PAPER

Reliability Enhancement for 5G End-to-End Network Slice Provisioning to Survive Physical Node Failures

Xiang WANG†, **Xin LU**†, **Meiming FU**†, ***Nonmembers***, **Jiayi LIU**††a), ***Member***, ***and*** **Hongyan YANG**††, ***Nonmember***

SUMMARY Leveraging on Network Function Virtualization (NFV) and Software Defined Networking (SDN), network slicing (NS) is recognized as a key technology that enables the 5G Infrastructure Provider (InP) to support diversified vertical services over a shared common physical infrastructure. 5G end-to-end (E2E) NS is a logical virtual network that spans across the 5G network. Existing works on improving the reliability of the 5G mainly focus on reliable wireless communications, on the other hand, the reliability of an NS also refers to the ability of the NS system to provide continued service. Hence, in this work, we focus on enhancing the reliability of the NS to cope with physical network node failures, and we investigate the NS deployment problem to improve the reliability of the system represented by the NS. The reliability of an NS is enhanced by two means: firstly, by considering the topology information of an NS, critical virtual nodes are backed up to allow failure recovery; secondly, the embedding of the augmented NS virtual network is optimized for failure avoidance. We formulate the embedding of the augmented virtual network (AVN) to maximize the survivability of the NS system as the survivable AVN embedding (S-AVNE) problem through an Integer Linear Program (ILP) formulation. Due to the complexity of the problem, a heuristic algorithm is introduced. Finally, we conduct intensive simulations to evaluate the performance of our algorithm with regard to improving the reliability of the NS system.

key words: *5G, network slicing, E2E network slice, reliability, virtual network embedding*

1. Introduction

Leveraging on new maturing paradigms, such as Network Function Virtualization (NFV) and Software Defined Networking (SDN) [1], network slicing (NS) empowers the 5G network to compose multiple separated end-to-end (E2E) logical networks over a shared common physical infrastructure to support multiple diversified vertical applications with efficiency and flexibility [2], [3]. Typically, the 3GPP defined three types of 5G use cases: enhanced Mobile Broad Band (eMBB), massive Machine Type Communications (mMTC), and Ultra Reliable Low Latency Communications (URLLC). eMBB is recognized as providing greater data-bandwidth. mMTC is designed to connect a large number of Internet of Things (IoT) devices. URLLC is expected to support applications such as industrial automation and automatic driving, where reliability and low latency are deemed to be of great importance.

Existing works on URLLC mainly focus on the wireless access domain to improve parameters such as packet delivery ratio, bit error rate, signal-to-interference-plus-noise ratio, and outage probability which represent reliable wireless communication [4]. On the other hand, reliability also refers to the ability that a system or network functions could provide services under various network conditions (such as physical network failures). Through the network slicing point of view, the reliability of an NS is also represented by the availability and functionality of the virtualized network functions (VNFs) that compose the corresponding NS to ensure the NS's service continuity. Hence, in the current work, we investigate the NS deployment problem to improve the reliability of the system represented by the NS.

The deployment of a 5G NS includes the initialization of the slice instance that contains all the required VNFs to compose the NS's service across the whole 5G system [5], [6]. Hence, the realization of such a 5G NS is essentially the deployment of the corresponding virtual network (VN) which involves the optimization of resource allocations by considering network and computing resources. The mapping from the slice instance VN towards the physical infrastructure is a typical virtual network embedding (VNE) problem and has been intensively studied in the literature [7]. The VNE problem is further extended as Survivable virtual network embedding (SVNE) problem [8] which aims at embedding the VN in a way such that the VN is still operating with physical network failures, which is a critical problem for end-users and service providers.

The reliability of a VN can be enhanced through two ways: (1) failure avoidance tries to optimize the resource allocation for the embedding of the VN such that failures affect as little as possible towards the VN's system; (2) failure protection tries to provide backup resources such that the VN can recover quickly after being affected by failures. In this current work, we focus on network node failures. Firstly, by fully considering the topology information of the VN, the important virtual nodes are backed up. The topology information can reflect the importance of virtual nodes. For example, edge node failure only affects a certain region, however core node failure can lead to the down of the whole system. Hence, we augment the original VN by the backed up virtual nodes and try to embed the augmented VN (AVN). We further formulate the embedding of the AVN to maximize the survivability of the VN system as the survivable AVN embedding (S-AVNE) problem through Integer Linear

Manuscript received January 26, 2021.
Manuscript revised April 12, 2021.
Manuscript publicized June 1, 2021.

†The authors are with State Grid Laboratory of Power Line Communication Application Technology, Shenzhen Guodian Technology Communication, Co., Ltd, China.
††The authors are with State Key Laboratory on ISN, Xidian University, China.
a) E-mail: jyliu@xidian.edu.cn

DOI: 10.1587/transcom.2021EBP3013

Program (ILP) formulation. Due to the complexity of the problem, a heuristic algorithm is introduced. We conduct intensive simulations and compare the S-AVNE algorithm towards an existing baseline algorithm. Simulation results demonstrate that our mechanism is more practical, and that it achieves higher reliability with less resource consumption.

Related works are summarized in Sect. 2. Then, Sect. 3 describes the system model and presents the S-AVNE problem formulation, with the algorithm presented in Sect. 4. In Sect. 5 we evaluate the performance gain of our proposals. Finally, we conclude this work in Sect. 6.

2. Related Works

Reliability and URLLC are novel topics that are emerging with the development of the 5G network. So far, the research related in this domain is still in the initial phase and attracting lots of attentions. Variety of existing works try to improve the reliability of the wireless transmission by improving parameters such as packet delivery ratio (PDR), bit error rate (BER), signal-to-interference-plus-noise ratio (SINR), and outage probability, which improve the reliability of wireless communications [4], [9], [10]. However, the end-to-end service reliability refers to multiple related domains. For instance, in [11], the authors analyze the space domain wireless access availability by considering various network topologies and user distributions. In this current work, we mainly focus on the reliability that the 5G NS system in providing continued service under network failures.

2.1 5G Network Slicing and VNE Problem

Actually, the idea of virtualizing the network for adopting new network architecture and applications has been proposed for a long time [12]. Based on the idea of network virtualization, the NS technique is currently gaining extreme research interests in the development of the 5G network. It has been also the focus of multiple standardization organizations, for example, the logical E2E slicing has been included in the 3GPP Release 15 as one of the fundamental technologies for the 5G network. 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 [3]; (2) E2E slicing (or CN slicing) refers to the embedding of the verticals' virtual networks towards the physical infrastructure [13], [14]. The E2E slicing stands for an important part for 5G NS, in the current paper, we focus on E2E 5G NS deployment problem.

The essential problem for 5G E2E NS implementation is the embedding of the NS corresponding virtual network, which is composed of VNFs to form the vertical's serving system, towards the physical infrastructure. The VN embedding is essentially a VNE problem, which has been intensively studied in the literature [7]. The VNE problem can be formulated as an ILP model, and the complexity of the problem is known as NP-hard [7]. In [15], the authors propose an exact mathematical model based on layered graph and column generation for the embedding of SFCs. In [16], the authors present an approach to accelerate VNE algorithms by reducing the searching space and extracting the VN into subgraphs.

