IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received May 4, 2018, accepted June 1, 2018, date of publication June 19, 2018, date of current version July 6, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2848919

Fiber-Wireless Network Virtual Resource Embedding Method Based on Load Balancing and Priority

SIYA XU, PENG LI, SHAO-YONG GUO, AND XUESONG QIU

State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100086, China

Corresponding author: Siya Xu (xusiyaxsy@hotmail.com)

This work was supported by the National Key R&D Program of China under Grant 2016YFB0901200.

ABSTRACT Network virtualization is becoming one of the most promising ways to solve resource solidification problem of fiber-wireless access networks. To fully utilize substrate resources, a virtual resource embedding method including three sub-mechanisms is presented in this paper. The first one is the priority-based virtual network request (VNR) selecting mechanism, which sorts VNRs and preferentially selects higher priority VNRs for embedding. Then a dynamic embedding mechanism based on load balancing is presented to increase the acceptance rate of VNRs. Second, when embedding is saturated, an embedding update mechanism is proposed to deal with the arrival of new VNRs. It promotes the acceptance rate of high-priority VNRs and the revenue of the infrastructure provider (InP). Finally, a backup resource sharing mechanism is designed to improve reliability, which enables backup resources to be shared and responds to the failures with minimal resources. The simulation results show that the proposed embedding method can balance the resource consumption better than the Greedy Node Embedding and Shortest-path Link Embedding Algorithm, as well as the Greedy Node Embedding and Maximum Residual Bandwidth Link Embedding Algorithm. Besides, by applying our proposed algorithm, higher VNR acceptance rate, InP revenue, and resource utilization are achieved.

INDEX TERMS Fiber-wireless network, load balancing, network virtualization, dynamic embedding mechanism, resource utilization.

I. INTRODUCTION

In recent years, the emerging prosperity of cloud computing, mobile Internet, big data applications and the rapid progress of communication technologies have jointly promoted the development of backbone transmission network towards larger capacity and higher rates. Nevertheless, the development of access networks that connect backbone transmission network with local user network is relatively slow. As a result, the access network becomes the bottleneck of ''last mile'' communication networks [1], [2]. In view of the potential advantages and complementary features of optical access technologies and wireless access technologies, academic community has proposed a novel Fi-Wi converged broadband access network structure, which has become a hot research topic [3]–[6].

Fi-Wi converged broadband access networks effectively combine advantages of both wire and wireless access technologies, providing high-quality broadband access services to users anytime, anywhere. It inherits the high bandwidth capacity and high stability from optical access networks, as well as the high flexibility and affordability from wireless networks. As shown in Fig.1, the Fi-Wi access network generally adopts a ''Tree-Mesh'' topology: each segment consists of a tree topological PON network in the back-end and a wireless mesh network (WMN) in the front-end. Different segmented OLTs can be interconnected by optical fiber links to form a multi-segment Fi-Wi broadband access network. In practical applications, the PON in the back-end can adopt mature TDM-PON standard such as EPON or GPON, as well as the emerging technologies such as WDM-PON and OFDM-PON to support the deployment of the next generation optical access networks. The front-end network can adopt open-spectrum Wi-Fi technology. Note that, the technical standards of the front-end wireless access network and the back-end optical access network should fully consider the difference in bandwidth capacity between

2169-3536 © 2018 IEEE. Translations and content mining are permitted for academic research only.
Personal use is also permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

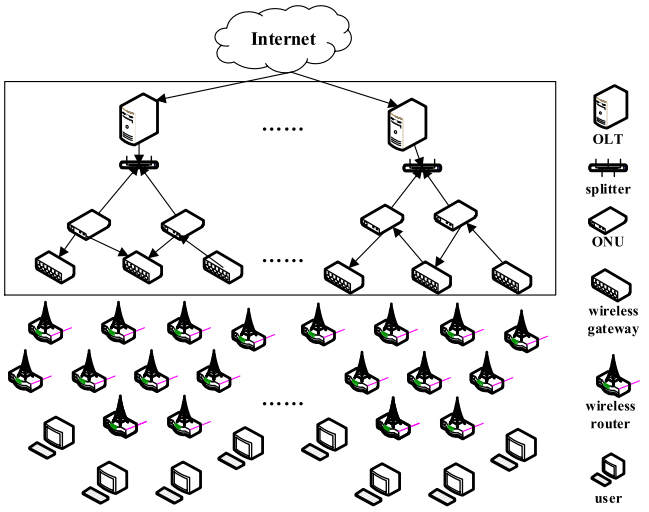

FIGURE 1. Framework of Fi-Wi network: The front-end is WMN (wireless routers and wireless gateways), and the back-end is PON (OLTs and ONUs).

wireless and fiber channels. Better rate matching of the front-end and the back-end can improve the overall resource utilization of Fi-Wi networks [7]–[10].

Compared with traditional access technologies, Fi-Wi broadband access networks have following advantages:

1) It supports flexible access anytime, anywhere.

2) It has higher data transmission capacity and reliability than wireless access networks.

3) It has stronger self-healing ability than optical access networks.

4) It is more cost-effective than fixed wired networks.

In summary, Fi-Wi network has become one of the most promising solutions for the ''last mile'' due to their high bandwidth capacity of PON and high flexibility of WMN. However, the differences in network topology, resource allocation mechanism, and protocol format between these two subnets make it hard to optimize Fi-Wi access networks globally. In addition, various network applications have placed higher demand for flexibility and scalability of Fi-Wi access networks.

Network virtualization has gained much popularity, allowing coexistence of heterogeneous virtual networks (VNs) on a common substrate network (SN). It allows multiple VNs to share SN's resources, and gives each VN an opportunity to run its own protocol. Through this approach, network resources can be managed and controlled more easily. Therefore, it is becoming one of the most effective ways to solve the solidification problem of Fi-Wi access networks and to optimize the overall utility of network resources [11]–[15]. In network virtualization, the internet service provider (ISP) can be divided into two parts: the infrastructure provider (InP) and the service provider (SP) [16]–[19].

As shown in Fig.2, InP who has Fi-Wi substrate resources is primarily responsible for deploying and managing substrate network infrastructure. By applying virtualization

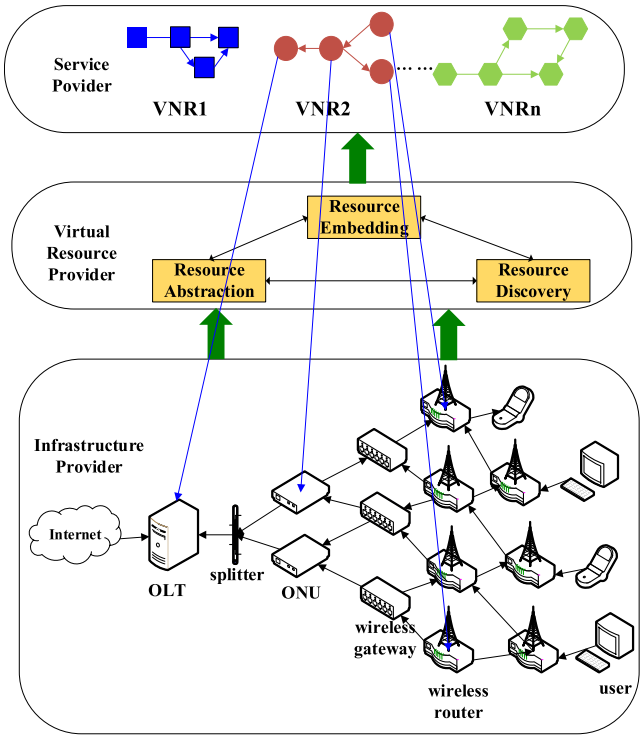

FIGURE 2. Functions of different roles: Service Provider (making different virtual network requests), Virtual Resource Provider (obtaining request information to create a virtual network) and Infrastructure Provider (providing Fi-Wi substrate resources).

technology, Fi-Wi network resources are abstracted as virtual resources. Then, virtual resources are sliced to be provided to different SPs. SP sends a virtual network request to the virtual resource provider (VRP) according to the needs of end-users and characteristics of service (i.e., required bandwidth, delay requirements, and geographical location constraints). The VRP obtains request information to create a virtual network, and be responsible for operation, maintenance and management of virtual networks. According to the requirements of SPs, virtual resources are found and rationally allocated to different SPs. On the basis of virtualization framework, VRP layer is between InP layer and SP layer, which simplifies the matching process between supply and demand into three steps: resource discovery, abstraction and embedding. Resource discovery is used to search for network resources and provide virtual resources to SP based on the usage of substrate resources and the operation status of virtual network. Resource abstraction is the process of abstracting network resources into virtual resource pools and logically cutting virtual resource into independent slices. Here, virtual resources include the power and CPU computing ability of virtual nodes, and bandwidth and spectrum of virtual links. In resource embedding stage, appropriate virtual resources are selected to form a virtual network and be provided it to SPs. Different virtual networks can develop and operate their own network architectures and protocols. Each SP can create more than one VNs according to end-user's needs, whereas

InP can manage resources in SN and rent them to SPs. The process of renting resources to SPs is called Virtual Network Embedding (VNE), and InPs will obtain revenue from SPs through VNE.

II. RELATED WORK

Network virtualization technology enables multiple virtual networks coexist on the same InP and share resources. To make full use of substrate resources, efficient and reliable resource allocation algorithms should be designed to reasonably substrate and schedule resources, aiming to improve resource utilization and achieve load balancing of the whole network.

