A New Virtual Network Embedding Framework based on QoS Satisfaction and Network Reconfiguration for Fiber-Wireless Access Network

Pengchao Han[1], Lei Guo[1], Yejun Liu[1], Xuetao Wei[2], Jian Hou[1], Xu Han[1]
[1]College of Information Science and Engineering, Northeastern University, Shenyang, China
[2]School of Information Technology, University of Cincinnati, USA

***Abstract*—Fiber-Wireless (FiWi) access network, which could provide an anytime-anywhere access for end users with high bandwidth capacity and long distance, is facing the challenge of resource allocation and optimization due to the complexity and diversity of traffic demands. Though network virtualization becomes a promising solution, which allows heterogeneous virtual networks coexisting on the shared substrate network, previous works ignored both varied requirements of Quality of Service (QoS) satisfaction of virtual networks and the flexibility of reconfiguring the resource of substrate FiWi access network. In this paper, we propose a new VNE framework based on QoS satisfaction and network reconfiguration. By equipping each virtual network with a specific QoS satisfaction requirement, the characteristics of virtual network demands are formulated from a more practical point of view. Moreover, the adaptive bandwidth allocation of substrate network and virtual network reconfiguration are exploited to maximize the InP revenue. Simulation results demonstrate that our proposed VNE algorithm outperforms previous approaches with multifold increment of InP revenue.**

***Index Terms*—fiber-wireless, virtual network embedding, QoS satisfaction, network reconfiguration.**

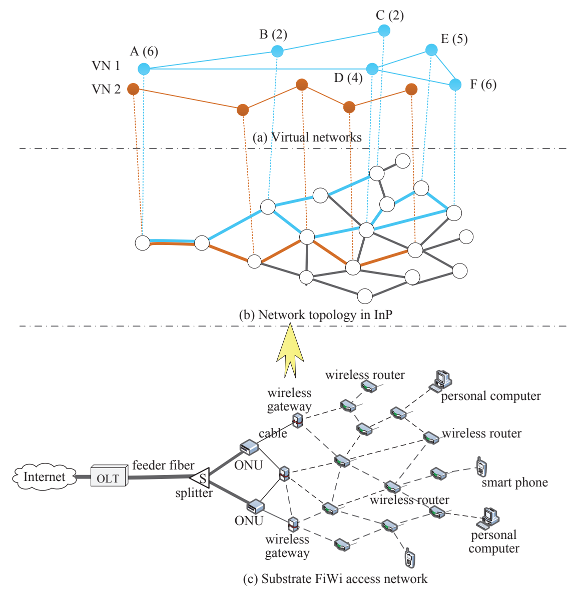

Fig. 1. Illustration of virtualized FiWi access network.

I. Introduction

Fiber-Wireless (FiWi) access network has become one of the most promising solutions for "last mile" due to both its high bandwidth capacity inherited from the back-end Passive Optical Network (PON) and flexibility benefited from the front-end Wireless Mesh Network (WMN) [1], [2]. However, the differences of network topology, resource allocation mechanisms and protocol formats between these two subnetworks pose a challenge for the global optimization of FiWi access network. Moreover, various kinds of network applications have put forward higher demands for the flexibility and scalability of FiWi access network [3]–[5].

Network virtualization has gained a lot of popularity for allowing heterogeneous Virtual Networks (VNs) coexisting on the common Substrate Network (SN) [6]–[8]. Network virtualization enables multiple VNs to share the resources of SN, and offers each VN the opportunity to run its own network protocols, which is more convenient to manage and control network resource. Thus, it is becoming one of the most promising ways to solve the problem of ossification in FiWi access network as well as achieving the global network resource optimization. In network virtualization, the Internet Service Provider (ISP) is decoupled into two parts, Infrastructure Provider (InP) and Service Provider (SP) [9]–[11]. Each SP is able to create at least one VN according to demands of end users, while InP can manage resources in SN and lease them to SPs for the creation of VNs accordingly. The process of leasing customized resources to SPs is called Virtual Network Embedding, in which InP will obtain revenue from SPs [12]. The illustration of virtualized FiWi access network is shown in Fig. 1, where network resources (CPU and bandwidth) will be first abstracted in InP and then leased to SPs for the VN service.

Prior works on virtualized FiWi access network have been proposed against the VNE problem. For example, in [13], the structure of virtualized FiWi access network was presented and the dynamic bandwidth allocation algorithm was used to achieve higher network resource efficiency and lower Round Trip Time (RTT). In [14], the heterogeneous nature of FiWi was handled by network virtualization, and a modified Weighted Round Robin (WRR) algorithm was proposed to provide services for traffic demands with different QoS requirements.