There are also a large amount of efforts devoted in designing heuristic algorithms to achieve a practical solution with efficiency. Some works adopt the two-step embedding method in which the virtual nodes are first placed, then based on the position of virtual nodes, virtual links are mapped as Multi-Commodity Flow problem [17], [18]. This method neglects the relation between network nodes and links, which may result in less efficient usage of physical resources. Another class of methods try to place virtual nodes and links coordinately within one stage. 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 on obtaining a better physical resource utilization. In [11], a Breath First Spanning Tree (BFS-Tree) based VNE embedding algorithm is proposed to facilitate the coordinated one-step node and link mapping process. By constructing the BFS-Tree from the VN, a sorted list of virtual nodes is determined, based on which the virtual nodes are mapped. This method seeks to map neighboring virtual nodes into near physical locations to reduce link mapping bandwidth consumption. As a well-known one-step VNE algorithm, it is further modified and utilized in multiple different scenarios for VN embedding [18], [20], [21]. We adapt this BFS-Tree based VNE embedding algorithm to embed our augmented virtual network, the detailed algorithm is discussed in Sect. 4.

2.2 Survivable VNE Problem

Survivable virtual network embedding (SVNE) aims at optimizing the embedding of a VN such that: (1) the reliability of the VN is optimized and (2) after being affected by physical network failures, the VN can still operate. The first one is achieved by failure avoidance through optimizing the resource allocation in a way that failures affect as little as possible, and the second one is achieved by failure recovery by providing backup resources.

Related works on survivable VNE resource allocation mainly focus on either physical link failures or physical node failures. We then outline the two groups of works. For link failures, in [8], the authors studied the SVNE problem by considering single substrate link failures in VNE. They added survivability mechanisms to the link mapping phase by utilizing a pre-reserved quota for backup on each physical link and modeled multiple re-routing strategy. In [22], the authors studied the SVNE problem for multi-path links embedding to provide maximal survivability to the VNs in case of multiple link failures. The problem is formulated as an ILP, and greedy proactive and reactive algorithms are proposed. In [23], the authors improve the reliability of the VN by augmenting the VN with backup capacity and studied the embedding problem of the resulting augmented VN accordingly. They jointly optimized backup capacity allocation and embedding for a VN for single substrate link failure.

In [24], the authors studied the SVNE problem in virtualized Fiber-Wireless (FiWi) access network for robust IoT service provisioning in case of link failures. The SVNE problem in FiWi is formulated through ILP to minimize physical resource cost and guarantee the availability requirements of the IoT VNs.

For physical node failures, in [25], a SVNE problem is studied to recover from physical node failures by backing up redundant physical nodes towards the critical nodes. They modeled the problem as an MILP and proposed heuristic algorithms. However, the determination of the critical nodes are not depicted. In [26], a SVNE problem is investigated to enhance the survivability of the VN against multiple node failures. The topology attributes are considered to provide each physical node with multiple candidate switching nodes, moreover a recoverability-based VN embedding and a remapping algorithm are presented. In [27], the authors try to map VNs to more reliable physical nodes to minimize the failure recovery cost. They model the corresponding VNE problem as a classical Steiner Minimal Tree (SMT) problem and propose SMT-based heuristics. However, they assume that the underlying physical links are with infinite bandwidth which is not realistic.

Our work deals with physical node failures. We improve the reliability of the E2E NS by two means: (1) by considering the topology information of the VN, virtual nodes with higher importance is backup to enhance the reliability of the VN system; (2) we model the resource allocation problem for embedding the augmented VN to further enhance the reliability of the VN.

In the literature, a number of works evaluate the topological importance of a network nodes. In [11], [28], network nodes are ranked by resource and topological attributes based on the Markov Random Walk model. In [18], the authors simply sum up the values of degree and betweenness centrality to represent the topological importance of a network node. In [20], topology importance is calculated from node degree and clustering coefficient. They utilize the function $u(x) = \frac{x}{\sqrt{x^2}}$ to combine the two values. In [29], only the node degree is utilized to measure the node's importance value. In [30], the reciprocal of the sum of the number of hops of a given node towards other nodes is utilized to represent the importance of the node. In this work, we comprehensively consider multiple topology attributes to measure node importance. The node degree and cluster coefficient reflect a network node's local importance, whilst betweeness centrality is utilized to show the node's global importance. Further, we utilize the TOPSIS method to comprehensively combine these factors.

2.3 Reliability-Oriented 5G E2E NS

There are also works studying the reliability oriented 5G network slicing problem through dedicated protection. In [31], the authors outline the resource allocation models for embedding 5G NSs in multiple scenarios: the nominal case, NS with uncertain traffic demands and survivable NS under link/node failures. They utilized a dedicated protection scheme in which each virtual node and link is duplicated. Thus, the resource consumption is scaled by a factor of 2. Then, this factor is reduced by mapping virtual link into disjoint multiple physical paths to cope with link failure. In the context of E2E 5G network slice embedding, the authors in [32] propose mechanisms to improve service reliability by adopting 5G network functions split based on path protection to re-establish paths after failures for uRLLC services. In [33], the reliability of 5G transport network slices in elastic optical networks (EON) is enhanced while providing dedicated protection for physical link failures. The backup resource is reduced by multi-path provisioning and providing reduced protection bandwidth. However, the above works mainly focus on link failures through mapping each link into multiple physical links and providing backup bandwidth resources. In this current work, we mainly focus on providing backup resources for virtual nodes. Moreover, in [32] the whole SFC is duplicated, we provide backup resources for the important virtual nodes by considering the topology information of the NS.

3. System Model

3.1 Physical Network Model

The physical infrastructure for 5G E2E 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 for the physical network nodes and E_I stands for physical communication links. One physical network node is denoted by $n_u^I \in N_I$. We distinguish three kinds of physical nodes: (1) the access/edge nodes (AN) set N_I^A, such as virtual Remote Radio Unit (RRU) and edge Data Centers (DCs); (2) the transport network (TN) nodes set N_I^T which denotes TN optical switches; (3) the core network (CN) nodes set N_I^C, which includes core DCs for CN data plane and control plane VNFs virtualization (as shown in Fig. 1). Hence, we have $N_I = N_I^A \cup N_I^T \cup N_I^C$. Moreover, each physical node $n_u^I \in N_I^A \cup N_I^C$ is characterized by its capacity C_u^I which represents the available computing resources of the node.

Physical communication links are represented by E_I, with $e_{uv}^I \in E_I$ denotes the physical link connecting two physical nodes n_u^I and n_v^I. For instance, virtual RRUs are connected to the edge centers via Common Public Radio Interface (CPRI) fronthaul transport, and the other types of links (including backhaul and TN CN links) are based on IP transportation. For each link $e_{uv}^I \in E_I$, we denote by B_{uv}^I the available bandwidth of the link. L_{uv}^I denotes the transmission delay of the link.

The physical network is vulnerable to failures, we denote by P_u the failure probability of physical node n_u^I and P_{uv} the failure probability of physical link e_{uv}^I. For the ease of understanding, the notations are summarized in Table 1.