Prior works [20]–[23] had been proposed to deal with VNE problems in virtualized Fi-Wi access networks. Dai and He *et al.* [24]–[26] applied virtualization into Fi-Wi networks [27], [28] to eliminate the differences between heterogeneous networks, and provided a unified virtual view of Fi-Wi networks. It simplified flow control and other traffic transmission algorithms in heterogeneous networks. In particular, a virtual resource manager was defined to operate heterogeneous networks. It allowed multiple routes to be used from the source node to the target node. Dai also examined load balancing performance of multiple-paths embedding modes [29], [30]. The simulation results showed that, the load-balanced virtualized Fi-Wi network significantly reduced the packet delay compared to the traditional Fi-Wi network. Besides, he constructed a test platform [31] and proposed a mathematical performance model of virtual Fi-Wi networks. The simulation results verified the performance improvements in Fi-Wi network virtualization in several aspects, including node throughput, link bandwidth utilization, and packet delay. In addition, it also demonstrated that Fi-Wi network virtualization improves the quality of experience (QoE) of video streaming. Meng *et al.* [30] adopted Fi-Wi network virtualization to overcome the differences in underlying physical infrastructure and broke the limitations of multi-path algorithms in traditional networks. He proposed a Modified Weighted Round Robin (MWRR) algorithm based on a fiber-optic wireless network virtualization model. According to the request's QoS requirements and the link state, the proposed algorithm arranged specific scheduling schemes to different requests to have lower end-to-end delay and better load balancing performance. Dashti *et al.* [32] studied Fi-Wi networking with a long-range PON with a propagation distance of 100km, considered a series of Fi-Wi network architectures with different dynamic bandwidth allocation (DBA) mechanisms, and proposed a new optical network unit (ONU) grouping strategy. In this strategy, when one group of ONUs merely transmit control information to optical line terminal (OLT), the ONUs belong to other group can utilize these idle channels. As a result, the OLT can achieve favorable packet delay performance through the proposed grouping method by cycle length (GCL) strategy.

In fact, different VNRs have differentiated QoS requirements. For instance, control services have high reliability requirements and low bandwidth requirement. While, as for video services with bigger data amount, they have high bandwidth requirement but general reliability requirements. However, previous works seldom considered the priorities of businesses. As a result, they could not provide differentiated service for different businesses. Therefore, to address this problem, we take QoS and priority factors into consideration. For example, we are eager to embed services with low latency requirements into non-critical nodes, in order to avoid resources competing with high-priority services to achieve load balancing. However, despite of their high resource consumption, services with strict latency requirements should be embedded first, Meanwhile, we will also prioritize the business with low resource demand and high returns. By this means, the Fi-Wi network virtualization based on priority and load balancing can avoid unbalanced resource consumption, and thereby increasing the reliability of the network.

In this paper, the incoming VNRs are prioritized first. Then, a load-balancing policy based on priority is presented for node and link embedding. In the end, a protection mechanism based on backup resource sharing is designed to improve network reliability. To sum up, the contributions of this paper are as follows:

1) A priority-based VNR selecting mechanism is presented. The value of priority is accurately calculated to sort incoming VNRs. High-priority requests are preferentially embedded for higher revenue.
2) A dynamic embedding mechanism based on load balancing is designed. During embedding process, nodes and links with more remaining resources are preferentially selected to be embedded. It achieves load balancing of entire network and increases the acceptance rate of VNRs.
3) An embedding updating mechanism is proposed. It improves long-term profitability of InP by embedding high-priority requests first when embedding is saturated. Besides, it sets three conditions for triggering re-embedding.
4) A backup resource sharing mechanism based on reliability is constructed. The substrate network is preprocessed before embedding, and the backup CPU computing resources are reserved and shared by all nodes in the same group.

III. SYSTEM MODEL

A. SUBSTRATE NETWORK

The substrate network can be represented as an undirected graph with weight $G^s = (N^s, L^s)$, where N^s is the set of substrate nodes. N^s_{num} is the number of substrate network nodes. For each substrate node $n^s_i \in N^s$, its CPU computing capability is $C\left(n^s_i\right)$, and its geographic location is $Loc\left(n^s_i\right) = (x_i, y_i)$. The type of node n^s_i is labeled by $type^s_i \in \{\text{olt, onu, wr, wg}\}$, where olt, onu, wr and wg denote OLT, ONU, wireless router and wireless gateway

respectively. L^s is the link set of substrate network, $l^s_{mn}\left(n^s_m, n^s_n\right)$ represents the link connecting node n^s_m and n^s_n. The link $l^s_{mn}\left(n^s_m, n^s_n\right)$ can be composed of several substrate links, and its usable bandwidth is the minimum bandwidth of all substrate links on the route between node n^s_m and n^s_n. The definitions of some important variables are given in TABLE 1.

Whether it is an optical fiber link or a wireless link, inter-link interference is inevitable. Ding et al. studied the interference problem in WMN [33]. They used four laptops equipped with a Netgear WAG511 802.11a/b/g PC Card. Paired laptops form a communication link, that is, the two end nodes of the link work on the same channel. They configured the two links with different channels and varied the distance between them. In this experiment, they used UDP packets in data transmission, aiming to measure the interference level between links configured with partially overlapping channels in 802.11b/g (2.4 GHz). They used the following metrics to evaluate the effect of interference. Let s_1 and s_2 be the throughput of *link1* and *link2*, on the condition that the other link is turned down. Let s'_1 and s'_2 be the throughput of *link1* and *link2* when both links are active. Then, the interference between these two links can be indicated by the interference factor *IF*:

$$IF = \frac{s'_1 + s'_2}{s_1 + s_2} \tag{1}$$

If $IF = 1$, it means that there is no interference between these two links, and they can transmit or receive data simultaneously. When $IF < 1$, interference exists between them. And, the lower the *IF* value is, the higher interference there will be.

The interference intensity of partially overlapping channels depends on not only the substrate distance between nodes but also the channel spacing. If the substrate distance between nodes is fixed, the link interference will be reduced with the increasing channel spacing. Once the channel spacing is fixed, the interference range will be the minimum distance between two nodes that do not interfere with each other. The interference of node m and n is defined as:

$$f(m,n) = \begin{cases} 1, & IR(|\tau|) \geq Dis(m,n) \\ 0, & IR(|\tau|) < Dis(m,n) \end{cases} \tag{2}$$

Where $IR(|\tau|)$ denotes the interference range with a channel spacing of τ. We can see from formula (2) that when interference range $IR(|\tau|)$ is greater than or equal to the substrate distance $Dis(m,n)$ between node m and n, these two nodes will interfere with each other. The experimentally measured interference ranges at different transmission rates are shown in TABLE 2, where R indicates the transmission range and the channel interval τ is set to be 0 to 5.

We use $B^{used}\left[l^s_{mn}\left(n^s_m, n^s_n\right)\right]$ to indicate the used bandwidth resources of substrate link $l^s_{mn}\left(n^s_m, n^s_n\right)$, and it is expressed as:

$$B^{used}\left[l^s_{mn}\left(n^s_m, n^s_n\right)\right] = \sum_{(G^S, G^V)} P^{Ind} B\left(l^v_{ij}\right) \tag{3}$$

TABLE 1. Definitions of the variables used in the model.

Symbol	Description
$G^s = \left(N^s, L^s\right)$	The undirected graph with weight for substrate network
N^s	Substrate nodes set
N^s_{num}	The number of substrate network nodes
$C\left(n^s_i\right)$, $i = 1,\ldots,N^s_{num}$	CPU computing capability of node n^s_i
$Loc\left(n^s_i\right) = \left(x_i, y_i\right)$, $i = 1,\ldots,N^s_{num}$	Geographic location of node n^s_i
L^s	The link set of substrate network
$l^s_{mn}\left(n^s_m, n^s_n\right)$, $m,n = 1,\ldots,N^s_{num}$	The link connecting node n^s_m and n^s_n
$B\left(l^s_{mn}\right)$	Bandwidth of link l^s_{mn}
$C^{init}\left[n^s_i\right]$, $i = 1,\ldots,N^s_{num}$	Initial CPU computing capability of node n^s_i
$B^{init}\left[l^s_{mn}\left(n^s_m, n^s_n\right)^s\right]$, $m,n = 1,\ldots,N^s_{num}$	Initial bandwidth of link $l^s_{mn}\left(n^s_m, n^s_n\right)$
$IR\left(\lvert\tau\rvert\right)$	Interference range with a channel spacing of τ
$Dis\left(m,n\right)$	Substrate distance between node m and n
$B^{used}\left[l^s_{mn}\left(n^s_m, n^s_n\right)\right]$, $m,n = 1,\ldots,N^s_{num}$	The used bandwidth resources of link $l^s_{mn}\left(n^s_m, n^s_n\right)$
$B^{re}\left[l^s_{mn}\left(n^s_m, n^s_n\right)\right]$, $m,n = 1,\ldots,N^s_{num}$	The remaining bandwidth resource of link $l^s_{mn}\left(n^s_m, n^s_n\right)$

TABLE 1. ***(Continued.)*** **Definitions of the variables used in the model.**

$\psi\left(n_m^s, n_n^s\right)$,	The average available link bandwidth between node
$m,n=1,\ldots,N_{num}^s$	n_m^s and n_n^s
$D\left(n_m^s, n_n^s\right)$, $m,n=1,\ldots,N_{num}^s$	Collision domain between node n_m^s and n_n^s
$G^v=\left(N^v, L^v\right)$	The undirected graph with weight for virtual network
N^v	The set of virtual nodes in virtual network
N_{num}^v	The number of virtual network nodes
$C^{re}\left(n_i^v\right)$, $i=1,\ldots,N_{num}^v$	CPU computing capability requirement of node n_i^v
L_{re}	Distance limitation of substrate node embedding
Del_{re}^v	Delay requirement of VNR
$B^{re}\left(l_{ij}^v\right)$, $i,j=1,\ldots,N_{num}^v$	Bandwidth requirement of virtual link l_{ij}^v

TABLE 2. **The interference ranges at different transmission rates.**

Transmission rate	Channel Spacing τ/R					
	0	1	2	3	4	5
2 Mbps	2	1.125	0.75	0.375	0.125	0
5.5 Mbps	2	1	0.625	0.375	0.125	0
11 Mbps	2	1	0.5	0.375	0.125	0

Where P^{Ind} is an indicator vector, indicating the embedding relationship between virtual link $l_{ij}^v\left(n_i^v, n_j^v\right)$ and substrate link $l_{mn}^s\left(n_m^s, n_n^s\right)$. If the substrate link carries the virtual link (that is, virtual link $l_{ij}^v\left(n_i^v, n_j^v\right)$ is embedded into substrate link $l_{mn}^s\left(n_m^s, n_n^s\right)$), then $P^{Ind}=1$, otherwise $P^{Ind}=0$.

