.

Finally, we set forth the problem objective formulation. In a dynamic system, in which NSRs arrive stochastically,
one reasonable strategy to allocate the infrastructure for the current NSR would be to evenly utilize the network resources in order to accommodate the future unknown NSRs. For example, by avoiding exclusively utilizing specific physical nodes and links resources, the resource fragmentation can be mitigated to further enhance resource utilization ratio. Hence, a practical objective of the NS deployment problem would be balancing workloads of both physical nodes and links, subjects to the NS’s SLA requirement and physical topology importance. We formulate the problem as minimizing the maximum node workload variable $W_{\text{max}}$ and the maximum link workload variable $W_{\text{max}}$.

$$W_n = \frac{(C^l_n - C^l_u) + \sum n^u_i C_i^S x^u_i}{C_i^l}, \quad \forall n^u_i \in N_I.$$

The workload of physical link $e_{uv}$ is calculated as:

$$W_{uv} = \frac{(B^l_{uv} - B^l_{uv}) + \sum e^l_{ij} B^S_{ij} y^u_{ij}}{B^l_{uv}}, \quad \forall e_{uv} \in E_I.$$

We define:

- The maximum node workload variable $W_{\text{max}}$ as $W_{\text{max}} \geq W_n, \forall n^u_i \in N_I$.
- The minimum node workload variable $W_{\text{min}}$ as $W_{\text{min}} \leq W_n, \forall n^u_i \in N_I$.
- The maximum link workload variable $W_{\text{max}}$ as $W_{\text{max}} \geq W_{uv}, \forall e_{uv} \in E_I$.
- The minimum link workload variable $W_{\text{min}}$ as $W_{\text{min}} \leq W_{uv}, \forall e_{uv} \in E_I$.

Then, the workload balancing for both nodes and links capacity NS embedding problem (short for LBNE problem) is formulated as Problem P1 based on max-min fairness principle:

$$\min \{x^l_i, y^u_{ij}\} \quad \alpha(W_{\text{max}} - W_{\text{min}}) + \beta(W_{\text{max}} - W_{\text{min}})$$

$$\text{s.t. (1), (2), (3), (4), (5), (6) and (7).}$$

where $\alpha$ and $\beta$ are two positive tunable parameters such that $\alpha + \beta = 1$. These two parameters are utilized to tune the objective towards node balancing and link balancing. The complexity of the Problem is NP-Hard, hence we focus on developing fast algorithms to practically tackle the problem for real and large system setting.

4.2 LBNE Deployment Algorithm

In this section, we describe our LBNE deployment algorithm. First of all, by measuring the resource and topology importance, we determine a sequence of virtual nodes such that nodes with more importance are deployed firstly. Then, we design a coordinated virtual node and link mapping algorithm to alternatively map virtual nodes and links. Especially, both workloads on physical nodes and links are considered for balancing workloads. In the literature, virtual links are mapped through the shortest path, which can drain the bandwidth resources in the fixed shortest path in a given topology and lead to resource fragmentation. We propose to utilize the maximum residual capacity path in the link mapping process to dynamically select paths with more residual capacity, and hence to improve the physical resource utilization ratio.

4.2.1 Network Node Importance

The virtual nodes are ordered by their importance in the process of node mapping. We first introduce how to measure their importance by considering both resource and topology information. The following concepts of node importance are conventional in general networks, hence we manifest the measurement of the virtual nodes, that of the physical nodes can be derived similarly.

**Resource Importance.** We define the resource importance of $n^S_i$ as:

$$\text{RI}(n^S_i) = C^S_i + \sum B^S_{ij}. \quad (10)$$

Thus, we consider both the computing resource required by $n^S_i$, and the bandwidth requirement of virtual links connected to $n^S_i$. The definition of $\text{RI}(n^S_i)$ is a conventional way to reflect virtual node resource requirement, and higher $\text{RI}$ value means higher importance.

**Topology Importance.** We utilize the following conventional concepts to measure the topological characteristics of $n^S_i$.

- **Degree Centrality.** The degree of $n^S_i$ measures the number of links that connect to it. The normalized node degree is defined as $d^S_{n^S_i} = \frac{d_{n^S_i}}{\hat{d}_{n^S_i}}$, where $d_{n^S_i}$ is node degree of $n^S_i$ and $|N_S|$ is the number of virtual nodes.
- **Betweenness Centrality.** The betweenness centrality calculates the fraction of all-pairs shortest paths pass through the node, which is defined as $b^S_{n^S_i} = \frac{1}{|N_S| \cdot |N_S| - 2} \sum_{j,k \neq i} p_{jk}(n^S_i) \cdot p_{jk}$, where $p_{jk}(n^S_i)$ is the number of those pathes goes through node $n^S_i$. We further normalize it into $b^0_{n^S_i} = \frac{1}{2^{|N_S| - 2}}$. 
- **Closeness Centrality.** The closeness centrality measures the distance of the node towards all other nodes in the network, which is defined as $c_{n^S_i} = \frac{1}{|N_S| \cdot |N_S| - 2} \sum_{j,k \neq i} d(n^S_i, n^S_j)$. $d(n^S_i, n^S_j)$ denotes the distance between node $n^S_i$ and $n^S_j$ in the network. We also normalize it into $c^0_{n^S_i} = \frac{1}{|N_S| \cdot |N_S| - 2}$.