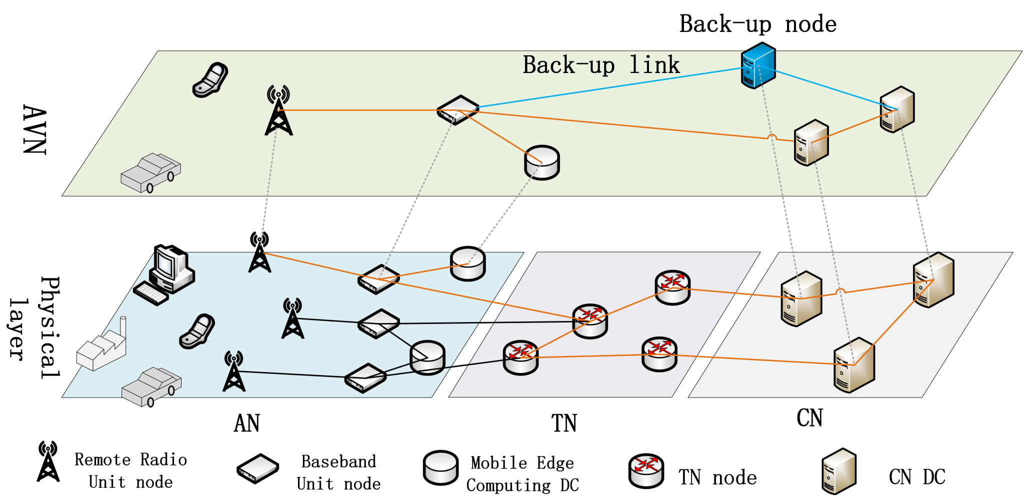

Fig. 1 The embedding of the augmented virtual network of 5G NS.

Table 1 Notations.

G_I, G_O, G_S	The physical/virtual/augmented network.
$N_I, n_u^I \in N_I$	Physical nodes set, one physical node.
C_u^I	Computing resources of the node.
$E_I, e_{uv}^I \in E_I$	Physical links set, one physical link.
B_{uv}^I, L_{uv}^I	Bandwidth and delay of e_{uv}^I.
P_u, P_{uv}	Failure probability of node n_u^I and link e_{uv}^I.
$N_O, n_i^O \in N_O$	Virtual nodes set, one original virtual node.
C_i^O	Computing resources requirement of n_i^O.
$E_O, e_{ij}^O \in E_O$	Virtual links set, one original virtual link.
B_{ij}^O, L_{ij}^O	Bandwidth and delay requirements of e_{ij}^O.
$N_A, n_i^A \in N_A$	Augmented virtual nodes, backup node of n_i^O.
C_i^A	Computing resources requirement of n_i^A.
$E_A, e_{ij}^A \in E_A$	Augmented links set, added virtual link.
B_{ij}^A, L_{ij}^A	Bandwidth and delay requirements of e_{ij}^A.
$L(i), G(i), S(i)$	Local/global/overall importance of node n_i^O.
x_{iu}^O	Binary variable of mapping n_i^O to n_u^I.
x_{iu}^A	Binary variable of mapping n_i^A to n_u^I.
y_{ijuv}^O	Binary variable of mapping e_{ij}^O on e_{uv}^I.
y_{ijuv}^A	Binary variable of mapping e_{ij}^A on e_{uv}^I.

3.2 Network Slice and Virtual Network Model

The network slice VN is modeled as a graph $G_O(N_O, E_O)$, where N_O denotes for the virtual network nodes and E_O stands for virtual links connecting the virtual nodes to form the VN. One virtual network node is denoted by $n_i^O \in N_O$. Each virtual network node represents a virtual VNF instance, such as virtual RRU, virtual Mobile Edge Computing (MEC) server, virtual video encoder, virtual Mobility Management Entity (MME), and virtual gateway, etc. We denote by C_i^O the physical resource requirement of the virtual node n_i^O. A virtual link is represented by $e_{ij}^O \in E_O$, with B_{ij}^O denoting its bandwidth requirement, and L_{ij}^O the link delay requirement.

3.3 Augmented Virtual Network Model

3.3.1 Virtual Node Importance

To cope with physical node failures, we utilize the dedicated protection for virtual nodes by considering the topology information of the VN: the virtual nodes with higher topology importance are duplicated with backup nodes. Hence, we first measure the importance of virtual nodes. To sum up, the difference between our work and other topology importance measurement methods are as follows: (1) both node local importance and global importance are considered through different graph concepts; (2) we utilize the TOPSIS method ([34]) to calculate the importance value of a node with multiple attributes.

We first introduce the mechanism to measure the *local importance* (denoted by $L(i)$) and *global importance* (denoted by $G(i)$) of a virtual node n_i^O. We utilize node degree and clustering coefficient to measure the local importance of a node in the VN's topology. $L(i)$ is defined as follows:

$$L(i) = \frac{d(i)}{\sqrt{\sum_{j=1}^{|N_O|} d(j)^2}} + \frac{c(i)}{\sqrt{\sum_{j=1}^{|N_O|} c(j)^2}} \tag{1}$$

where $d(i)$ is the *degree* of virtual node n_i^O in the VN, and $c(i)$ is the *clustering coefficient* of n_i^O in the VN. $|N_O|$ is the number of nodes in the VN. Hence, $L(i)$ is calculated by adding the two corresponding elements from the normalized node degree vector and the normalized clustering coefficient vector.

Then, we utilize the *betweenness centrality* to measure the global importance of a virtual node. The betweenness centrality calculates the fraction of all-pairs shortest paths pass through the node, which is defined as $b(i) = \frac{\sum_{j \neq i \neq k} p_{jk}(i)}{\sum_{j \neq i \neq k} p_{jk}}$. p_{jk} is the number of shortest (n_j^O, n_k^O)-paths, and $p_{jk}(i)$ is the number of those pathes goes through node n_i^O. Then, the global importance of node n_i^O is given by:

$$G(i) = \frac{b(i) - \min\{b(j)|n_j^O \in N_O\}}{\max\{b(j)|n_j^O \in N_O\} - \min\{b(j)|n_j^O \in N_O\}} \tag{2}$$

where $\max\{b(j)|n_j^O \in N_O\}$ and $\min\{b(j)|n_j^O \in N_O\}$ represent the maximum and minimum $b(i)$ values for all virtual nodes. Hence, $G(i)$ is calculated as the min-max normalization of the $b(i)$ value.

Finally, we utilize the *technique for order preference by similarity to an ideal solution, TOPSIS* method to describe the importance of a virtual node with multiple attributes. We construct a matrix A such that each row i of the matrix is composed as: $A(i) = [L(i), G(i)]$. Hence, the matrix A has $|N_O|$ rows and 2 columns, where $|N_O|$ is the number of virtual nodes. Then, the importance of node n_i^O is given by:

$$S(i) = \frac{A_i^-}{A_i^+ + A_i^-} \tag{3}$$

where

$$A_i^+ = \sqrt{\sum_{f=1}^{2} (A_{if} - A_f^{max})^2} \tag{4}$$

and

$$A_i^- = \sqrt{\sum_{f=1}^{2} (A_{if} - A_f^{min})^2} \tag{5}$$

A_{if} represents the element of the row i and column f in A, A_f^{min} is the minimum value of elements in column f of A, and A_f^{max} is the maximum value of elements in column f of A. The range of the value of $S(i)$ is between 0 and 1.