$B^{re}\left[l_{mn}^s\left(n_m^s, n_n^s\right)\right]$ is the remaining bandwidth resource of substrate link $l_{mn}^s\left(n_m^s, n_n^s\right)$, which can be expressed by:

$$B^{re}\left[l_{mn}^s\left(n_m^s, n_n^s\right)\right] = B^{init}\left[l_{mn}^s\left(n_m^s, n_n^s\right)\right] - \sum_{(G^S, G^V)} P^{Ind} B\left(l_{ij}^v\right) \quad (4)$$

Note that, due to the radio interference, the sum of used bandwidth resources and remaining available bandwidth resources is less than or equal to the total bandwidth of a substrate link. And, all links share the remaining available bandwidth in a collision domain due to interference. In addition, the collision domain $D\left(n_m^s, n_n^s\right)$ is defined as the set of links that interfere with $l_{mn}^s\left(n_m^s, n_n^s\right)$.

Specifically, in PON sub-networks, collision domain contains all the optical links between OLT and ONUs. It is because that different ONUs cannot transmit packets to OLT simultaneously for PON is operated on Time Division Multiplexing (TDM) standard. In WMN sub-networks, the sufficient condition for an interference-free scheduling in WMN [15] has been shown as:

$$\sum_{l_{(p,q)}^S \in D_{(x,y)}} \frac{f_{(p,q)}^S}{b_{(p,q)}^S} \le 1, \quad \forall l_{(p,q)}^S \in L^S \quad (5)$$

The theorem can be expended to Fi-Wi access networks [34]. We introduce the average available link bandwidth $\psi\left(n_m^s, n_n^s\right)$ into collision domain $D\left(n_m^s, n_n^s\right)$, and it can be described as:

$$\psi\left(n_m^s, n_n^s\right) = \frac{B^{re}\left[l_{mn}^s\left(n_m^s, n_n^s\right)\right]}{\left|D\left(n_m^s, n_n^s\right)\right|} = \left(\frac{B^{init}\left[l_{mn}^s\left(n_m^s, n_n^s\right)\right]}{\left|D\left(n_m^s, n_n^s\right)\right|} - \frac{\sum_{(G^S, G^V)} P^{Ind} B\left(l_{ij}^v\right)}{\left|D\left(n_m^s, n_n^s\right)\right|}\right) \quad (6)$$

where $\left|D\left(n_m^s, n_n^s\right)\right|$ is the number of links in $D\left(n_m^s, n_n^s\right)$.

B. VIRTUAL NETWORK

Similarly, virtual network is also can be described by an undirected graph $G^v = (N^v, L^v)$, where N^v is the set of virtual nodes and N_{num}^v is the number of virtual network nodes. For each virtual node $n_i^v \in N^v$, it has a CPU computing capability requirement $C^{re}\left(n_i^v\right)$, distance limitation L_{re} for substrate node embedding, and service delay requirement Del_{re}^v of virtual network request. L^v is the link set of virtual networks, and each virtual link $l_{ij}^v \in L^v$ has a bandwidth requirement $B^{re}\left(l_{ij}^v\right)$. The bandwidth $B\left(l_{mn}^s\right)$ of substrate link should not be less than the bandwidth requirement $B^{re}\left(l_{ij}^v\right)$ of virtual link l_{ij}^v, that is,

$$B\left(l_{ij}^v\right) \le B\left(l_{mn}^s\right) \quad (7)$$

$C^{re}\left(n_i^v\right)$ represents CPU computing capability requirement of virtual node n_i^v. When performing embedding, the CPU computing capability of a substrate node should be stronger than the total requirement of virtual nodes. And, more than one virtual nodes are permitted to be embedded to the same substrate node. Besides, the distance of a VNR between substrate nodes must be less than the distance limitation L_{re} of VN.

C. COST, REVENUE AND PROFIT

According to network virtualization framework, virtual network makes a request to substrate network, and InP can get benefits only if substrate network accepts this request and finishes resource allocation. In general, the benefits of accepting a VNR depend on the service duration of the virtual network and the amount of occupied resources. Suppose that $T\left(G^v\right)$ is the duration that virtual network needs to service after resource embedding. p_{cpu} and p_B are adjustment parameters for computing capability and bandwidth, respectively. Therefore, the revenue for InP from accepting VNRs are described as follows:

$$R_{evenue}\left(G^v\right) = T\left(G^v\right) * \left(p_{cpu} * \sum_{n_i^v \in N^v} C^{re}\left(n_i^v\right) + p_B * \sum_{l_{ij}^v\left(n_i^v, n_j^v\right) \in L^v} B^{re}\left(l_{ij}^v\right) \right) \tag{8}$$

Where $B^{re}\left(l_{ij}^v\right)$ is the bandwidth requirement of virtual link l_{ij}^v, and $C^{re}\left(n_i^v\right)$ is the CPU computing capability requirement of virtual node n_i^v.

In addition, it will cost additional expense for InPs when providing backup resources to virtual network as a survivability guarantee. Let p_{pri} and p_{backup} be the adjustment parameters for primary resources and backup resources respectively. H is the number of virtual links in a VNR, and $\sum_{l_{mn}^s\left(n_m^s, n_n^s\right) \in L^s} B\left(l_{mn}^s\right)$ is bandwidth resource of substrate link corresponding to one virtual link. $\sum_{n_i^s \in N^s} C_{backup}\left(n_i^s\right)$ is CPU computing resource reserved for reliability requirements and $\sum_{n_i^s \in N^s} C_{pri}\left(n_i^s\right)$ is the actual amount of resource allocated to SP. And they are subject to:

$$C\left(n_i^v\right) \leq \sum_{n_i^s \in N^s} C_{pri}\left(n_i^s\right) \tag{9}$$

$$C\left(n_i^v\right) \leq \sum_{n_i^s \in N^s} C_{backup}\left(n_i^s\right) \tag{10}$$

So, the cost of accepting the virtual network for the InP is:

$$Cost\left(G^s\right) = T\left(G^s\right) * \left(\begin{array}{l} p_{cpu} * p_{pri} * \sum_{n_i^s \in N^s} C_{pri}\left(n_i^s\right) \\ + p_{cpu} * p_{backup} * \sum_{n_i^s \in N^s} C_{backup}\left(n_i^s\right) \\ + p_B * \sum_{i=1}^{H} \sum_{l_{mn}^s\left(n_m^s, n_n^s\right) \in L^s} B\left(l_{mn}^s\right) \end{array} \right) \tag{11}$$

For InPs, they get revenue by allocating resources to SPs and generate the cost for resource consumption at the same time. Therefore, the goal of our algorithm is to maximize the profit of InP, which is:

$$P_{rofit} = R_{evenue}\left(G^v\right) - C_{ost}\left(G^s\right) = T\left(G^v\right) * \left(\begin{array}{l} p_{cpu} * \sum_{n_i^v \in N^v} C\left(n_i^v\right) \\ + p_B * \sum_{l_{ij}^v\left(n_i^v, n_j^v\right) \in L^v} B\left(l_{ij}^v\right) \\ - p_{cpu} * p_{pri} * \sum_{n_i^s \in N^s} C_{pri}\left(n_i^s\right) \\ - p_{cpu} * p_{backup} * \sum_{n_i^s \in N^s} C_{backup}\left(n_i^s\right) \\ - p_B * \sum_{i=1}^{H} \sum_{l_{mn}^s\left(n_m^s, n_n^s\right) \in L^s} B\left(l_{mn}^s\right) \end{array} \right) \tag{12}$$

IV. ALGORITHM DESCRIPTION

A. PRIORITY-BASED VNR SELECTING MECHANISM

In this section, a priority-based VNR selecting mechanism is designed to realize the differentiated embedding. In order to maximize the revenue, InP needs to sort the incoming VNRs so that high-yielding VNs can have priority to be embedded. First, we precisely calculate the priority P_v of a VNR, which is a normalized value. P_v is calculated as:

$$P_v = P_{pro} + QoE_i * R_{res} \tag{13}$$

Where P_{pro} is an indicator factor for the protection service. The first half of formula (13) indicates the type of business. Only if the current service is a protection service, then $P_{pro} = 1$, and the embedding operation must be performed preferentially. Otherwise, if the current service is another kind of service, then $P_{pro} = 0$. QoE_i is a normalized QoE value for the current VNR, $R_{res}$is a normalized amount of resources. In this article, we assume that a VNR can only carry one type of service. And QoE_i and R_{res} are calculated as:

$$QoE_i = Q_R^i * (1 - QoS(X))^{\frac{QoS(X) * A_i}{R_i}} \tag{14}$$

$$QoS(X) = K * (P_{cpu} * C(n_i^v) + P_B * B(l_{ij}^v) + P_{delay} * Del_{re}^v) \tag{15}$$

Where Q_R^i, R_i, A_i are service-related fixed values, and K, P_{cpu}, P_B, P_{delay} are weights of QoS adjustment factor, CPU computing capability, link bandwidth and VNR delay.

$$R_{res}^a = \frac{p_{cpu} * \sum_{n_i^v \in N^v} C^a\left(n_i^v\right) + p_B * \sum_{l_{ij}^v\left(n_i^v, n_j^v\right) \in L^v} B^a\left(l_{ij}^v\right)}{p_{cpu} * \sum_{n_i^s \in N^s} C\left(n_i^s\right) + p_B * \sum_{l_{ij}^s\left(n_i^s, n_j^s\right) \in L^s} B\left(l_{ij}^s\right)} \tag{16}$$

The upper part of formula represents the number of weighted resources required by a VNR, and the lower part of formula is the total amount of weighted resources of the InP.