In order to overcome the deficiency of current two-stage VNE algorithm, the priority of substrate candidate nodes was defined in [15] according to Google's Pagerank algorithm so that the mapping of subsequent nodes and links was optimized and embedded smoothly. In [16], the difference between wired and wireless links was first analyzed, and then the VNE algorithm based on channel reconfiguration in 802.11 WMN was proposed. Results showed that the channel reconfiguration can effectively improve virtual network acceptance ratio and InP revenue.

However, most of prior work improved the InP revenue by just optimizing the embedding of VNs, and they all ignored characteristics of real-world traffic demands and substrate FiWi access network. For example, InP was assumed to provide all the required resources for each accepted VN. In the real world, VNs have different QoS requirements, which reflects that VNs have varied QoS satisfaction requirements. Furthermore, not only channel reallocation causes high cost for the SN to frequently switch the channels of radio interfaces, but also VNs with higher CPU or bandwidth resource requirements are harder to find feasible embedding solutions. Thus, a flexible network reconfiguration is required to better achieve the global resource optimization. For example, improving the virtual network acceptance ratio can be achieved by optimizing the embedding order of VNs that arrive in a certain time window and reconfiguring the embedding of existing VNs without channel reallocation.

In this paper, we propose a new VNE framework to maximize the InP revenue in FiWi access network, where each online VN request is associated with node and link constraints as well as a QoS satisfaction requirement that is embedded on the substrate network. Our contributions are as follows:

- We are the first to formulate each VN with the QoS satisfaction requirement to obtain practical models of characteristics and constraints of virtual networks in FiWi access network.
- We design novel VNE algorithms, which are based on the QoS satisfaction of each online VN and new network reconfiguration operations, to maximize the InP revenue. These new network reconfiguration operations (i.e., substrate network reconfiguration and virtual network reconfiguration) are proposed to take both the type of network congestion and the cost of reconfiguration into consideration strategically.
- We propose an adaptive bandwidth allocation scheme to achieve the low-cost substrate network reconfiguration.

The remainder of this paper is organized as follows. Section II presents the mathematical models of substrate network, virtual network and the VNE problem in FiWi, respectively. The heuristic VNE algorithms are proposed in section III. Simulation is implemented in section IV to analyze the performance of the proposed algorithms. Finally, section V concludes this paper.

II. Network Models and Problem Formulation

A. Substrate Network Model

The substrate network can be represented by a weighted undirected graph $G^S = (N^S, L^S)$, where N^S and L^S indicate the sets of substrate nodes and links, respectively. For each substrate node $n_x^S \in N^S$, $x \in \{1, 2, \cdots, N\}$, where N is the total number of substrate nodes including OLT, ONUs, and wireless routers (gateways), the type of n_x^S is labeled by t_x^S and $t_x^S \in \{olt, onu, wr\}$ where *olt*, *onu*, and *wr* denotes OLT, ONU, and wireless router (gateway), respectively. The CPU capacity and the location of n_x^S are represented by c_x^S and lct_x^S, respectively. For each link $l_{(x,y)}^S \in L^S$, $x, y \in \{1, 2, \cdots, N\}$ between n_x^S and n_y^S, $b_{(x,y)}^S$ denotes its bandwidth capacity and $f_{(x,y)}^S$ denotes its traffic load. The binary variable $e_{(x,y)}^S$ takes 1 if the link $l_{(x,y)}^S$ exists and 0 otherwise. In addition, the collision domain $D_{(x,y)}$ of $l_{(x,y)}^S$ is defined as the set of links that interfere with $l_{(x,y)}^S$. Specifically, in PON subnetwork, $D_{(x,y)}$ contains all the optical fibers between OLT and ONUs because OLT cannot transmit packets to different ONUs simultaneously when PON is operated on Time Division Multiplexing (TDM) standard. In WMN subnetwork, $D_{(x,y)}$ is formulated as (1):

$$D_{(x,y)} = \{l_{(p,q)}^S \in L^S | ch_{(p,q)}^S = ch_{(x,y)}^S \text{ and } [d(x,p) \le d^I \text{ or } d(x,q) \le d^I \text{ or } d(y,p) \le d^I \text{ or } d(y,q) \le d^I]\}. \quad (1)$$

where $ch_{(x,y)}^S$ indicates the channel of $l_{(x,y)}^S$, d^I denotes the jamming range of wireless radios under given emitting power and $d(x,y)$ indicates the Euclidean distance between n_x^S and n_y^S. According to (1), the collision domain of $l_{(x,y)}^S$ contains the links that are working on the same channel as $l_{(x,y)}^S$ and their source or destination nodes are within the jamming range of n_x^S or n_y^S. Particularly, the link itself is also contained in its collision domain. All the links in the same collision domain should transmit traffic by TDM. Moreover, the sufficient condition for an interference-free schedule in WMN [17] has been shown as (2):

$$\sum_{l_{(p,q)}^S \in D_{(x,y)}} \frac{f_{(p,q)}^S}{b_{(p,q)}^S} \le 1, \forall l_{(x,y)}^S \in L^S. \quad (2)$$