3.3.2 Augmented Virtual Network

Based on the measurement of virtual node importance $S(i)$, we are capable to determine the augmented VN where important virtual nodes are backed-up. For a given importance threshold S_T, the virtual nodes with $S(i) > S_T$ are backed up. Then, we define the augmented virtual network (AVN) as follows:

- The AVN is modeled by an undirected graph $G_S(N_S, E_S)$, where N_S represents the VNFs set and E_S represents the virtual links set.
- The VNFs set $N_S = N_O \cup N_A$, where N_O represents VNFs of the original VN and N_A represents the backup virtual nodes set. For ease of notation, $n_i^A \in N_A$ represents the backup node of the original virtual node $n_i^O \in N_O$ with capacity requirement C_i^A.
- The virtual link set $E_S = E_O \cup E_A$, where E_O represents the original VN's virtual links and E_A represents the added virtual links connecting the backup virtual nodes. Specifically, $E_{A1} = \{e_{ii}^A | \forall n_i^O \in N_O, n_i^A \in N_A\}$ and $E_{A2} = \{e_{ij}^A | \forall n_j^O \in \delta(n_i^O), n_i^A \in N_A\}$. E_{A1} connects each backup virtual node to its original virtual node, the resource requirement of these links are the data rates of necessary information transmission between the original node and the backup node to allow a fast migration in case of physical node failure. Then, E_{A2} connects each backup virtual node to its original virtual node's neighboring nodes, where $\delta(n_i^O)$ denotes the neighboring nodes of n_i^O in G_O. The resource requirement of these links are also the backup information transmission rate. Finally, $E_A = E_{A1} \cup E_{A2}$, and each virtual link in E_A requires bandwidth B_{ij}^A, and transmission delay L_{ij}^A.

3.4 Reliability-Oriented AVN Embedding Problem Formulation

3.4.1 Variables

To enhance the reliability of the NS, we embed the AVN towards the physical network to maximize the realiability of the whole AVN, as shown in Fig. 1. Hence, we first introduce the following binary variables:

- Two node mapping variables x_{iu}^O and x_{iu}^A, which take value 1 if the corresponding original n_i^O / backup n_i^A node is mapped to physical node n_u^I, and 0 otherwise.
- Two link mapping variables y_{ijuv}^O and y_{ijuv}^A, which take value 1 if the corresponding original / backup link is mapped to the physical link, and 0 otherwise.

3.4.2 Constraints

Then, we consider the following AVN embedding constraints. The first set of constraints (6) to (8) are node mapping constraints. Constraint (6) and (7) enforce that every original/augmented virtual node is mapped. Constraint (8) makes sure that every physical node can only accommodate one original or augmented virtual node to further enhance the reliability of the whole NS.

$$\sum_{n_u^I \in N_I} x_{iu}^O = 1, \quad \forall n_i^O \in N_O. \tag{6}$$

$$\sum_{n_u^I \in N_I} x_{iu}^A = 1, \quad \forall n_i^A \in N_A. \tag{7}$$

$$\sum_{n_i^O \in N_O} x_{iu}^O + \sum_{n_i^A \in N_A} x_{iu}^A = 1, \quad \forall n_u^I \in N_I. \tag{8}$$

We then consider the 5G network topology related constraints. In the 5G network, VNFs are position specific. For example, the virtual RRU can only be deployed on physical RRUs to provide wireless transmission in a certain area. In MEC service scenarios, the related VNFs should be placed on the edge of the network. To describe such location specification, we define a binary location parameter loc_i^u which takes value 1 if the corresponding virtual node n_i^O can be placed on the corresponding physical node n_u^I, and 0 otherwise. Hence, we introduce a location constraint in (9) and (10).

$$x_{iu}^O \leq loc_i^u, \quad \forall\, n_u^I \in N_I,\ n_i^O \in N_O. \tag{9}$$

$$x_{iu}^A \leq loc_i^u, \quad \forall\, n_u^I \in N_I,\ n_i^A \in N_A. \tag{10}$$

Then, the second set of constraints (11) and (12) are about the capacity restrictions of the physical nodes and links.

$$\sum_{n_i^O \in N_O} x_{iu}^O C_i^O + \sum_{n_i^A \in N_A} x_{iu}^A C_i^A \leq C_u^I, \quad \forall n_u^I \in N_I. \tag{11}$$

$$\sum_{e_{ij}^O \in E_O} y_{ijuv}^O B_{ij}^O + \sum_{e_{ij}^A \in E_A} y_{ijuv}^A B_{ij}^A \leq B_{uv}^I, \quad \forall e_{uv}^I \in E_I. \tag{12}$$

Finally, we consider the link mapping constraints (13) to (18). Constraint (13) is the loop free constraint. Constraint (14) is the flow reservation constraint for the original virtual links. Constraint (15) is the flow reservation constraint for the added virtual links from the backup virtual node to its original virtual node and its neighboring virtual nodes. Constraint (16) and (17) imply the link delay requirements of the virtual links. Finally, constraint (18) ensures that original virtual link and its backup links are mapped towards different physical links to further enhance the reliability of the NS.

$$\sum_{e_{ij}^O \in E_O} y_{ijuv}^O + \sum_{e_{ij}^A \in E_A} y_{ijuv}^A \leq 1, \quad \forall n_u^I \in N_I, n_v^I \in \delta(n_u^I). \tag{13}$$

$$\sum_{n_v^I \in N_I} (y_{ijuv}^O - y_{ijvu}^O) = x_{iv}^O - x_{ju}^O, \quad \forall e_{ij}^O \in E_O, n_u^I \in N_I. \tag{14}$$

$$\sum_{n_v^I \in N_I} (y_{ijuv}^A - y_{ijvu}^A) = x_{iv}^A - x_{ju}^O, \quad \forall e_{ij}^O \in E_O, n_u^I \in N_I. \tag{15}$$

$$\sum_{e_{uv}^I \in E_I} y_{ijuv}^O L_{uv}^I \leq L_{ij}^O, \quad \forall e_{ij}^O \in E_O. \tag{16}$$

$$\sum_{e_{uv}^I \in E_I} y_{ijuv}^A L_{uv}^I \leq L_{ij}^A, \quad \forall e_{ij}^A \in E_A. \tag{17}$$

$$y_{ijuv}^O + y_{ijuv}^A \leq 1, \quad \forall e_{ij}^O \in E_O, e_{uv}^I \in E_I. \tag{18}$$

3.4.3 Objective

Then, we derive the objective of the AVN embedding problem. The aim of the InP is mapping the AVN towards physical network with higher reliability. We first introduce two variables as follows:

$$x_u = \begin{cases} 1 & \text{if } \sum_{n_i} (x_{iu}^O + x_{iu}^A) \geq 1 \\ 0 & \text{otherwise} \end{cases} \tag{19}$$

$$y_{uv} = \begin{cases} 1 & \text{if } \sum_{e_{ij}} (y_{ijuv}^O + y_{ijuv}^A) \geq 1 \\ 0 & \text{otherwise} \end{cases} \tag{20}$$

Thus, when a physical node is mapped with virtual nodes x_u is 1, and when a physical link is mapped with virtual links y_{uv} is 1. Then, the survivability of the embedded NS system can be calculated as:

$$\prod_{n_u^I \in N_I} (1 - P_u x_u) \cdot \prod_{e_{uv} \in E_I} (1 - P_{uv} y_{uv}) \tag{21}$$

We then further apply the logarithm function to Eq. (21):

$$\sum_{n_u^I \in N_I} log(1 - P_u x_u) + \sum_{e_{uv} \in E_I} log(1 - P_{uv} y_{uv}). \tag{22}$$

Because x_u is a binary variable, which only takes two possible values: 1 and 0. In this case, we could further derive the above Equation as follows:

- In case $x_u = 0$, then $log(1 - P_u x_u) = log(1) = 0 = log(1 - P_u)x_u$.
- In case $x_u = 1$, then $log(1 - P_u x_u) = log(1 - P_u) = log(1 - P_u)x_u$.