B. DYNAMIC EMBEDDING MECHANISM BASED ON LOAD BALANCING AND PRIORITY:

1) VN NODE EMBEDDING

Different from previous works, we operate node embedding based on load balancing. Therefore, when selecting a node for embedding, we firstly sort the CPU computing ability of all virtual nodes of the current VNR in descending order. Then, we prioritize to select the substrate nodes with more remaining CPU to embed in order to balance the resource of all substrate nodes.

$$\left[\begin{array}{c}\dfrac{C^{Init}\left(n_i^s\right)-C_{left}^{\max}\left(n_i^s\right)}{C^{Init}\left(n_i^s\right)}\\ -\dfrac{C^{Init}\left(n_j^s\right)-C_{left}^{\min}\left(n_j^s\right)}{C^{Init}\left(n_j^s\right)}\end{array}\right]\leq M \tag{17}$$

The above formula indicates that the difference between maximum and minimum CPU utilization of a node in the current substrate network should be less than a fixed value M. $C^{Init}\left(n_i^s\right)$ indicates the initial CPU computing capability of the current node n_i^s. $C_{left}^{\max}\left(n_i^s\right)$ and $C_{left}^{\min}\left(n_j^s\right)$ indicates the maximum and minimum remaining CPU computing capability of the current node respectively. This constraint should be satisfied when a substrate node is selected for embedding.

$$Loc_{(a,b)}=\sqrt{(x_a-x_b)^2+(y_a-y_b)^2}\leq L_{re} \tag{18}$$

This formula is a constraint of geographic location, meaning that the distance between any substrate nodes embedded by the same VNR must be within the distance limitation of VN.

2) VN LINK EMBEDDING

When carrying on link embedding, we also adopt a link embedding mechanism based on load balancing. Under this mechanism, the link with higher utilization can appropriately avoid being selected for embedding, while the link with a higher possible delay and lower utilization will be selected to achieve load balancing.

$$\left[\begin{array}{c}\dfrac{B^{Init}\left(l_{mn}^s\left(n_m^s,n_n^s\right)\right)-B_{left}^{\max}\left(l_{mn}^s\left(n_m^s,n_n^s\right)\right)}{B^{Init}\left(l_{mn}^s\left(n_m^s,n_n^s\right)\right)}\\ -\dfrac{B^{Init}\left(l_{op}^s\left(n_o^s,n_p^s\right)\right)-B_{left}^{\min}\left(l_{op}^s\left(n_o^s,n_p^s\right)\right)}{B^{Init}\left(l_{op}^s\left(n_o^s,n_p^s\right)\right)}\end{array}\right]\leq N \tag{19}$$

The above formula indicates that the difference between maximum bandwidth utilization and minimum bandwidth utilization of a link in the current substrate network should be less than a fixed value N. $B^{Init}\left(l_{mn}^s\left(n_m^s,n_n^s\right)\right)$ indicates the initial bandwidth of link $l_{mn}^s\left(n_m^s,n_n^s\right)$. Once a link is selected for embedding, this constraint should be met first. And the following constraint should be satisfied at the same time:

$$\sum_{l_{mn}^s\left(n_m^s,n_n^s\right)\in L^s} T_{l_{mn}^s}\leq Del_{re}^v \tag{20}$$

The formula above indicates that the delay between source node and sink node of the embedded substrate path should be less than the current delay requirement of VN. According to Michael's Lyapunov optimization theory [35]–[39], the worst-case delay is a constant. It is defined as:

$$W_n^{\max}<\frac{Q_n^{\max}+Z_n^{\max}}{\varepsilon_n} \tag{21}$$

Where $Q_n^{\max}$ and $Z_n^{\max}$ are finite upper bounds of actual queue and virtual queue. The worst-case guarantee theorem gives the delay of node, that is, all the data of a node must be processed and forwarded before $W_n^{\max}$. That is:

$$\overline{\tau_{\overrightarrow{uv}}}\leq W_n^{\max} \tag{22}$$

Where $\overline{\tau_{\overrightarrow{uv}}}$ is the average delay of a single hop.

A route length threshold is defined to limit the number of hops for source embedding. The worst case end-to-end delay is obtained by $H_L\times W_n^{\max}$ that bounds the delay as:

$$\sum_{\overrightarrow{uv}\in H_i}\tau_{\overrightarrow{uv}}=\overline{\tau_{\overrightarrow{uv}}}\times H_i\leq H_L\times W_n^{\max},\quad \forall H_i\in H \tag{23}$$

Where H_i is the hops of one path. Then, divide both sides of the inequality by $\overline{\tau_{\overrightarrow{uv}}}$, we can get:

$$H_i\leq H_L \tag{24}$$

Therefore, the above delay constraint can be converted into a hop constraint:

$$\sum_{l_{mn}^s\left(n_m^s,n_n^s\right)\in L^s} hop_{l_{mn}^s}\leq\frac{Del_{re}^v}{T_{value}} \tag{25}$$

$hop_{l_{mn}^s}$ denotes the hops of substrate link corresponding to the virtual link, T_{value} denotes the data delay between nodes, Del_{re}^v denotes the delay requirement of the current VNR.

C. PRIORITY-BASED EMBEDDING UPDATE MECHANISM

In addition to the previously mentioned mechanisms, a priority-based embedding update mechanism is designed to handle the unsatisfied VN. That is, when the proposed embedding mechanisms traverses to an unsatisfied VN, the QoE_i value of the current VN will update according to formula (14). The VN embedding continues until the traversal is completed. If there is a new VNR coming at this time, re-embedding will be triggered in the following cases:

1) If the new VNR is a protection service, $P_v^{new}=1$, then re-embedding will be triggered.
2) For other kind of business, If the QoE_i value of the new VN exceeds the pre-set threshold $QoE_{threshold}$, re-embedding will also be triggered. That is:

$$QoE_i\geq QoE_{threshold} \tag{26}$$

3) If the difference between P_v^{new} and $P_{v_{\min}}^{old}$ exceeds the pre-set threshold P_{value}, re-embedding will be triggered as well. That is:

$$P_v^{new}-P_{v_{\min}}^{old}\geq P_{value} \tag{27}$$

When satisfying the triggering conditions, if the current remaining resources can be successfully embedded to the new VN, the re-embedding will perform directly. Otherwise, the VNs that have been embedded successfully will be sorted in ascending order according to P_v. The VN with the lowest P_v should release the occupied resource. Note that, the expiring VNs will not be selected to released. Then VN embedding process continues according to embedding mechanism proposed in this article after the release. If the new VNR still unable to be embedded, then the next VN will be released until the re-embedding is successful.

We combine these three sub-mechanisms into a Dynamic Embedding Mechanism Heuristic Algorithm Based on Load Balancing and Priority(DEM-HA) and give the pseudo code:

As shown in Fig.3, our proposed embedding mechanism embeds VN1 and VN2 to the substrate network. When VN3 arrives, for VN3 is a protection service and meets the re-embedding condition proposed, the embedding update mechanism is triggered. As a result, VN2 is released, and VN3 is embedded.

D. RELIABILITY-BASED BACKUP RESOURCE SHARING MECHANISM

1) K-NODE NODE BACKUP

The K-means clustering algorithm is used to group all substrate nodes into K groups. The node with highest CPU computing capability is selected to be the backup node of the current group. The backup node reserves some of CPU computing capability as backup resources. Before implementing embedding mechanism proposed in this paper, all substrate nodes have been preprocessed, grouped and examined for backup. The backup node can be regarded as a node with reduced CPU computing capability. And we have:

$$C_{backup} \geq C(n_i^s) \tag{28}$$

The formula above indicates that the backup node has more CPU computing capability than other nodes in the same group. By this means, the backup node can replace other nodes in the same group in case of node failure to improve the reliability of substrate network.

2) 2) K-LINK LINK BACKUP

The backup node reserves part of CPU computing resources, as a result, the connecting links of the backup node will become relatively idle. Therefore, when a single-link fault occurs, the backup node can be used as an intermediate node to recover the faulty link. To further improve the reliability of the Fi-Wi networks, we propose a resource backup mechanism, which sacrifices the resources in exchange for reliability. If the reliability requirements of the current group are not satisfied, then a backup node is added to the current group. If the previous backup node cannot meet

Algorithm 1 Dynamic Embedding Mechanism Heuristic Algorithm Based on Load Balancing and Priority (DEM-HA)

Input:

$G^s = (N^s, L^s)$, $G^v = (N^v, L^v)$,

$C^{init}\left[n_i^s \in N^s\right], m, n = 1, \ldots, N_{num}^s$,

$Loc\left(n_i^s\right) = (x_i, y_i), i = 1, \ldots, N_{num}^s$,

$C^{init}\left(n_i^v\right), i = 1, \ldots, N_{num}^v$,

L_{re}, Del_{re}^v,

$B^{init}\left[l_{mn}^s\left(n_m^s, n_n^s\right)\right], m, n = 1, \ldots, N_{num}^s$,

$QoE_i, i = 1, \ldots, R_{num}^{req}$

Output:

$C^{left}\left[n_i^s \in N^s\right], i = 1, \ldots, N_{num}^s$,

$B^{left}\left[l_{mn}^s\left(n_m^s, n_n^s\right)\right], m, n = 1, \ldots, N_{num}^s$,

P_v

Initialization:

Assignment to $C^{init}\left[n_i^s \in N^s\right]$, $C^{init}\left(n_i^v\right)$, $Loc\left(n_i^s\right) = (x_i, y_i)$, L_{re}, Del_{re}^v, and $B^{init}\left[l_{mn}^s\left(n_m^s, n_n^s\right)\right]$

1: **for** all incoming VNRs, **do** find the VNR with the highest QoE_i to be selected for embedding first

2: Sort the virtual nodes in G^v in descending order according to $C\left(n_i^v\right)$