Then, the topology importance of $n^S_i$ is defined as:
The \( TI(n^S) \) value is normalized into between 0 and 3, and higher TI value means higher topology importance. Finally, we define the node importance of \( n^S_i \) by considering both resource importance and topology importance. The node importance is proportional to the above two values, hence it is calculated as:

\[
NI(n^S_i) = RI(n^S_i) TI(n^S_i). \tag{12}
\]

Then, we obtain the sorted list of virtual nodes \( L(N_S) \) from the following process:

- For all \( n^S_i \in N_S \), calculate its \( NI(n^S_i) \) based on Eq. (12).
- Find \( n^S_{i_1} \) as the virtual node with the largest NI value. Build the searching tree \( T(n^S) \) on top of \( G_S \) by setting \( n^S_{i_1} \) as the root and using the Breadth-First Search algorithm.
- For each layer of \( T(n^S) \), sort the virtual nodes in NI decreasing order, and push this sorted list of virtual nodes into \( L(N_S) \).

### 4.2.2 Coordinated Virtual Node and Link Mapping Algorithm

Our proposed algorithm, which coordinately maps virtual nodes and links, is depicted in detail in Algorithm 1:

- The input of the algorithm is the physical network model \( G_T \) and the NS model \( G_S \). The output of the algorithm is the mapping from the NSR to the physical network. If the NSR cannot be mapped due to the violation of the problem P1’s constraints, a rejection is returned.
- The algorithm tries to map the virtual node in the list \( L(N_S) \), which is ordered by virtual nodes network importance values, one by one (step 4). For the current to be mapped virtual node \( n^S_i \), define the set of physical nodes towards which \( n^S_i \) can be mapped as \( Loc(n^S_i) \) (step 5, for Constraint (1)). Then, depending on the role of \( n^S_i \) in the virtual network, we further classify two sub-situations.
- First, if \( n^S_i \) is the root of the BFS tree (step 6), the candidate physical node set \( Cand(n^S_i) \) contains nodes in \( Loc(n^S_i) \) with available computing resources greater than the resource requirement of \( n^S_i \) (step 7, for Constraints (3)). Then, we map \( n^S_i \) to the physical nodes in \( Cand(n^S_i) \) with the highest NI value (step 8 and 9). Hence, for root virtual node, the physical node is selected by comprehensively considering both its available resources and network importance.
- If \( n^S_i \) is not the root of the BFS tree (step 10), it implies that the parent node in the BFS tree has been already mapped. Let \( n^S_p(n^S_i) \) denote the parent virtual node of \( n^S_i \), and \( n^S_p(n^S_i) \) denotes the physical node to which \( n^S_p(n^S_i) \) is mapped. Then, we try to map both the virtual node \( n^S_i \) and already mapped virtual nodes (including \( n^S_p(n^S_i) \)). We first construct the candidate physical node set \( Cand(n^S_i) \) which contains \( Loc(n^S_i) \) with sufficient large available resources and within \( k \) hops from \( n^S_p(n^S_i) \) in \( G_T \) (denoted as the set \( N^k_{p^S}(n^S_{p^S}) \), step 13). Then, we sort physical nodes in \( Cand(n^S_i) \) in decreasing order (step 14). Hence, for non-root virtual nodes, we intend to map it towards physical nodes with lower workloads.
- Then, we try to map the virtual node and links simultaneously. Specifically, we find the maximum residual capacity path between the current candidate physical node and \( n^S_p(n^S_i) \) with the required bandwidth (step 16 to 18, for Constraints (4)). If the virtual link’s SLA is guaranteed, the node mapping is also determined (step 19 to 21, for Constraints (5)). Otherwise, we continue to try the next candidate physical node in \( Cand(n^S_i) \), until the corresponding virtual node and link is successfully mapped (step 22). If the current virtual node cannot be mapped to any physical node in \( Cand(n^S_i) \), a rejection is returned such that the NS cannot be deployed.