Hence, we finally linearize Eq. (21) as:

$$\sum_{n_u^I \in N_I} log(1 - P_u)x_u + \sum_{e_{uv} \in E_I} log(1 - P_{uv})y_{uv}. \tag{23}$$

Finally, maximizing the surviving probability of the NS corresponds to the following Survivable Augmented Virtual Network Embedding (S-AVNE) problem formulated through ILP:

$$\max \sum_{n_u^I \in N_I} log(1 - P_u)x_u + \sum_{e_{uv} \in E_I} log(1 - P_{uv})y_{uv} \tag{S-AVNE}$$

$$\text{s.t.} \quad \text{Constraints (6)–(18).}$$

4. Heuristic Algorithm

The complexity of the above S-AVEN ILP model is NP-hard. Hence, we adopt a BFS-Tree based coordinated mapping algorithm to embed the augmented virtual network of the NS towards the physical network [11]. The basic idea of constructing a BFS-Tree on the VN is to form a sorted list of virtual nodes. The list is utilized to determine the order of virtual node mapping. The algorithm first maps the root of the tree. Then, the 1-hop neighbors of the root are mapped. Since the root is already mapped, when mapping its 1-hop neighbors, the corresponding virtual links towards the already mapped nodes can be determined at the same time. Then the algorithm iterates to map the 2-hops neighbors, and so on. This method seeks to map neighboring virtual nodes onto near physical locations to reduce link mapping bandwidth consumption.

To apply this BFS-Tree based mapping algorithm, we further adapt the algorithm in the following aspects:

- Firstly, the BFS-Tree is constructed from the original VN based on the importance value of virtual nodes. Then, when constructing the sorted list of virtual nodes, the backup nodes are inserted into the position just behind its original virtual nodes. Hence the backup nodes are mapped just after the mapping of the original nodes.
- The criteria for selecting the candidate physical nodes are modified: (1) we distinguish between original nodes and backup nodes, the backup nodes cannot be mapped towards the same physical nodes of the original nodes. (2) physical nodes are selected based on their failure probability to enhance the reliability of the VN.

The detailed description of the algorithm is depicted in the following sections.

4.1 Virtual Node Sorting Algorithm

Algorithm 1 Virtual node sorting algorithm

Require: The original virtual network G_O
Ensure: The sorted virtual node list $L(N_S)$ and tree T
for $N_i^O \in N_O$ **do**
    Calculate $S(i)$ of n_i^O based on Eq. (3).
end for
Find the node $\hat{n}_i^O$ with the highest $S(i)$, add $\hat{n}_i^O$ to $L(N_S)$.
Taking $\hat{n}_i^O$ as the root node, using BFS to generate tree T from G_O.
for Each layer of the tree T **do**
    Sort nodes in $S(i)$ descending order, and get list C.
    for $n_i^O \in C$ **do**
        Add n_i^O to the end of $L(N_S)$.
        if $S(i) > S_T$ **then**
            Add n_i^A to the end of $L(N_S)$.
        end if
    end for
end for

Algorithm 1 sorts a list of virtual nodes based on the importance of the nodes. This list determines the order of mapping of the virtual nodes. This list is established from building a spanning tree of the VN. The algorithm first finds the virtual node with the highest criticality $S(i)$ value as the tree root, which is denoted by $\hat{n}_i^O$. Then, it traverses the entire original virtual network N_O according to the breadth first traversal algorithm to generate the spanning tree T. Then, each layer of the tree is sorted in descending order according to $S(i)$ value. If the original node has a corresponding backup node, the original node and the backup node are added to the sorted list. If the node has no backup node, only the node needs to be added to the queue.

The basic idea of the algorithm is to firstly deploy the virtual node with the maximum $S(i)$ value and its backup node (if any). Then, nodes of the tree T are mapped layer by layer. On one hand, by utilizing the BFS-Tree, virtual links are mapped coordinately with the virtual nodes, and link mapping bandwidth consumption can be reduced. On the other hand, for every layer of the tree, virtual nodes are sorted based on their importance values such that more important nodes will be mapped firstly towards physical nodes with lower failure probability. Hence, the reliability of the NS can be improved.

4.2 Candidate Physical Node Set Selection Algorithm

Algorithm 2 Candidate physical node set selection algorithm

Require: Virtual node n_i^S, tree T return by Algorithm 1.
Ensure: Candidate physical node ordered set $Q(n_i^S)$
if $n_i^S = \hat{n}_i^S$ **then**
    $Q(n_i^S) = \{n_u^I \mid C_u^I > C_i^S\} \bigcap Loc(n_i^S)$.
else
    if n_i^S is an original virtual node n_i^O **then**
        Find the parent node n_p^O of n_i^O in T.
        Find the physical node n_p^I mapped by n_p^O.
        Add n_p^I and its neighbor nodes in $Loc(n_i^O)$ to $Q(n_i^S)$.
    else
        The current virtual node is a backup node n_i^A.
        Find the physical node n_i^I mapped by n_i^O.
        Add n_i^I's neighbor nodes in $Loc(n_i^A)$ to$Q(n_i^S)$.
    end if
end if
The nodes in $Q(n_i^S)$ are sorted in ascending order according to their failure probability.

The Algorithm 2 selects a candidate physical node set to each virtual node. A set $Loc(n_i^S) = \{n_u^I | loc_i^u = 1, \forall n_u^I \in N_I\}$ is firstly defined to describe a set of physical nodes that satisfy the location constraint for mapping the virtual node. If the current virtual node is the root node of the tree T, the physical nodes meeting the resource requirement of the virtual node and satisfying the position constraint will be added in the candidate physical node set $Q(n_i^S)$. Otherwise, the algorithm further acts depending on the type of the virtual node. If the current virtual node is an original virtual node n_i^O, the parent node of this virtual node in tree T (generated from Algorithm 1) is firstly found, noted by n_p^O. The physical node, toward which the parent node is mapped (noted by n_p^I) and n_p^I's neighbor nodes are added to the candidate physical node set $Q(n_i^O)$. If the current virtual node is a backup virtual node n_i^A whose original virtual node is denoted by n_i^O, the algorithm makes sure that these two nodes are maped towards closer physical nodes (but not the same physical node). Please also note that, in the virtual nodes sorting list, n_i^O is mapped before n_i^A, hence we map n_i^A towards the physical neighbor nodes of n_i^O. Finally, all the physical nodes in $Q(n_i^S)$ are sorted in ascending order according to the failure probability of the nodes. And, the resulting candidate physical node set $Q(n_i^S)$ is returned.

4.3 S-AVNE Algorithm

Algorithm 3 describes the overall process of the S-AVNE NS embedding algorithm. Based on the sorted list of $L(N_S)$, the virtual nodes (either original or backup) are mapped one by one, and the corresponding virtual links are mapped at the same time. Typically, virtual nodes with higher $S(i)$ value and their corresponding backup nodes are mapped first. For each virtual node, its candidate physical mapping nodes are obtained, and virtual links are mapped towards the candidate physical nodes through Dijkstra shortest path algorithm. The algorithm distinguishes between original nodes and backup nodes to avoid mapping these nodes to the same physical nodes and links. Hence, the reliability of the NS is effectively improved.

In order to prevent long distance mapping between adjacent virtual nodes in the physical network, the above S-AVNE algorithm adopts the scheme of coordinated mapping between virtual nodes and virtual links. Moreover, physical nodes reliability levels are considered in the process of node mapping to improve failure avoidance. The performance of the algorithm is evaluated through simulation.