3: **for** $(i = 1; i+ = 1)$ **do**

4: **if** $((j == L^s))$ **then**

5: Select a suitable substrate node according to $C\left(n_i^v\right) \leq C_{pri}\left(n_i^s\right)$

6: Embed the virtual node

7: **go to** line 3

8: **else if** $\left(\left[\frac{C^{Init}\left(n_i^s\right) - C_{left}^{\max}\left(n_i^s\right)}{C^{Init}\left(n_i^s\right)} - \frac{C^{Init}\left(n_j^s\right) - C_{left}^{\min}\left(n_j^s\right)}{C^{Init}\left(n_j^s\right)}\right] \leq M$ and $Loc_{(a,b)} = \sqrt{(x_a - x_b)^2 + (y_a - y_b)^2} \leq L_{re}\right)$ **then**

9: **go to** line 15

10: **else if** $(i < N_{num}^s)$ **then**

11: Update substrate node and go to line 9

12: **else**

13: **go to** line 32

14: **end if**

15: **for** all paths, **do** find feasible paths, which satisfy $B\left(l_{ij}^v\right) \leq B\left(l_{mn}^s\right)$

16: Sort all feasible paths in ascending order according to $hop_{l_{mn}^s}$

17: **for** $(j = 1; j+ = 1)$ **do**

18: **if** $\left(\left[\frac{B^{Init}\left(l_{mn}^s\left(n_m^s, n_n^s\right)\right) - B_{left}^{\max}\left(l_{mn}^s\left(n_m^s, n_n^s\right)\right)}{B^{Init}\left(l_{mn}^s\left(n_m^s, n_n^s\right)\right)} - \frac{B^{Init}\left(l_{op}^s\left(n_o^s, n_p^s\right)\right) - B_{left}^{\min}\left(l_{op}^s\left(n_o^s, n_p^s\right)\right)}{B^{Init}\left(l_{op}^s\left(n_o^s, n_p^s\right)\right)}\right] \leq N$ and $\sum_{l_{mn}^s\left(n_m^s, n_n^s\right) \in L^s} hop_{l_{mn}^s} \leq \frac{Del_{re}^v}{T_{value}}\right)$ **then**

Algorithm 1 *(Continued.)* Dynamic Embedding Mechanism Heuristic Algorithm Based on Load Balancing and Priority (DEM-HA)

19: **go to** line 27
20: **else if** $(j == L^s)$ **then**
21: **go to** line 32
22: **else**
23: **go to** line 17
24: **end if**
25: **end for**
26: **end for**
27: **if** $(i == N^v_{num})$ **then**
28: **go to** line 36
29: **else**
30: **go to** line 3
31: **end if**
32: Update the QoE_i of the current VNR according to $QoE_i = Q^i_R * (1 - QoS(X))^{\frac{QoS(X)*A_i}{R_i}}$
33: **end for**
34: **for** remaining VNRs,**do** find a VNR, which has the highest P_v
35: **if** any VNR can't be embedded **then**
36: **go to** line 41
37: **else**
38: **go to** line 2
39: **end if**
40: **end for**
41: **if** a new VNR satisfies re-embedding triggering conditions, **then**
42: **if** remaining resources can meet the resource requirements of the new VNR **then**
43: Embed the VNR
44: **else**
45: Sort the embedded VNRs according to P_v in ascending order
46: Release VNR with minimum P_v and recalculate CPU and bandwidth resources
47: Embed the new VNR according to embedding mechanism
48: **if** the new VNR still unable to be embedded **then**
49: **go to** line 46
50: **end if**
51: **end if**
52: **end if**

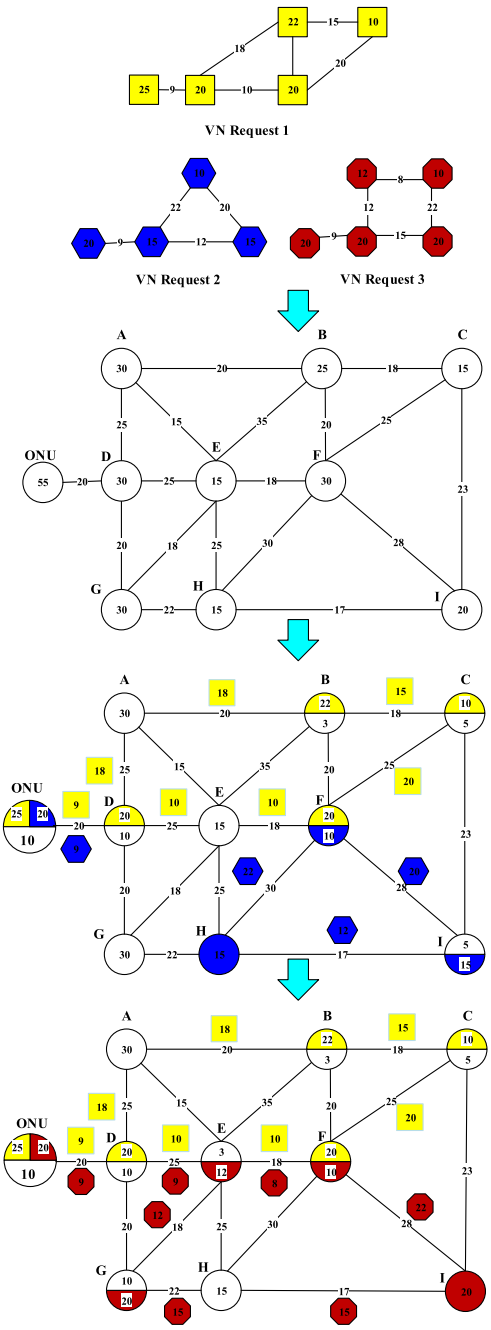

FIGURE 3. The process of re-embedding: At first, the old VNR (previous embedded VNR) releases resources, then the new virtual network request is embedded.

the requirements, then a new backup node is selected for embedding.

η_{res} denotes resource utilization of the entire substrate network, And we have:

$$\eta_{res} = \left[\begin{array}{l} p_{cpu} \times \dfrac{\sum C_{used}\left(n_i^s\right)}{\sum C\left(n_i^s\right)} \\ +p_B \times \dfrac{\sum B_{used}\left(l^s_{mn}\left(n_m^s, n_n^s\right)\right)}{\sum B\left(l^s_{mn}\left(n_m^s, n_n^s\right)\right)} \end{array} \right] \tag{29}$$

Where $\sum C_{used}\left(n_i^s\right)$ and $\sum C\left(n_i^s\right)$ represent the used CPU resources and the total CPU resources, $\sum B_{used}\left(l^s_{mn}\left(n_m^s, n_n^s\right)\right)$ and $\sum B\left(l^s_{mn}\left(n_m^s, n_n^s\right)\right)$ denote the used bandwidth resources and the total bandwidth resources. If more CPU computing resources are reserved, $\sum C_{used}\left(n_i^s\right)$ and η_{res} will be reduced.

η_{rel} is an reliability indicator of the entire substrate network, that is

$$\eta_{rel} = \frac{VNE^{num}_{success}}{X} \tag{30}$$

Where $VNE_{success}^{num}$ denotes the number of processing round before successful embedding after a single-link fault occurs. X indicates the number of randomly generated single-link faults. Therefore, the reliability is the probability of successful embedding when single-link fault occurs randomly. we can also get that, the more CPU computing resources reserved, the higher $VNE_{success}^{num}$ it will be.

The working mechanism of backup resource sharing is shown in Fig.4. Assume that all substrate nodes are in a same group. E is used as the backup node in the current group, and 30 CPU computing resources are reserved for backup. When a single node fails, E can be used as an alternative node for any node in this group for its reserved computing resources. If a single-link fault occurs (i.e. link A-B is failed), the route of A-B is unreachable. We can use E as an intermediate node and A-E-B as the new link to replace the previous failed link to ensure the network reliability. We give the pseudo code for the described mechanism.

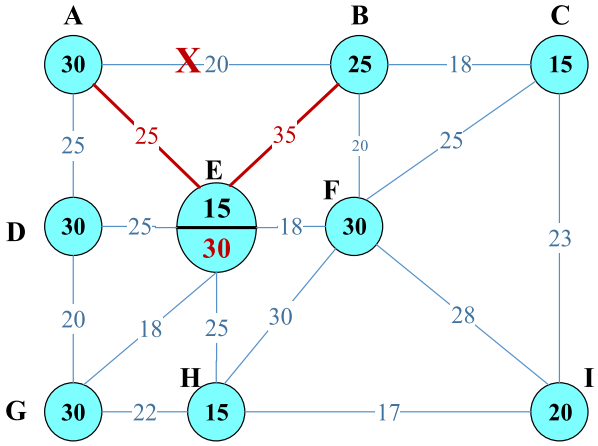

FIGURE 4. The process of resource backup: Select a node from the current group as the backup node, and then reserve some resources as backup resources.