Our algorithm has the following advantages: (1) Net-

### Algorithm 1 5G Core LBNE Algorithm

1. **Input:** The physical network \( G_T \), the NSR \( G_S \).
2. **Output:** The mapping from NSR \( G_S \) to \( G_T \).
3. Obtain the sorted virtual node list \( L(N_S) \).
4. for \( n^S_i \in L(N_S) \) do
   5. Define \( Loc(n^S_i) = \{ n^S_p | loc^p = 1, \forall n^S_p \in N_T \} \).
   6. if \( n^S_i \) is the root of the BFS tree then
      7. Let \( Cand(n^S_i) = \{ n^S_p | C^p_t > C^S_t, \forall n^S_p \in N_T \} \cap Loc(n^S_i) \) the physical candidate node set of \( n^S_i \).
      8. Sort \( n^S_p \in Cand(n^S_i) \) in NI non-increasing order.
      9. Map the current virtual node \( n^S_i \) to the physical node in \( Cand(n^S_i) \) with the largest NI value.
   10. else
      11. Let \( n^S_p(n^S_i) \) be the parent virtual node of \( n^S_i \) in the BFS tree.
      12. Let \( n^S_p(n^S_i) \) be the physical node to which \( n^S_i \) is mapped.
      13. Let \( Cand(n^S_i) = Loc(n^S_i) \cap \{ n^S_p | C^p_t > C^S_t \} \cap N^k_{p^S}(n^S_{p^S}) \).
      14. Sort \( n^S_p \in Cand(n^S_i) \) in decreasing order.
      15. for \( n^S_p \in Cand(n^S_i) \) do
         16. Map the virtual link between \( n^S_i \) and \( n^S_p(n^S_i) \):
         17. Remove links in \( G_T \) with bandwidth lower than \( B^S(n^S_i,n^S_p(n^S_i)) \).
         18. Find the maximum residual capacity link between \( n^S_p(n^S_i) \) and \( n^S_i \) in \( G_T \).
         19. if The link SLA is satisfied then
             20. Map \( n^S_p(n^S_i) \) to \( n^S_p \).
         21. else
             22. Go to step 15 to try the next \( n^S_p \).
      23. end if
      24. end for
      25. if The current virtual node \( n^S_i \) cannot be mapped to any physical node in \( Cand(n^S_i) \), then
         26. Reject the NSR.
      27. end if
      28. end if
      29. end for
work importance are considered such that more important virtual nodes are mapped firstly; (2) The virtual neighbor nodes are mapped to physical nodes within a certain range (k−hops) to reduce link mapping bandwidth resource consumption; (3) The coordinated node and link mapping process try to map both nodes and links simultaneously. Moreover, we adopt a back-tracking mechanism to further improve the NSRs accept ratio. If the link mapping fails, the algorithm tries another physical candidate node; (4) Physical nodes and links are selected by considering the actual workloads. Especially, virtual links are mapped to physical paths with the maximum residual capacity [34], to avoid draining intensively the shortest path bandwidth, and to balance links workloads dynamically in the NS deployment process.

5. Reactive NS Reconfiguration Strategy

In Sec. 4, we introduced the deployment algorithm for an arriving NSR. In practical running, an NSR can be rejected due to lack of physical resources. In such situation, the InP could try to reconfigure the network by reorganizing the already implemented slices in order to accommodate the rejected NSR. Such reconfiguration strategy is designated as a reactive one, which is triggered after the NSR rejection.

For the NS reconfiguration strategy, there are mainly three issues to concern:

1. Which are the resource deficit physical nodes and links that need to be reconfigured?
2. For these physical nodes, which are the virtual nodes and links that need to be migrated?
3. For these virtual nodes and links, to which physical destination nodes and links they should be migrated?

For the reactive reconfiguration strategy, our main idea is not to blindly rebalance the workload of the network, but to dredging the congested parts of the network in order to accommodate the rejected NSR. Such reconfiguration strategy is designated as a reactive one, which is triggered after the NSR rejection.

The CN set contains physical nodes either endures computing resource deficits, or one of its links suffers from bandwidth deficits. Then, we further define a BN set, which only contains physical nodes with link bandwidth congestion:

$$\mathbb{B}N = \{ n_u^t | \sum_{s=1}^{S+1} C_{us}^t > \hat{C}_u^t \text{ or } \exists v : \sum_{s=1}^{S+1} B_{uv}^t > \hat{B}_{uv}^t \}$$

Consequently, $\mathbb{B}N \subseteq \mathbb{C}N$. Clearly, if virtual nodes and links on these physical nodes could be reconfigured to release some resources, the rejected NSR could be embedded.