Algorithm 3 S-AVNE algorithm

Require: Physical network topology G_I, the AVN G_S
Ensure: Mapping results of the NS to the physical network
1: Obtain sorted virtual node list $L(N_S)$ by Algorithm 1
2: **for** $n_i^S \in L(N_S)$ **do**
3: Obtain $Q(N_i^S)$ from the Algorithm 2
4: **if** n_i^S is an original virtual node **then**
5: **if** The current virtual node is the root node **then**
6: Map n_i^S to the first physical node in $Q(n_i^S)$.
7: **else**
8: **for** $n_u^I \in Q(n_i^S)$ **do**
9: Remove physical links with bandwidth lower than B_{ip}.
10: Use Dijkstra shortest path to map from n_u^I to n_p^I.
11: **if** The path does not meet the delay requirement **then**
12: Go to step 9) and try the next physical node in $Q(n_i^S)$.
13: **end if**
14: **end for**
15: **if** Failed to map the node **then**
16: return Map Failure
17: **end if**
18: **end if**
19: **else**
20: //The current node is a backup node.
21: **for** $n_u^I \in Q(n_i^S)$ **do**
22: Remove physical nodes and links mapped by its original virtual node and links. Remove physical links with less bandwidth than the requirement.
23: Using Dijkstra to map from n_u^I to its original nodes.
24: **if** The path does not meet the delay requirement **then**
25: Go to step 22) and try the next physical node in $Q(n_i^S)$.
26: **end if**
27: **end for**
28: **if** Failed to map the node **then**
29: return Map Failure
30: **end if**
31: **end if**
32: **end for**

5. Evaluations

We build a simulation platform to evaluate the performance of our proposed algorithm. Firstly, in the experiment, the underlying 5G physical network and NS's virtual network topology is generated by a modified Barabasi-Albert (BA) scale-free network model construction algorithm [35]. The BA model is largely utilized to represent the 5G system and NS's VN topology [18], [36]. Moreover, the physical network is composed of AN, TN and CN nodes, the proportion of these three nodes follows $3:4:3$.

For the physical network, physical node CPU capacity follows uniform distribution on the interval $[20, 60]$, physical link capacity follows uniform distribution on the interval $[200, 600]$, physical link delay follows uniform distribution on the interval $[3, 5]$. Especially, we consider the failure probability of physical nodes and links, which follows uniform distribution on the interval $[0, 0.05]$, and the default S_T value is set to 0.9. For virtual network, virtual node CPU requirement follows uniform distribution on the interval $[20, 30]$, virtual link bandwidth requirement follows uniform distribution on the interval $[5, 20]$ and virtual link delay requirement follows uniform distribution on the interval $[30, 50]$. Virtual network composition (node location specification) follows the same as the physical network. Simulations with the same setting are performed for 100 times, and the results are taken as the average value.

5.1 Simulations on the $S(i)$ Value

We first investigate our mechanism to compute node importance value $S(i)$ based on the TOPSIS method. Another conventional method for combining multiple attributes is using a linear combination such that $S(i) = \alpha L(i) + \beta G(i)$, then the value of $S(i)$ is tuned by changing the value of α and β. We compute $S(i)$ values for 30 NSs, each NS has 10 virtual nodes. We select the result of one NS and compare our method towards the linear combination method. The result is shown in Table 2. These three methods show the similar trends as node 5 has the highest importance value. However, our method produces $S(i)$ value within the range of $[0, 1]$. The linear combination has variable maximum $S(i)$ values. Then, our method is more practical since it involves less parameters.

We further investigate the impact of the node importance threshold value S_T. The simulation setting is 500 physical nodes and 30 NSs with each NS contains 10 virtual

Table 2 $S(i)$ value obtained through different methods.

Method	$S(1)$	$S(2)$	$S(3)$	$S(4)$	$S(5)$
Ours method	0.02	0.83	0.54	0.43	0.99
$0.3L(i) + 0.7G(i)$	0.12	11.01	7.15	5.71	13.10
$0.7L(i) + 0.3G(i)$	0.31	5.02	3.37	2.66	5.92
Method	$S(6)$	$S(7)$	$S(8)$	$S(9)$	$S(10)$
Our Method	0.01	0.32	0.43	0.02	0.01
$0.3L(i) + 0.7G(i)$	0.08	4.35	5.72	0.11	0.09
$0.7L(i) + 0.3G(i)$	0.19	2.15	2.68	0.27	0.21

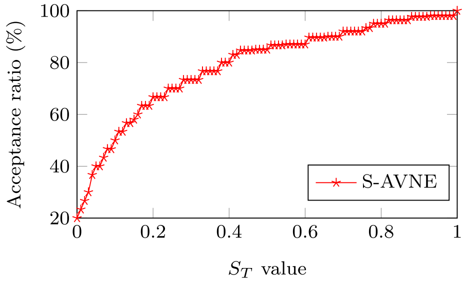

Fig. 2 Acceptance ratio with different S_T value.

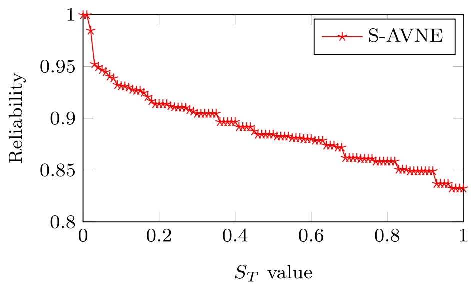

Fig. 3 Slice reliability with different S_T value.

nodes. For virtual nodes with importance value $S(i)$ higher than S_T, the node should be backed up. Hence, for lower S_T value, more virtual nodes are backed up and more resources are required by each NS. In Fig. 2, we show the acceptance ratio for different S_T value. For $S_T = 0$, all virtual nodes are backed up; for $S_T = 1$, no virtual node is backed up. Hence, the acceptance ratio increases as the S_T value increases. In Fig. 3, we calculate the reliability of the embedded NSs. The reliability is calculated based on Eq. (21) which calculates the NS's probability of providing services under failures. As the S_T value increases, less nodes are backed up, hence the reliability of the NSs decreases. The above simulation results demonstrate that the AVN consumes more physical resources since backup resources are added. But the reliability of the NS system is largely improved by the backup resources.

5.2 Simulations on NS Deployment Performance

We also compare our algorithm towards a baseline algorithm [25]. The baseline algorithm is a k-redundant algorithm where a fixed number of k nodes are backed up. For $k = 0$, no resource redundancy is deployed. We considered three cases, 0-redundant, 2-redundant and 5-redundant. The latter two deploy 2 backup nodes and 5 backup nodes respectively. Moreover, the baseline algorithm involves two steps: (1) the original VN is firstly mapped towards the physical network by utilizing the D-ViNE algorithm proposed in [37]; (2) after the determination of the mapping of the original VN, the complexity of the problem of embedding the VN with redundant backup nodes is significantly reduced, hence they utilize an optimizer (such as CPLEX) to map the redundant backup nodes.

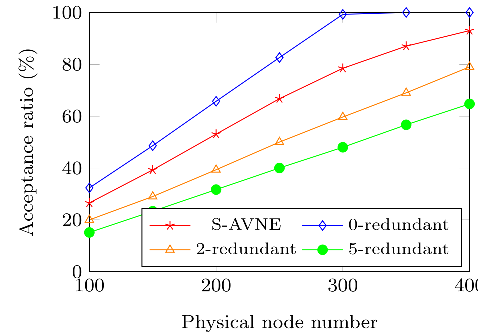

Fig. 4 Acceptance ratio with different physical networks.