Algorithm 2 Reliability-based Backup Resource Sharing Mechanism (Rb-BRSM)

Input:
$G^s = (N^s, L^s)$, $G^v = (N^v, L^v)$,
$C\left(n_i^s\right), i = 1, \ldots, N_{num}^s$,
$B\left(l_{mn}^s\right), m, n = 1, \ldots, N_{num}^s$
Output:
$C_{pri}\left(n_i^s\right), i = 1, \ldots, N_{num}^s$,
$C_{backup}\left(n_i^s\right), i = 1, \ldots, N_{num}^s$,
$B\left(l_{mn}^s\right), m, n = 1, \ldots, N_{num}^s$
Initialization:
Assignment to $C\left(n_i^s\right)$, $B\left(l_{mn}^s\right)$
1: Group all substrate nodes into K groups by the K-means clustering algorithm, so there are $\frac{Node_s^{sum}}{K}$ nodes in each group
2:**for** $(i = 1; i+ = 1)$ **do**
3: Select the node with highest CPU computing capacity from the $\frac{N_{num}^s}{K}$ nodes as the backup node of the current group
4: Allocate backup CPU computing resource C_{backup} and ensure $C_{backup} \geq C(n_i^s)$.
5: **if** $(i == K)$ **then**
6: **go to** line 10
7: **else**
8: **go to** line 2
9: **end for**
10: **if** a single-node fault occurs **then**
11: Use the backup node to replace the failed node.
12: **end if**
13: **if** a single-link fault occurs **then**
14: Use the backup node as an intermediate node to forward data around the failed node
15: **end if**

V. SIMULATION RESULTS AND ANALYSIS

In this section, we develop a network simulation platform based on Java and design a series of simulations to prove the effectiveness of proposed method. The parameter settings in simulation are listed in TABLE 3. The substrate network topology consists of 100 substrate nodes and about 400 substrate links. All substrate nodes are randomly distributed in a $100 * 100$ grid (the geographical location of simulated nodes). The forwarding capability of substrate nodes are evenly distributed between 30 and 40, and the bandwidths of substrate links are evenly distributed between 20 and 30. The number of virtual nodes in VNR topology is uniformly distributed from 2 to 10, and the nodes are randomly and evenly distributed in an $8 * 8$ grid. The allowable embedding range is 20, the connection rate between virtual nodes is 50%. The CPU computing capability of VNR is uniformly distributed from 0 to 5, and the bandwidth capability of virtual link is uniformly distributed from 0 to 3. Assuming 10 requests arrive at virtual network every 100ms, the lifetime of virtual network follows an exponential distribution with an average duration of 100 time-units.

To analyze the efficiency of resource allocation, different resources allocation strategies are simulated as comparison, which are:

1) Greedy Node Embedding and Shortest-path Link Embedding Algorithm (GN-Sp) [31]: One of the most classic algorithms in virtual network embedding to minimize costs of INPs.
2) Greedy Node Embedding and Maximum Residual Bandwidth Link Embedding Algorithm (GN-MaxBW): One of the improved shortest path algorithms, choosing links with the highest remaining bandwidth for embedding from all the shortest paths.
3) Priority-based Load Balancing Node and Link Embedding Algorithm (Pb-LB): It combines the priority-based VNR selecting mechanism and the dynamic embedding mechanism based on load balancing.

TABLE 3. Simulation parameter settings.

Parameter	Value
Number of nodes	100
Number of links	400
Distribution area	100*100
CPU computing capability of nodes	30-40
Bandwidth of links	20-30
Number of nodes for VNR	2-10
Distribution area of VNR	8*8
The allowable embedding range	20
Connection rate between virtual nodes	50%
CPU computing capability of VNR	0-5
Bandwidth of VNR	0-3
Arriving rate of VNR/100ms	10

We design Pb-LB algorithm in this article to achieve effective, differentiated, balanced and reliable service.

We simulated the embedding process of long-term online requests. By periodic sampling and monitoring, we got the following simulation results:

Fig.5 shows the total revenue of InP under three algorithms. It can be seen that, the revenue of three algorithms are similar when link utilization is relatively low at the beginning. After operating for a while, starting at 400ms, the advantage of the proposed algorithm becomes obvious. When virtual resource embedding is saturated, the total revenue of InP keeps in a relatively stable state. The total revenue of Pb-LB is stable at about 12×10^4, and the total revenue of GN-MaxBW and GN-Sp is stable at 8×10^4 to 10×10^4. At the same time, the total revenue of the proposed algorithm is about 33% higher than the other two algorithms. The reason for higher revenue is that, our algorithm preferentially embeds requests with higher-priority which have higher revenue. In addition, the embedding update mechanism further increases the acceptance rate of high-priority requests, resulting in higher total revenue in return.

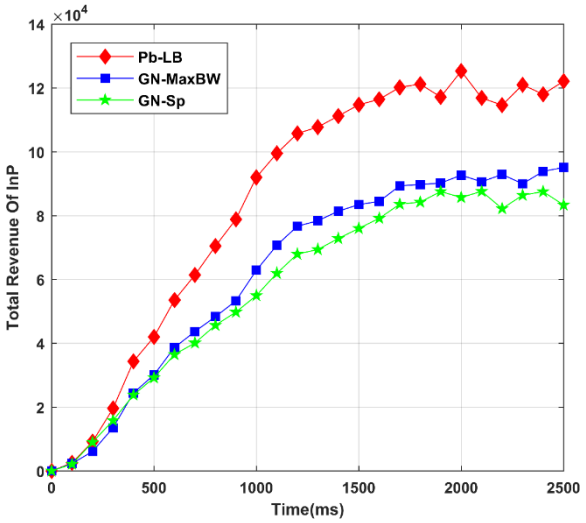

FIGURE 5. Total revenue of InP.

Fig.6 shows the total cost of InP under three algorithms. The link utilization is relatively low at the beginning, and all algorithms have approximately the same cost until 1,200 ms. Then, the cost of GN-MaxBW and Pb-LB start to increase. The cost of GN-Sp starts to reach saturation and be stabilized around 4,000. While, the cost of GN-MaxBW saturates at 2,000ms, and be stable at around 6,000. And the cost of Pb-LB reaches saturation at 2,300ms, stabilizes around 7,500. Aiming at achieving load-balancing, the proposed Pb-LB causes more cost. It is because that the requests with loose delay constraint actively avoid selecting the link with a high usage rate. On one hand, the GN-Sp chooses shortest path every time, so it has the lowest cost. On the other hand, the GN-Sp reaches saturation rapidly, and the acceptance rate of VNRs decreases rapidly due to its low embedding efficiency.

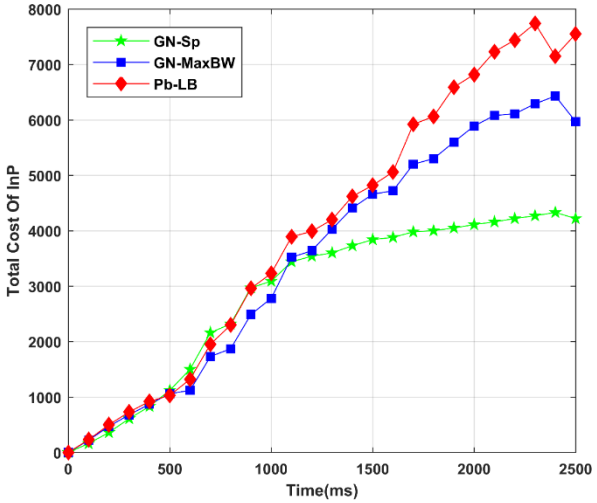

FIGURE 6. Total cost of InP.

Fig.7 shows the number of VNR for three algorithms. The ordinate is the number of VNRs, and the abscissa lists three algorithms. The blue histogram presents the number of all incoming VNRs. The gray histogram is the number of successfully embedded VNRs. The yellow histogram is the number of high-priority VNRs that was successfully embedded. The total number of incoming requests is 1,000. From the simulation results, Pb-LB has the highest embedding success rate compare to GN-MaxBW and GN-Sp, especially for high priority VNRs. The number of successfully embedded high-priority VNRs of Pb-LB algorithm is 110 and 136 more than the other two algorithms respectively. It can be explained that, the proposed Pb-LB algorithm sorts the service and embeds the services with higher priority. Moreover, the

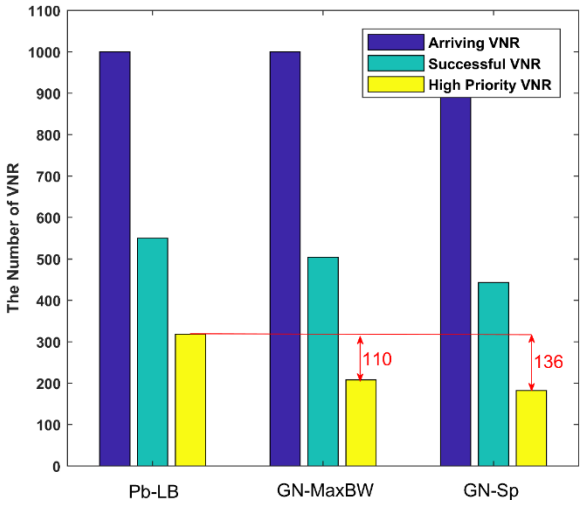

FIGURE 7. The number of VNR.

embedding update mechanism further increases the embedding success rate of high-priority requests.

Fig.8 shows the average acceptance rate of VNRs under three algorithms. It can be clearly seen that the proposed Pb-LB algorithm always has the highest acceptance rate, which remains at about 90% from the beginning until 3,000ms. Acceptance rate of GN-MaxBW keeps at about 90% from the beginning until 1,000ms, while the GN-Sp acceptance rate has been declining. The trends of three algorithms are roughly the same after 4,000 ms. GN-Sp tends to select the shortest path, resulting in unbalanced resource consumption. That is, some of the key nodes and critical links may be used for multiple times, causing their resources rapidly runs out and cannot be embedded any more. GN-MaxBW improves GN-Sp to some extent, it chooses the path with the most residual bandwidth resource from all shortest paths to relatively balance the network load,

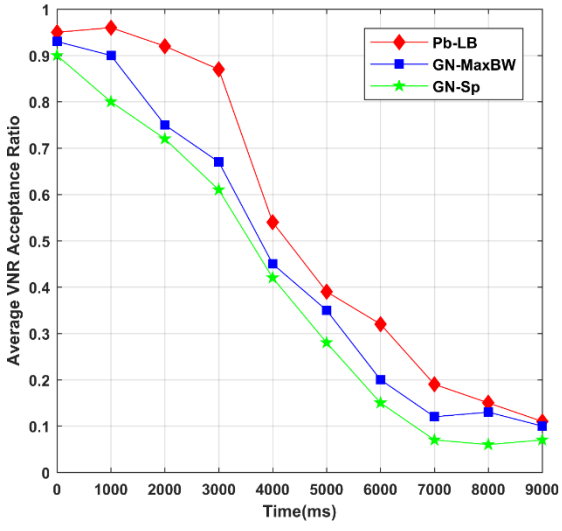

FIGURE 8. The average acceptance rate of VNRs.

so the request acceptance rate is elevated. The proposed Pb-LB adopts a load-balancing strategy for node and link embedding. While meeting hop constraints, the requests with low latency requirements avoid competing resource with the requests with strict latency requirements, to achieve load balancing of entire network, and thus resulting in the highest acceptance rate.