Please note that, with slacked capacity, our NSR deployment Algorithm 1 selects physical nodes and links with the minimum workloads, which are also appropriate to be rearranged with low reconfiguration costs. Then, it is necessary to measure the congestion level of physical nodes.

$$\mathbb{C}L(n_u^t) = (1 - \frac{\sum_{s=1}^{S+1} C_{us}^t - \hat{C}_u^t}{\hat{C}_u^t}) \sum_{v} \left( 1 - \frac{\sum_{s=1}^{S+1} B_{uv}^t - \hat{B}_{uv}^t}{\hat{B}_{uv}^t} \right) \left| N(n_u^t) \right|$$

Consequently, a lower CL value implies higher resource deficits. Then, we determine a list of the CN nodes as $L(\mathbb{C}N)$, according to which to treat the congested physical nodes sequentially in the reconfiguration process. There are two situations to consider:

• Firstly, for $n_u^t \in (\mathbb{C}N - \mathbb{B}N)$, these nodes endures
computing resource deficits. Hence, the embedded virtual nodes should be migrated to release computing resources. At the same time, virtual node migration also leads to reconfiguration of links which could possibly release some bandwidth resources. Hence, such physical nodes should be firstly considered to reconfigure. We sort these physical nodes in CL increasing order and push them into L(CN).

• Then, for \( n_u \in BN \), the virtual links should be remapped. Hence \( BN \) is sorted in CL value increasing order and appended in the list L(CN). Then, L(CN) is returned for the next procedure.

5.2 Determine to-be-migrated Virtual Nodes

After determining the sequence of to-be-reconfigured physical nodes list L(CN), the next problem is, for a given physical node \( n_u \), which virtual nodes and links should be reconfigured such that sufficient resources can be released with minimum reconfiguration cost. Without loss of generality, we first consider the physical node in the \( (CN - BN) \) set in which the computing resources need to be reconfigured.

Firstly, a set of virtual nodes are determined as the candidates (denoted as the \( CVN \) set) for migration, of which the life time is greater than a given threshold \( T_l \):

\[
CVN(n_u) = \{n^*_i | x^u_i = 1 \text{ and } T_s \geq T_l\} \tag{16}
\]

The migration cost of reconfiguring a virtual node \( n^*_i \in CVN \) is represented by a parameter \( C_{\text{mig}}^i \), which depends on the network function type of the virtual node, the amount of resources required by this virtual node, the number of virtual links connected to this virtual node, etc. We further define a binary variable \( z^*_i \) which corresponds to each virtual node \( n^*_i \in CVN(n_u) \), \( z^*_i = 1 \) represents that the corresponding virtual node is selected to migrate, and 0 otherwise. Then, for each congested physical node \( n_u \), we formulate a migration virtual node selection problem as:

\[
\begin{align*}
\min \quad & C_{\text{mig}}^i z^*_i \\
\text{s.t.} \quad & \sum_{n^*_i \in CVN(n_u)} C^i z^*_i \geq \sum_{s \in [1,S+1]} C^l_{us} - C^l_u \\
\end{align*}
\tag{P2}
\]

Problem (P2) is a typical 0-1 Knapsack Problem which is NP-Hard, pseudo-polynomial algorithms exist to solve this problem [35].

5.3 Determine Candidate Migration Destination Physical Nodes

Problem (P2) selects a set of virtual nodes, a consequent question is: a virtual node can be migrated to which physical nodes? Suppose the current to-be-migrated virtual node is \( n^*_i \) on congested physical node \( n_u \), then we define the candidate migration destination physical node as:

\[
\begin{align*}
\mathcal{CPN}(n^*_i) = \{n^*_i | & \text{loc}^u_i = 1 \text{ and } n^*_i \notin CN \text{ and } C^l_{v} > C^l_{i} \text{ and } n^*_i \in N^K_i (n_u)\} \tag{17}
\end{align*}
\]

Hence, these physical nodes can hold the virtual node, non-congested, having sufficient resources and within \( k \)-hops from the original embedding physical node. Then, we sort nodes in \( \mathcal{CPN}(n^*_i) \) in \( W_u \) non-decreasing order and try to map the related virtual links. The link remapping process is introduced in detail in Sec. 5.5.

5.4 Determine to-be-reconfigured Virtual Links

For physical nodes endure bandwidth deficits, the virtual links should be reconfigured to release sufficient bandwidth resources. For a physical node \( n_u \in BN \), we first define its congested link set as:

\[
\mathcal{CPL}(n_u) = \{e^l_{uv} | B^l_{uv} > B^l_{uv} \} \tag{18}
\]

For a physical link \( e^l_{uv} \in \mathcal{CPL}(n_u) \), we define the candidate remapping virtual links set as:

\[
\mathcal{CVL}(e^l_{uv}) = \{e^s_{ij} | y^s_{ij} = 1 \text{ and } T_s \geq T_i\} \tag{19}
\]