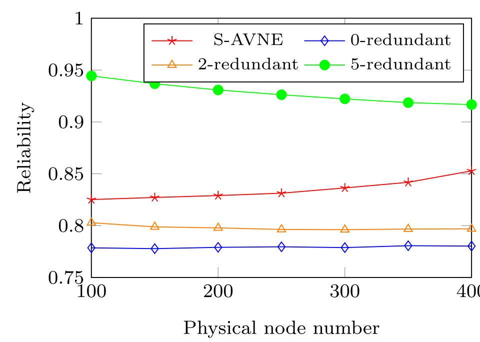

Fig. 5 Slice reliability with different physical network.

We conducted three sets of simulations, shown in Fig. 4 to Fig. 9. In the first set simulation, we vary the physical network scale from 100 physical nodes to 400 physical nodes, and the algorithms try to embed 30 NSs and each NS with 10 virtual nodes. Figure 4 shows the NSs acceptance ratio which is calculated as the number of embedded NSs over the total number of NSs. 0-redundant achieves the highest acceptance ratio, and S-AVNE shows higher acceptance ratio than 2-redundant and 5-redundant. This shows that S-AVNE is more efficient in resource utilization. In Fig. 5, we further calculate the reliability of the embedded NSs (the probability that no link and no nodes are crashed in the NS) for the two algorithms. Comparing to 0 and 2-redundant, S-AVNE's reliability is improved, however the reliability of 5-redundant is the highest due to the utilization of more backup resources. However, comparing to 2-redundant, our algorithm achieves higher acceptance ratio with higher reliability, which shows the efficiency of our solution.

Then, we vary the number of NSs from 20 to 70. Each NS's VN contains 10 virtual nodes and the underlying physical network has 500 physical nodes. Figure 6 shows the NSs acceptance ratio of the two algorithms. Comparing to 2 and 5-redundant, S-AVNE shows higher acceptance ratio. Figure 7 calculates the reliability of the embedded NSs. S-AVNE's reliability is improved comparing to 2-redundant algorithm, and 5-redundant achieves the highest reliability.

Finally, we vary the NS's VN scale from 10 virtual nodes to 40 virtual nodes for each NS. The underlying physical network contains 500 physical nodes, and the number of NSs is 30. Figure 8 shows the acceptance ratio of the two

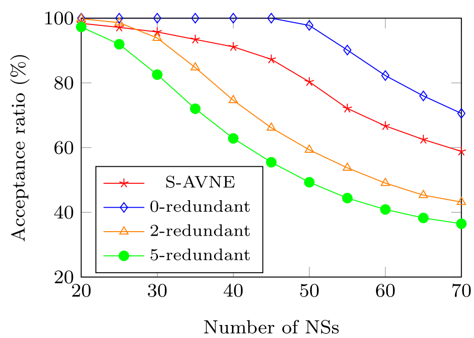

Fig. 6 Acceptance ratio with different number of slices.

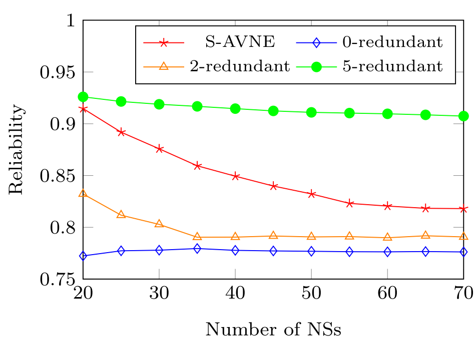

Fig. 7 Slice reliability with different number of slices.

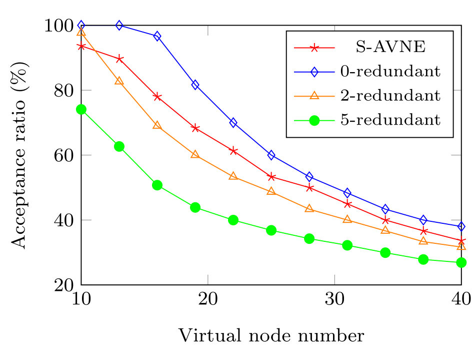

Fig. 8 Acceptance ratio with different virtual networks.

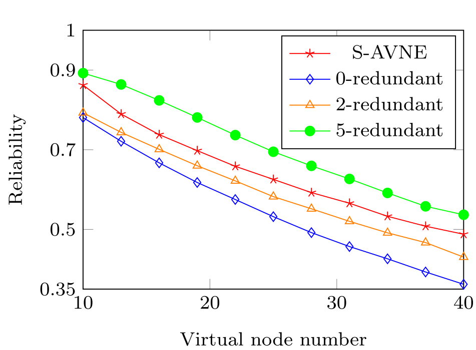

Fig. 9 Slice reliability with different virtual network.

algorithms. S-AVNE still achieves higher acceptance ratio than 2 and 5-redundant. Figure 9 calculates the reliability of the embedded NSs for the two algorithms. As the VN scales up, since each VN contains more physical nodes and physical links, then, it is more vulnerable than a VN with smaller scale. Hence, both algorithms' reliability decreases as the number of virtual nodes increases. S-AVNE still achieves higher reliability than 2-redundant algorithm.

6. Conclusions

Network Slicing is regarded as a key technology for the 5G network, and reliability is one of the most important key performance indices for 5G services. In this work, we investigate the NS service reliability enhancement mechanism to cope with physical network node failures. Typically, the reliability is improved by augmenting the virtual network with backup resources and optimally deploying the AVN. We use an ILP formulation to solve the S-AVNE problem. Due to the complexity of the problem, we propose an heuristic algorithm to solve the problem in practical time. Finally, we demonstrate the performance of the algorithms through intensive simulations.

Acknowledgments

This work is supported by the Laboratory Open Fund of Beijing Smart-chip Microelectronics Technology Co., Ltd.

References

[1] V. Nguyen, A. Brunstrom, K. Grinnemo, and J. Taheri, "Sdn/nfv-based mobile packet core network architectures: A survey," IEEE Commun Surveys Tuts., vol.19, no.3, pp.1567–1602, 2017.
[2] K. Samdanis, X. Costa-Perez, and V. Sciancalepore, "From network sharing to multi-tenancy: The 5G network slice broker," IEEE Commun. Mag., vol.54, no.7, pp.32–39, 2016.
[3] X. Foukas, G. Patounas, A. Elmokashfi, and M.K. Marina, "Network slicing in 5G: Survey and challenges," IEEE Commun. Mag., vol.55, no.5, pp.94–100, 2017.
[4] P. Popovski, C. Stefanovic, J.J. Nielsen, E. de Carvalho, M. Angjelichinoski, K.F. Trillingsgaard, and A. Bana, "Wireless access in ultra-reliable low-latency communication (URLLC)," IEEE Trans. Commun., vol.67, no.8, pp.5783–5801, 2019.
[5] A. Nakao, P. Du, and T. Iwai, "Application specific slicing for MVNO through software-defined data plane enhancing SDN," IEICE Trans. Commun., vol.E98-B, no.11, pp.2111–2120, Nov. 2015.
[6] A. Nakao, P. Du, Y. Kiriha, F. Granelli, A.A. Gebremariam, T. Taleb, and M. Bagaa, "End-to-end network slicing for 5G mobile networks," Journal of Information Processing, vol.25, pp.153–163, 2017.
[7] A. Fischer, J.F. Botero, M.T. Beck, H. de Meer, and X. Hesselbach, "Virtual network embedding: A survey," IEEE Commun. Surveys Tuts., vol.15, no.4, pp.1888–1906, 2013.
[8] M.R. Rahman and R. Boutaba, "Svne: Survivable virtual network embedding algorithms for network virtualization," IEEE Trans. Netw. Serv. Manage., vol.10, no.2, pp.105–118, 2013.
[9] M. Alsenwi, N.H. Tran, M. Bennis, A. Kumar Bairagi, and C.S. Hong, "eMBB-URLLC resource slicing: A risk-sensitive approach," IEEE Commun. Lett., vol.23, no.4, pp.740–743, 2019.
[10] B. Chang, L. Zhang, L. Li, G. Zhao, and Z. Chen, "Optimizing resource allocation in URLLC for real-time wireless control systems," IEEE Trans. Veh. Technol., vol.68, no.9, pp.8916–8927, 2019.
[11] X. Cheng, S. Su, Z. Zhang, H. Wang, F. Yang, Y. Luo, and J. Wang, "Virtual network embedding through topology-aware node ranking," SIGCOMM Comput. Commun. Rev., vol.41, no.2, pp.38–47, April