Fig.9 shows the variance of the remaining CPU computing resource under three algorithms. Since the remaining CPU resources of all nodes are evenly distributed from 30 to 40 during initialization, the variance of remaining CPU resources in initial state is not equal to 0. Due to the successful embedding of a single request, the remaining CPU computing resource of the embedded substrate nodes will decrease, and the variance will increase. As the embedding process goes on, all CPU computing resource will eventually be exhausted, so the variance begins to drop rapidly until it approaches zero. From the figure, we can see that the variance of the proposed Pb-LB begins to decline at about 600ms, while the other two algorithms begin to decline at about 1,400ms. It is because that, we prefer to select substrate nodes with more remaining CPU computing resources to embed, so that the remaining CPUs of nodes are more balanced.

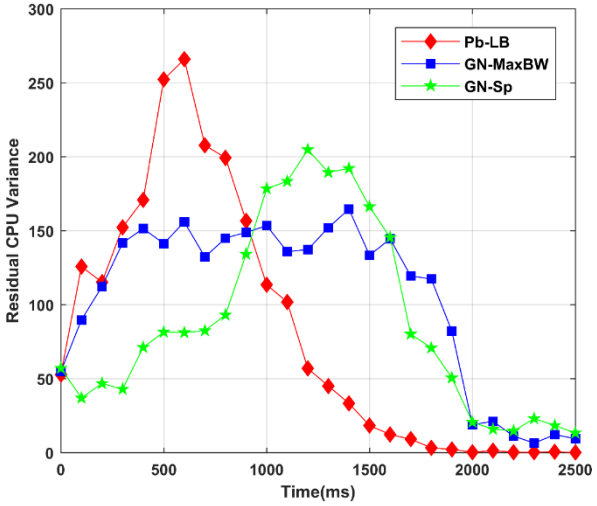

FIGURE 9. Variance of the remaining CPU computing resource.

Fig.10 shows the variance of the remaining bandwidth resources of three algorithms. Similar to the remaining CPU resources, the variance of the remaining bandwidth resource in the initial state is 58 by calculation. The decline begins at about 500ms under the proposed Pb-LB algorithm, while the other two algorithms begin to decrease from 900ms.

Note that, the variance of the remaining bandwidth resources of these three algorithms will not reduce to 0. It is because that, if the CPU computing resources of the node are exhausted, the node embedding cannot be performed despite of the remaining bandwidth of the links. The final values of the variance under these three algorithms are 79, 43, 12 respectively. Since GN-MaxBW and GN-Sp tend to select

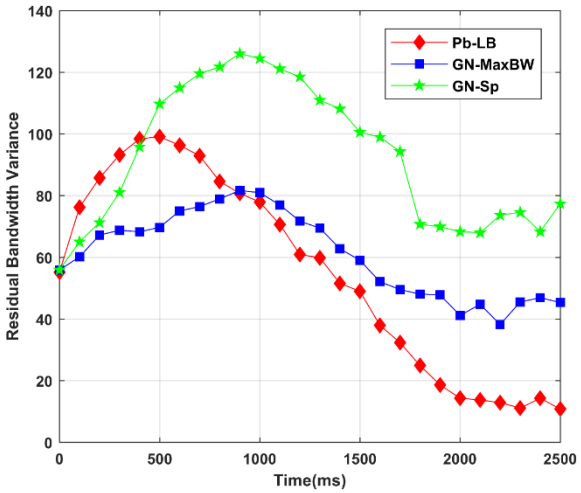

FIGURE 10. Variance of the remaining bandwidth resource.

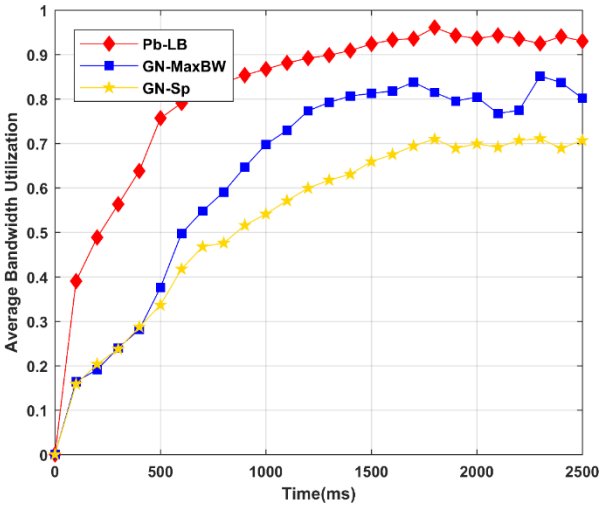

FIGURE 12. The average bandwidth utilization.

the shortest path, the available paths are limited and lead to a waste of resources. As a result, the variance of these two algorithms are higher than the proposed Pb-LB algorithm.

Fig.11 shows the average node utilization of three algorithms. It is shown in the figure that, at the beginning, three algorithms have almost the same node utilization. At about 500ms, we can see the advantages of the proposed Pb-LB algorithm clearly. Then, it reaches saturation first at 1,500 ms, and the utilization rate is about 90%. While GN-MaxBW and GN-Sp reach saturation at 2000ms, and the average node utilization rates are about 85% and 80% respectively. It can be explained that, the proposed Pb-LB algorithm adopts node embedding based on load balancing, resulting in a higher acceptance rate.

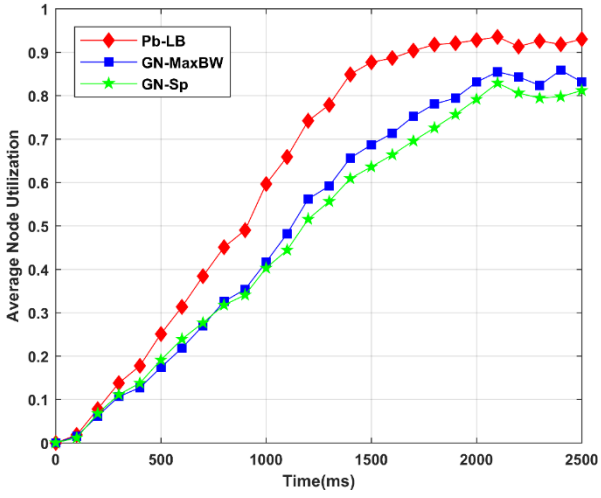

FIGURE 11. The average Node utilization.

Fig.12 reveals the average link utilization of three algorithms. Unlike the average node utilization, the average link utilization rate of Pb-LB algorithm rapidly shows the advantage from 100ms. It reaches saturation at 1,500ms with a utilization rate of 92%. GN-MaxBW and GN-Sp reach saturation at 1,800ms with an average link utilization of around 80% and 70% respectively. It is because that, according to the proposed Pb-LB algorithm, the use of links is more balanced, so the acceptance rate of VNRs and the average link utilization rate are higher than MaxBW and GN-Sp.

VI. CONCLUSIONS

Real-time network resources allocation based on virtualization technology is an important method to solve the solidification problem of Fi-Wi access networks. Therefore, it becomes crucial to make full use of substrate network through reasonable resource allocation algorithms. A dynamic embedding mechanism is proposed in this paper to achieve load balancing and maximizes resource utilization. When selecting nodes and links for embedding, it prefers to choose resources with lower utilization rate to balance the network resource and improve the acceptance rate of VNRs. At the same time, it also prioritizes to select higher priority VNRs to embed for increasing the revenue of InPs. Since we rank the priorities of VN requests, we tend to choose higher-priority service for embedding, such as services with high-revenue or strict latency requirements. Therefore, our proposed algorithm is more efficient and can quickly accept the request of SPs. The simulation results show that the total revenue of our algorithm is 33% higher than the comparison algorithms, the average node utilization is 5% and 10% higher than these other two algorithms, and the average bandwidth utilization is 12% and 22% higher than these other two algorithms.

In this paper, we discuss the Fi-Wi network virtual resource embedding for a single OLT and the corresponding ONU. After that, we will continue to study the issue of multiple ONU cross-domain embedding in future works. Furthermore, we also propose a node and link backup mechanism based on

backup resource sharing, and the specific details will continue to be discussed in later articles.

REFERENCES

[1] M. Maier and M. Lévesque, "Dependable fiber-wireless (FiWi) access networks and their role in a sustainable third industrial revolution economy," *IEEE Trans. Rel.*, vol. 63, no. 2, pp. 386–400, Jun. 2014.

[2] C. Ranaweera, E. Wong, C. Lim, A. Nirmalathas, and C. Jayasundara, "Architecture discovery enabled resource allocation mechanism for next generation optical-wireless converged networks," *IEEE/OSA J. Opt. Commun. Netw.*, vol. 5, no. 9, pp. 1083–1095, Sep. 2013.

[3] S. Basu and G. Das, "Scheduling hybrid WDM/TDM Ethernet passive optical networks using modified stable matching algorithm," *J. Lightw. Technol.*, vol. 32, no. 15, pp. 2613–2622, Aug. 1, 2014.

[4] M. Mahloo *et al.*, "Toward reliable hybrid WDM/TDM passive optical networks," *IEEE Commun. Mag.*, vol. 52, no. 2, pp. S14–S23, Feb. 2014.

[5] M. P. I. Dias, B. S. Karunaratne, and E. Wong, "Bayesian estimation and prediction-based dynamic bandwidth allocation algorithm for sleep/doze-mode passive optical networks," *J. Lightw. Technol.*, vol. 32, no. 14, pp. 2560–2568, Jul. 15, 2014.