Hence, these are the virtual links that are mapped to the congested physical link with life time longer than a predefined threshold \( T_l \). For each virtual link in \( \mathcal{CVL}(e^l_{uv}) \), we define a corresponding binary variable \( z^s_{ij} \), which takes value 1 if the virtual link is selected to remap, and 0 otherwise. The remapping cost of this link is denoted by a parameter \( C_{\text{mig}}^s_{ij} \), which depends on the service type of \( N.S_s \), the link length of the virtual link \( e^s_{ij} \), etc. For each congested physical link \( e^l_{uv} \), we formulate a remapping virtual link selection problem as:

\[
\begin{align*}
\min \quad & C_{\text{mig}}^s_{ij} z^s_{ij} \\
\text{s.t.} \quad & \sum_{e^s_{ij} \in \mathcal{CVL}(e^l_{uv})} B^s_{ij} z^s_{ij} \geq \sum_{s \in [1,S+1]} B^l_{uv} - B^l_{uv} \\
\end{align*}
\tag{P3}
\]

Problem (P3) is also a typical 0-1 Knapsack Problem which can be solved through existing algorithms.

5.5 Virtual Link Remapping

Then, we introduce the virtual link remapping process. There are two situations which lead to the remap of a virtual link: (1) a virtual node is migrated to a new physical node, consequently its connected virtual links should be remapped; (2) a virtual link should be remapped to avoid passing through a congested physical link. In the first situation, the virtual link should be mapped to a whole new physical path; in the second situation, only the congested part of the old physical path need to be remapped. Nevertheless, for both situations, the remapping process can be summarized as the following steps:
Suppose a virtual link with bandwidth requirement $B$ should be remapped to between physical nodes $n_u^l$ and $n_u^r$. Then, on the infrastructure graph $G_I$, we first remove all congested links, then we further remove links with bandwidth lower than $B$. On the reduced graph, we find the maximum residual capacity path between $n_u^l$ and $n_u^r$, to which the virtual link is mapped. Otherwise, a link remapping rejection is returned.

5.6 Reconfiguration Algorithm

Based on the above description, we could finally depict the whole reconfiguration process in Algorithm 2. The algorithm is triggered when a NSR is rejected. Firstly, a list of to-be-reconfigured physical nodes is calculated as $L(CN)$ which is sort in congestion level decreasing order. The algorithm contains two parts. In the first phase (step 5-19), we first treat physical nodes with computing resource deficits by migrating its hosted virtual nodes to other physical nodes. For a current congested physical node $n_u^l$, we solve Problem (P2) to obtain a list of to-be-migrated virtual nodes $MVN$ (step 6), and try to migrate these nodes one-by-one (step 7). For each virtual node, a candidate list of migration destination physical nodes is obtained (step 8), and the virtual node could finally migrate to one destination only if all its connected virtual links could be successfully remapped (step 10, 11). If all the candidate physical nodes cannot fulfill the link remapping requirement, the virtual node will remain on the original physical node (step 14,15). In this case, the computing resource deficit cannot be alleviated. After trying all virtual nodes in $MVN$, if the resource deficit still exist, we go back to step 6 to determine a new set of $MVN$ for migration, until the physical node is not congested. In case the resource deficit cannot be solved after all virtual nodes on $n_u^l$ are tried for migration, a rejection is returned such that the rejected NS $S_{S+1}$ cannot be re-deployed through the reconfiguration.

In the first phase (step 5-19) where we try to solve computing resource deficit for physical nodes in the set $(CN - BN)$, we do not treat link congestion, because virtual node migration leads to virtual link migration which could change the bandwidth state of physical links. After a physical node’s computing resource deficit is solved, we check whether the bandwidth resource deficit still exists, if so the physical node’s congestion level is recomputed and insert in to the $BN$ set, and it will be treat in the second phase of the algorithm (step 21-27).

In the second part of the algorithm, we treat physical nodes in the $BN$ set by reconfiguring its congested physical links. For each congested physical link, the algorithm first determine a set of to-be-remapped virtual links by solving Problem (P3) (step 23). Then, we try to remap these virtual links one-by-one by just remapping one-hop from the current congested physical node. After trying all virtual links in the $MVL$ set, if the bandwidth deficit still exists, we go back to step 23 to determine a new set of $MVL$ for remapping. If all the virtual links are tried for remapping and the deficit cannot be solved, a rejection is returned that $NS_{S+1}$ cannot be re-deployed.