2011.
[12] A. Nakao, "Network virtualization as foundation for enabling new network architectures and applications," IEICE Trans. Commun., vol.E93-B, no.3, pp.454–457, March 2010.
[13] I. Afolabi, T. Taleb, K. Samdanis, A. Ksentini, and H. Flinck, "Network slicing and softwarization: A survey on principles, enabling technologies, and solutions," IEEE Commun. Surveys Tuts., vol.20, no.3, pp.2429–2453, 2018.
[14] S. Zhang, "An overview of network slicing for 5G," IEEE Wireless Commun., vol.26, no.3, pp.111–117, 2019.
[15] N. Huin, B. Jaumard, and F. Giroire, "Optimal network service chain provisioning," IEEE/ACM Trans. Netw., vol.26, no.3, pp.1320–1333, 2018.
[16] A. Blenk, P. Kalmbach, J. Zerwas, M. Jarschel, S. Schmid, and W. Kellerer, "NeuroViNE: A neural preprocessor for your virtual network embedding algorithm," IEEE INFOCOM 2018 - IEEE Conference on Computer Communications, pp.405–413, 2018.
[17] L. Gong, Y. Wen, Z. Zhu, and T. Lee, "Toward profit-seeking virtual network embedding algorithm via global resource capacity," IEEE INFOCOM 2014 - IEEE Conference on Computer Communications, pp.1–9, 2014.
[18] W. Guan, X. Wen, L. Wang, Z. Lu, and Y. Shen, "A service-oriented deployment policy of end-to-end network slicing based on complex network theory," IEEE Access, vol.6, pp.19691–19701, 2018.
[19] A. Jarray and A. Karmouch, "Decomposition approaches for virtual network embedding with one-shot node and link mapping," IEEE/ACM Trans. Netw., vol.23, no.3, pp.1012–1025, 2015.
[20] P. Zhang, H. Yao, and Y. Liu, "Virtual network embedding based on the degree and clustering coefficient information," IEEE Access, vol.4, pp.8572–8580, 2016.
[21] C. Mei, J. Liu, J. Li, L. Zhang, and M. Shao, "5G network slices embedding with sharable virtual network functions," J. Commun. Netw., vol.22, no.5, pp.415–427, 2020.
[22] M.M. Alam Khan, N. Shahriar, R. Ahmed, and R. Boutaba, "Multi-path link embedding for survivability in virtual networks," IEEE Trans. Netw. Serv. Manage., vol.13, no.2, pp.253–266, 2016.
[23] N. Shahriar, S.R. Chowdhury, R. Ahmed, A. Khan, S. Fathi, R. Boutaba, J. Mitra, and L. Liu, "Virtual network survivability through joint spare capacity allocation and embedding," IEEE J. Sel. Areas Commun., vol.36, no.3, pp.502–518, 2018.
[24] Y. Liu, P. Han, J. Hou, and J. Zheng, "Resource-efficiently survivable iot services provisioning via virtual network embedding in fiber-wireless access network," IEEE Access, vol.7, pp.65007–65018, 2019.
[25] H. Yu, V. Anand, C. Qiao, and G. Sun, "Cost efficient design of survivable virtual infrastructure to recover from facility node failures," 2011 IEEE International Conference on Communications (ICC), pp.1–6, 2011.
[26] A. Xiao, Y. Wang, L. Meng, X. Qiu, and W. Li, "Topology-aware virtual network embedding to survive multiple node failures," 2014 IEEE Global Communications Conference, pp.1823–1828, 2014.
[27] S. Li, M.Y. Saidi, and K. Chen, "A failure avoidance oriented approach for virtual network reliability enhancement," 2017 IEEE International Conference on Communications (ICC), pp.1–6, 2017.
[28] P. Zhang, S. Wu, M. Wang, H. Yao, and Y. Liu, "Topology based reliable virtual network embedding from a QoE perspective," China Commun., vol.15, no.10, pp.38–50, 2018.
[29] A. Hashmi and C.P. Gupta, "VNE-NR: A node-ranking method for performing topology-aware and resource-driven virtual network embedding," 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT), pp.1–6, 2020.
[30] S. Zhang, "Reliable virtual network mapping algorithm with network characteristics and associations," IEEE Access, vol.9, pp.48121–48130, 2021.
[31] A. Baumgartner, T. Bauschert, A.M.C.A. Koster, and V.S. Reddy, "Optimisation models for robust and survivable network slice design: A comparative analysis," GLOBECOM 2017 - 2017 IEEE Global Communications Conference, pp.1–7, 2017.
[32] B.M. Khorsandi, F. Tonini, E. Amato, and C. Raffaelli, "Dedicated path protection for reliable network slice embedding based on functional splitting," 2019 21st International Conference on Transparent Optical Networks (ICTON), pp.1–4, 2019.
[33] N. Shahriar, S. Taeb, S.R. Chowdhury, M. Zulfiqar, M. Tornatore, R. Boutaba, J. Mitra, and M. Hemmati, "Reliable slicing of 5G transport networks with bandwidth squeezing and multi-path provisioning," IEEE Trans. Netw. Serv. Manage., vol.17, no.3, pp.1418–1431, 2020.
[34] M. Behzadian, S.K. Otaghsara, M. Yazdani, and J. Ignatius, "A state-of the-art survey of TOPSIS applications," Expert Syst. Appl., vol.39, no.17, pp.13051–13069, 2012.
[35] A.L. Barabasi and R. Albert, "Emergence of scaling in random networks," Science, vol.286, no.5439, pp.509–512, 1999.
[36] W. Chang, G. Shou, Y. Liu, Z. Guo, Y. Hu, and X. Jin, "Constructing scale-free topologies for low delay of 5G," 2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC), pp.1–6, 2017.
[37] N.M.M.K. Chowdhury, M.R. Rahman, and R. Boutaba, "Virtual network embedding with coordinated node and link mapping," IEEE INFOCOM 2009, pp.783–791, 2009.

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.

Xin Lu received his Master's degree from Beijing University of Aeronautics and Astronautics (BUAA). He is Senior Engineer of Power Line Communication Application Technology Laboratory of State Grid. His research focuses on power line communication, wireless communication, electrical information collection systems, and Internet of Things.

Meiming Fu received his Bachelor's degree from Qingdao Binhai University. He is Engineer of Power Line Communication Application Technology Laboratory of State Grid. His research focuses on power line communication, 4G communication, 5G communication.

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.

Hongyan Yang received the B.S. degree in Xidian University, Xi'an, China, in 2018, where she is currently studying for a master's degree in school of communication engineering of Xidian University. Her main research direction is 5G network slicing.