[6] F. Aurzada, M. Scheutzow, M. Reisslein, N. Ghazisaidi, and M. Maier, "Capacity and delay analysis of next-generation passive optical networks (NG-PONs)," *IEEE Trans. Commun.*, vol. 59, no. 5, pp. 1378–1388, May 2011.

[7] C. Vulkan and B. Heder, "Congestion control in evolved HSPA systems," in *Proc. IEEE 73rd Veh. Technol. Conf. (VTC Spring)*, Yokohama, Japan, May 2011, pp. 1–6.

[8] S. Mohan, R. Kapoor, and B. Mohanty, "Enhanced HSDPA mobility performance: Quality and robustness for voice over HSPA service," in *Proc. IEEE 71st Veh. Technol. Conf.*, Taipei, Taiwan, May 2010, pp. 1–5.

[9] N. Ghazisaidi and M. Maier, "Fiber-wireless (FiWi) networks: A comparative techno-economic analysis of EPON and WiMAX," in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM)*, Honolulu, HI, USA, Dec. 2009, pp. 1–6.

[10] P. Chowdhury, B. Mukherjee, S. Sarkar, G. Kramer, and S. Dixit, "Hybrid wireless-optical broadband access network(woban): Prototype development and research challenges," *IEEE Netw.*, vol. 23, no. 3, pp. 41–48, May 2009.

[11] A. Belbekkouche, M. M. Hasan, and A. Karmouch, "Resource discovery and allocation in network virtualization," *IEEE Commun. Surveys Tuts.*, vol. 14, no. 4, pp. 1114–1128, 4th Quart., 2012.

[12] N. M. M. K. Chowdhury and R. Boutaba, "Network virtualization: State of the art and research challenges," *IEEE Commun. Mag.*, vol. 47, no. 7, pp. 20–26, Jul. 2009.

[13] M. Hoffmann and M. Staufer, "Network virtualization for future mobile networks: General architecture and applications," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC)*, Kyoto, Japan, Jun. 2011, pp. 1–5.

[14] K. Rattanaopas and K. Boonchuay, "A high performance network virtualization architecture with bandwidth guarantees," in *Proc. Int. Elect. Eng. Cong. (iEECON)*, Pattaya, Thailand, Mar. 2017, pp. 1–4.

[15] A. S. Khot, J. Gawas, and S. Waman, "Network virtualization on optical networks," in *Proc. Int. Conf. Wireless Commun., Signal Process. Networking (WiSPNET)*, Chennai, India, 2016, pp. 568–573.

[16] Y. Ou *et al.*, "Optical network virtualization using multi-technology monitoring and optical virtualize-able transceiver," in *Proc. Opt. Fiber Commun. Conf. Exhib. (OFC)*, Anaheim, CA, USA, 2016, pp. 1–3.

[17] J. Rubio-Loyola, C. Aguilar-Fuster, G. Toscano-Pulido, R. Mijumbi, and J. Serrat-Fernández, "Enhancing metaheuristic-based Online embedding in network virtualization environments," *IEEE Trans. Netw. Service Manag.*, vol. 15, no. 1, pp. 200–216, Mar. 2018.

[18] Y. Zhang, C. Jiang, L. Song, W. Saad, Z. Dawy, and Z. Han, "Complementary investment of infrastructure and service providers in wireless network virtualization," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Washington, DC, USA, Dec. 2016, pp. 1–6.

[19] S. Karmoshi, A. Hawbani, A. Ghannami, S. Mohammed, and M. Zhu, "VNE-Greedy: Virtual network embedding algorithm based on OpenStack cloud computing platform," in *Proc. 6th Int. Conf. Digit. Home (ICDH)*, Guangzhou, China, 2016, pp. 143–149.

[20] C. Liang and F. R. Yu, "Wireless network virtualization: A survey, some research issues and challenges," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 1, pp. 358–380, Mar. 2015.

[21] V.-G. Nguyen, T.-X. Do, and Y. Kim, *SDN and Virtualization-Based LTE Mobile Network Architectures: A Comprehensive Survey*. Norwell, MA, USA: Kluwer, 2016.

[22] P. Bhaumik, S. Zhang, P. Chowdhurym, S.-S. Lee, J. H. Lee, and B. Mukherjee, "Software-defined optical networks (SDONs): A survey," *Photonic Netw. Commun.*, vol. 28, no. 1, pp. 4–18, 2014.

[23] A. S. Thyagaturu, A. Mercian, M. P. Mcgarry, M. Reisslein, and W. Kellerer, "Software defined optical networks (SDONs): A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 4, pp. 2738–2786, 4th Quart., 2016.

[24] Q. Dai, J. Zou, G. Shou, Y. Hu, and Z. Guo, "Network virtualization based seamless networking scheme for fiber-wireless (FiWi) networks," *China Commun.*, vol. 11, no. 5, pp. 1–16, May 2014.

[25] Q. Dai, G. Shou, Y. Hu, and Z. Guo, "A general model for hybrid fiber-wireless (FiWi) access network virtualization," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC)*, Budapest, Hungary, Jun. 2013, pp. 858–862.

[26] S. He, G. Shou, Y. Hu, and Z. Guo, "Performance of multipath in fiber-wireless (FiWi) access network with network virtualization," in *Proc. IEEE Mil. Commun. Conf. (MILCOM)*, San Diego, CA, USA, Nov. 2013, pp. 928–932.

[27] C. Bock *et al.*, "Techno-economics and performance of convergent radio and fibre architectures," in *Proc. 16th Int. Conf. Transparent Opt. Netw. (ICTON)*, Graz, Austria, Jul. 2014, pp. 1–4.

[28] X. Liu and F. Effenberger, "Trends in PON—Fiber/wireless convergence and software-defined transmission and networking," in *Proc. IEEE OECC*, Shanghai, China, Jun. 2015, pp. 1–3.

[29] S. He, G. Shou, Y. Hu, and Z. Guo, "Intelligent multipath access in fiber-wireless (FiWi) network with network virtualization," in *Proc. OSA Asia Commun. Photon. Conf.*, Beijing, China, 2013, Paper AF2G.38.

[30] X. Meng, G. Shou, Y. Hu, and Z. Guo, "Efficient load balancing multipath algorithm for fiber-wireless network virtualization," in *Proc. IET Int. Conf. Inf. Commun. Technol. (ICT)*, Nanjing, China, May 2014, pp. 1–6.

[31] Q.-L. Dai, G.-C. Shou, Y.-H. Hu, and Z.-G. Guo, "Performance improvement for applying network virtualization in fiber-wireless (FiWi) access networks," *J. Zhejiang Univ. Sci. C*, vol. 15, no. 11, pp. 1058–1070, 2014.

[32] Y. Dashti, A. Mercian, and M. Reisslein, "Grouping by cycle length (GCL) for long-range FiWi networks," *Opt. Switching Netw.*, vol. 21, pp. 43–57, Jul. 2016.

[33] Y. Ding, Y. Huang, G. Zeng, and L. Xiao, "Using partially overlapping channels to improve throughput in wireless mesh networks," *IEEE Trans. Mobile Comput.*, vol. 11, no. 11, pp. 1720–1733, Nov. 2012.

[34] P. Han, L. Guo, Y. Liu, X. Wei, J. Hou, and X. Han, "A new virtual network embedding framework based on QoS satisfaction and network reconfiguration for fiber-wireless access network," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Kuala Lumpur, Malaysia, May 2016, pp. 1–7.

[35] M. J. Neely, "Opportunistic scheduling with worst case delay guarantees in single and multi-hop networks," in *Proc. IEEE INFOCOM*, Shanghai, China, Apr. 2011, pp. 1728–1736.

[36] M. J. Neely, "Delay-based network utility maximization," *IEEE/ACM Trans. Netw.*, vol. 21, no. 1, pp. 41–54, Feb. 2013.

[37] M. J. Neely, "Energy optimal control for time-varying wireless networks," *IEEE Trans. Inf. Theory*, vol. 52, no. 7, pp. 2915–2934, Jul. 2006.

[38] M. J. Neely, *Stochastic Network Optimization with Application to Communication and Queueing Systems*. San Rafael, CA, USA: Morgan & Claypool, vol. 2010, pp. 1–137.

[39] M. J. Neely, E. Modiano, and C. E. Rohrs, "Dynamic power allocation and routing for time-varying wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 1, pp. 89–103, Jan. 2005.

SIYA XU was born in Beijing, China, in 1988. She received the B.S. degree in communication engineering from the University of Science and Technology Beijing, the M.S. and Ph.D. degrees in communication and information system from the Beijing University of Posts and Telecommunications. She is currently a Post-Doctoral Researcher with the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications. Her research interests include smart grid communication network management and optimization.

PENG LI was born in Yuncheng, Shanxi, China, in 1994. He received the B.S. degree in communication engineering from the Minzu University of China. He is currently pursuing the M.S. degree with the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications.

SHAO-YONG GUO received the Ph.D. degree from the Beijing University of Posts and Telecommunication, Beijing, China, in 2013. He is currently a Lecturer with the Beijing University of Posts and Telecommunication. His research interests include blockchain, Internet of Things, ubiquitous network, and smart grids.

XUESONG QIU was born in 1973. He received the Ph.D. degree from the Beijing University of Posts and Telecommunications, Beijing, China, in 2000. Since 2013, He has served as the Deputy Director of the State Key Laboratory of Networking and Switching Technology. He is currently a Professor and a Ph.D. Supervisor with the Beijing University of Posts and Telecommunications. He is the author of over 100 SCI/EI index papers. His current research interests include network management and service management. He currently presides over a series of key research projects on network & service management, including the projects supported by the National Natural Science Foundation and the National High-Tech Research and Development Program of China. He was a recipient of awards and honors include 13 national and provincial scientific and technical awards, including the National Scientific and Technical Awards (second-class) twice.