6. Evaluations

6.1 Simulation Setting

We build a simulation platform to evaluate the performance of our proposed algorithm. Firstly, in the experiment, the underlying 5G physical network and NSR’s virtual network topology is generated by a modified Barabasi-Albert (BA) scale-free network model construction algorithm [36]. The BA model is largely utilized to represent the 5G system and NSR topology [16], [37]. The network is generated from an initially fully connected CN network. Then, new nodes are added into the network one-by-one. Each time, the incoming new node is connected towards existing nodes in the network based on a probability $P_i = \frac{\sum_j d_i}{d_i}$, where $N$ denotes the total number of current existing nodes and $d_i$ is the degree of the incoming node. Moreover, TN nodes can be connected to TN or CN nodes, whilst AN nodes can only be connected

---

**Algorithm 2 NSR Reconfiguration Algorithm**

1. **Input**: The physical network $G_I$, the mapped NSs $\{NS_1, \cdots , NS_S\}$, the rejected $NS_{S+1}$.
2. **Output**: The remapping from $NS_{S+1}$ to $G_I$.
3. **Obtain the CN, BN set and the sorted list $L(CN)$.
4. **For** congested physical node $n_u^l \in L(CN)$ **do**
5. **if** $n_u^l \notin BN$ **then**
6. **solve Problem (P2)** to obtain a list of migration virtual node $MVN$.
7. **for** Virtual node $n_v^s \in MVN$ **do**
8. **Obtain $CPF(n_v^s)$ based on (17).**
9. **for** $n_v^s \in CPF(n_v^s)$ **do**
10. **if** All virtual links of $n_v^s$ can be remapped **then**
11. **Migrate** $n_v^s$ to $n_u^l$, break the loop and update network states.
12. **end if**
13. **end for**
14. **if** $n_v^s$ cannot be migrated to any node in $CPF(n_v^s)$ **then**
15. **Do not migrate** $n_v^s$.
16. **end if**
17. **else**
18. **Check** if $n_v^s$’s computing deficit is solved. If not, go back to Step 6 to try to migrate a new set of virtual nodes with updated state. If all virtual nodes are tried, return a Rejection.
19. **Check** if $n_v^s$’s links bandwidth deficits are solved. If not, add $n_u^l$ to $BN$ and resort $L(CN)$.
20. **else**
21. **Obtain** the congested physical link set $CPF(n_u^l)$.
22. **for** Physical link $e_{ij}^s \in CPF(n_u^l)$ **do**
23. **solve Problem (P3)** to obtain a list of remapping virtual links $MVL$.
24. **for** Virtual link $e_{ij}^s \in MVL$ **do**
25. Remap one-hop from $n_v^s$ of $e_{ij}^s$’s mapped path.
26. **end for**
27. **Check** if $e_{ij}^s$’s bandwidth deficit is solved. If not, go back to Step 23 to try to remap a new set of virtual links with updated state. If all virtual links are tried, return a Rejection.
28. **end for**
29. **end if**
30. **end for**
to TN nodes.

We introduce the overall simulation settings. For the physical network, physical node CPU capacity follows uniform distribution on the interval [50, 100] and physical link capacity follows uniform distribution on the interval [50, 100]. For virtual network, virtual node CPU requirement follows uniform distribution on the interval [5, 10] and virtual link bandwidth requirement follows uniform distribution on the interval [5, 10]. Simulations with the same setting are performed for 100 times, and the results are taken as the average value.

We conducted two sets of simulations. In the first set, we evaluate the performance of the LBNE algorithm in a static environment (detailed in Sec. 6.2). Especially, we compared our algorithm towards two baseline algorithms in the literature. We denote the algorithm presented in [16] as the baseline-1 algorithm. This is a two-step 5G Core NS deployment algorithm which first places virtual nodes and then maps virtual links. When placing virtual nodes, the topological information of nodes are considered. We denote the algorithm presented in [38] as the baseline-2 algorithm. This is a one-step coordinated virtual node and link mapping algorithm based on subgraph isomorphism detection. In the second set of simulation, we further evaluate the performance of the reconfiguration algorithm in a dynamic environment (detailed in Sec. 6.3). For this dynamic experiment, we build an event-driven simulation platform to simulate a system with 20000 time units. For the dynamic system, NSR arrival follows Poisson Process with average arrival rate of 4 NSRs per 100 time units. The lifetime of each NSR is uniformly distributed between 800 to 1500 time units.

6.2 Deployment Performance in a Static Environment

We first discuss the results of the static simulation, in which we compare the LBNE algorithm towards the two baseline algorithms. Typically, we show the underlying physical resource utilization ratio by showing the acceptance ratio in Fig. 3 and Fig. 4. The acceptance ratio is calculated as the number of deployed NSRs over the number of overall NSRs. Then, we further investigate the workload distribution for both physical nodes and links by showing the Cumulative Distribution Function (CDF) of resource usage from Fig. 5 to Fig. 8.

6.2.1 Acceptance Ratio

In Fig. 3, we investigate the acceptance ratio with different physical network scales by varying physical network nodes number from 30 to 150. The number of NSRs is set to 30, with each NSR contains 10 virtual nodes. The result shows our LBNE algorithm outperforms the two baseline algorithms in achieving higher acceptance ratio. As the scale of the physical network increases, the acceptance ratio also increases because more physical resources are provided. In Fig. 4, we investigate the acceptance ratio with different number of NSRs by fixing the physical network to 100 physical nodes. The acceptance ratio decreases as the number of NSRs increases, because the physical network becomes more and more under provisioned. Our LBNE still achieves higher acceptance ratio.

6.2.2 Workloads of Physical Nodes and Links

The objective of the LBNE algorithm is to deploy NSs by balancing the workloads of the underlying physical nodes and links. Workload balancing is beneficial in avoiding resource fragmentation, hence it can improve resource utilization ratio by increasing the potential to accommodate future incoming NSRs.

Firstly, we show the Standard Deviation of physical node usage and link usage in Table 2. The physical node usage is calculated as the percentage of allocated CPUs over the node’s equipped CPUs. The physical link usage is calculated as the percentage of allocated bandwidth over the link’s bandwidth. The standard deviation measures the dispersion of a data set relative to its mean and is calculated as the square root of the variance. Hence, the standard deviation is frequently utilized as a performance index to reflect the level of workload balancing, and lower value indicates more balanced workloads. In Table 2, we vary the number of NSRs from 10 to 70, and these NSRs are deployed towards a physical network with 100 nodes. The results shows that our LBNE algorithm achieves more balanced workloads on both physical nodes and physical links, comparing to the other two baseline algorithms.

Further, to investigate the details of physical node usage and link usage distributions, we show the Cumulative Distribution Function (CDF) of physical node usage and link usage distributions from Fig. 5 to Fig. 8. In Fig. 5, we show the results with 10 NSRs for a physical network with 100 nodes. The figure demonstrates that our load-balancing oriented algorithm achieves more balanced load distribution on physical node usage. The linear increase for the curve of LBNE shows that the workloads distribution is evenly distributed. For baseline-1, there are more nodes with lighter workloads than our LBNE algorithm. For baseline-2, almost 60% of physical nodes are not used, and almost 40% nodes are intensively used (the sharp increase from usage ratio 0.8 to 1). In Fig. 6, we show the results with 70 NSRs for a physical network with 100 nodes. Baseline-1 shows similar results comparing to 10 NSRs, that is because the acceptance ratio of baseline-1 is the lowest, hence the node utilization ratio is the lowest. For baseline-2, node usage is improved as the workload distributes between usage ratio 0.3 to 1. The result still shows that our LBNE achieves more balanced node usage, as the workload distributes between usage ratio from 0.6 to 1.

Fig. 7 shows the workload distribution of physical links for 10 NSRs with a physical network with 100 nodes. Our algorithm also achieves more balanced link load distribution. For the baseline-1 algorithm, more than 80% of links are not used, hence the workloads are very unbalanced. For the baseline-2 algorithm, almost 40% of links are not used. For
our algorithm, about 10% of links are not used. Moreover, all link loads are distributed between usage ratio 0 and 0.3.

Fig. 8 shows the workload distribution of links for 70 NSRs with the same physical network scale. Our algorithm still achieves more balanced link load among the three. The baseline-1 algorithm performs similar as in Fig. 7, which show low utilization ratio and unbalanced link workload. Comparing our algorithm to the baseline-2 algorithm, the link usage of LBNE is more balanced, which lies between 0.2 and 0.8.

The above simulation results show that our LBNE algorithm performs better in using the physical network resource more efficiently, and balances the workloads of both physical nodes and links.

6.3 Reconfiguration Performance in a Dynamic Environment

In the second part of simulation, we compare our reconfiguration scheme (which is noted as LBNE-Reconf) towards the LBNE algorithm without reconfiguration (noted as LBNE-noReconf), the baseline-1 algorithm without reconfiguration and the baseline-2 without reconfiguration. The results are presented in Fig. 9 and Fig. 10. Fig. 9 shows the variation of acceptance ratio along simulation time for the four algorithms. With reconfiguration, the acceptance ratio is
Table 2  Statistics on physical nodes and links workloads

<table>
<thead>
<tr>
<th>NSRnum</th>
<th>10</th>
<th>20</th>
<th>30</th>
<th>40</th>
<th>50</th>
<th>60</th>
<th>70</th>
</tr>
</thead>
<tbody>
<tr>
<td>LBNE</td>
<td>0.208321</td>
<td>0.189625</td>
<td>0.0927186</td>
<td>0.0806441</td>
<td>0.0756057</td>
<td>0.0771249</td>
<td>0.0781882</td>
</tr>
<tr>
<td>baseline-1</td>
<td>0.219969</td>
<td>0.222448</td>
<td>0.235637</td>
<td>0.230606</td>
<td>0.238299</td>
<td>0.234295</td>
<td>0.236605</td>
</tr>
<tr>
<td>baseline-2</td>
<td>0.404883</td>
<td>0.302474</td>
<td>0.171676</td>
<td>0.143359</td>
<td>0.131018</td>
<td>0.128910</td>
<td>0.130173</td>
</tr>
</tbody>
</table>

Standard deviation $\sigma$ for physical links usages

<table>
<thead>
<tr>
<th>NSRnum</th>
<th>10</th>
<th>20</th>
<th>30</th>
<th>40</th>
<th>50</th>
<th>60</th>
<th>70</th>
</tr>
</thead>
<tbody>
<tr>
<td>LBNE</td>
<td>0.082344</td>
<td>0.143795</td>
<td>0.154261</td>
<td>0.15593</td>
<td>0.157115</td>
<td>0.158181</td>
<td>0.158658</td>
</tr>
<tr>
<td>baseline-1</td>
<td>0.190653</td>
<td>0.194067</td>
<td>0.204623</td>
<td>0.202896</td>
<td>0.207851</td>
<td>0.210383</td>
<td>0.207965</td>
</tr>
<tr>
<td>baseline-2</td>
<td>0.219362</td>
<td>0.272490</td>
<td>0.269601</td>
<td>0.272897</td>
<td>0.269371</td>
<td>0.271510</td>
<td>0.270378</td>
</tr>
</tbody>
</table>

improved, as more NSRs can be successfully deployed. As the system runs, the acceptance ratio of the four algorithms converges into stable state. In Fig. 10, we show the average CPU usage ratio of all physical nodes along the simulation time for the four algorithms. The usage ratio varies, but reconfiguration achieves better physical resource utilization ratio.

7. Conclusions

In this work, we investigate the load balancing oriented deployment and reconfiguration of 5G Core network slices. For NS deployment, our goal is achieving balanced workloads of the underlying physical network in embedding the VN of the NS. We formulate a load-balancing oriented 5G E2E NS deployment problem through an Integer Linear Program (ILP) formulation. Due to the complexity of the problem, we design a LBNE algorithm to effectively deploy NSs. We especially pay attention to also balance the workloads of physical links by utilizing the maximum residual capacity paths in the process of link mapping. Then, for a rejected NSR, we propose a reactive reconfiguration strategy to accommodate the rejected NSR. By rerun the LBNE algorithm with slacked physical resources, the congested part of the physical network is identified and corresponding resources are reorganized to re-balance the network. Through the simulation, we demonstrate the high performance of our algorithms regarding load balancing and acceptance ratio.

References

[19] A. Jarar and A. Karmouch, “Decomposition approaches for vir-


Xin LU received his Master’s degree from Beijing University of Aeronautics and Astronautics (BUAA). He is a Senior Engineer of Power Line Communication Application Technology Laboratory of State Grid. His research interest focuses on power line communication, wireless communication, electrical information collection systems, and Internet of Things.
Xiang WANG received his Master’s degree from Beijing Institute of Technology (BIT). He is Director of Power Line Communication Application Technology Laboratory of State Grid. His research focuses on chip technology, power information and communication technology.

Lin PANG received his Master’s degree from Changchun University of Technology. He is Engineer of Power Line Communication Application Technology Laboratory of State Grid. His research focuses on power information and communication technology standardization.

Jiayi LIU obtained her Bachelors of Science degree in Electronic Engineering from Xidian University in Xi’an China, in 2007. She received her Master of Science and Ph.D., both in Computer Science, from Telecom-Bretagne and Rennes 1 University, France, in 2009 and 2013, respectively. Since 2014, she works as Lecturer with Xidian University, in School of Telecommunication Engineering and Guangzhou Institute of Technology. Her research interests include 5G network, content distribution, Network resource scheduling.

Qinghai YANG received the B.S. degree in communication engineering from the Shan-dong University of Technology, China, in 1998, the M.S. degree in information and communication systems from Xidian University, China, in 2001, and the Ph.D. degree in communication engineering from Inha University, South Korea, in 2007, with the University-President Award. From 2007 to 2008, he was a Research Fellow with UWB-ITRC, South Korea. Since 2008, he has been a Full Professor with Xidian University. His current research interest lies in the fields of autonomic communication, content delivery networks, and LTE-A techniques.

Xingchen SONG received the B.S. degree in information engineering from Xidian University, Xi’an, China, in 2018, where he is currently pursuing the M.S. degree in communication and information system. His research interests include 5G network and network slicing.