The theorem can also be expanded into FiWi access network. We denote the number of links in $D_{(x,y)}$ as $|D_{(x,y)}|$, thus the average residual bandwidth capacity of $D_{(x,y)}$ can be represented as (3):

$$\psi_{(x,y)} = \frac{p_{(x,y)}^S}{|D_{(x,y)}|} = \frac{C^R - \sum_{l_{(p,q)}^S \in D_{(x,y)}} f_{(p,q)}^S}{|D_{(x,y)}|}. \quad (3)$$

where $p_{(x,y)}^S$ indicates the potential bandwidth capacity of $l_{(x,y)}^S$ and C^R denotes the bandwidth capacity of $ch_{(x,y)}^S$.

B. Virtual Network Model

The virtual network is represented by $G^V = (N^V, L^V)$, where N^V and L^V are the sets of virtual nodes and links, respectively. The arrival time and duration of G^V are labeled by t^a and t^d, respectively. For each node $n_i^V \in N^V$, $i \in \{1, 2, \cdots, M\}$, where M denotes the total number of virtual nodes, it is associated with a node type t_i^V that designates the node function (A for "access" and T for "transmit"), a CPU demand c_i^V, a preferred location lct_i^V and a distance constraint Δlct_i that limits the distance of embedded node from lct_i^V. Each link $l_{(i,j)}^V \in L^V$, $i, j \in \{1, 2, \cdots, M\}$ is associated with a bandwidth demand $b_{(i,j)}^V$ and a path length constraint $len_{(i,j)}^V$ for the embedded path in SN in the unit of hops. In addition, each VN is associated with a specific QoS satisfaction requirement $s^q \in [0, 1]$, which specifies the least level of resources that InP should provide to accept the VN.

C. VNE Problem in FiWi Access Network

Based on the above, the problem of VNE in FiWi access network is to find a feasible mapping solution in SN that satisfies the resource constraints of VN meanwhile achieves the objective of maximizing InP revenue as (4):

$$\begin{aligned} \text{Revenue}(G^V) = t^d * \Bigg\{ \alpha \left(\rho^n \sum_{i=1}^{M} c_i^V + \rho^l \sum_{i=1}^{M} \sum_{j=1}^{M} b_{(i,j)}^V \right) - \\ \beta \left(\rho^n \sum_{i=1}^{M} (s_i^q \cdot c_i^V) + \rho^l \sum_{i=1}^{M} \sum_{j=1}^{M} (s_{(i,j)}^q \cdot b_{(i,j)}^V \cdot len_{(i,j)}^S) \right) - \\ \gamma \Bigg(\rho^n \sum_{i=1}^{M} (1 - s_i^q) \cdot c_i^V + \\ \rho^l \sum_{i=1}^{M} \sum_{j=1}^{M} ((1 - s_{(i,j)}^q) \cdot b_{(i,j)}^V \cdot len_{(i,j)}^S) \Bigg) \Bigg\}. \end{aligned} \tag{4}$$

where $len_{(i,j)}^S$ indicates the length of substrate path on which $l_{(i,j)}^V$ is embedded. The InP revenue consists of three parts, i.e., the InP income, the InP resource cost and the QoS penalty of InP. The coefficients α, β and γ stand for the income, resource cost, and QoS penalty of each unit of resource respectively, and ρ^n and ρ^l are the relative weights between CPU and bandwidth. It should be noticed that s_i^q and $s_{(i,j)}^q$ are the actual QoS levels provided by InP, where they represents the QoS satisfactions of n_i^V and $l_{(i,j)}^V$, respectively. Based on above objective, the process of VNE can be divided into two parts:

- Node embedding: Each virtual node will be embedded to a substrate node that satisfies the CPU demand and location constraint. Moreover, the virtual node with $t_i^V = T$ should be embedded onto ONU or wireless router (gateway), while virtual node with $t_i^V = A$ should be embedded only onto OLT because OLT plays the role of access node of FiWi to Internet.
- Link embedding: Each link in VN will be embedded onto a loop-free path in substrate network that does not violate the bandwidth demand and path length constraint.

III. Heuristic VNE Algorithms

A. VNE Algorithm Based on QoS Satisfaction

The procedure of VNE in FiWi access network is described in Algorithm 1. Each VN G^V is embedded on the SN G^S by determining the embedding solution of each virtual node and its related links with maximum InP revenue. In Algorithm 1, $e_{(i,j)}^V = 1$ indicates that the link $l_{(i,j)}^V$ exists, $\xi_i = 1$ implies that n_i^V has been embedded successfully, and $\chi_{(i,x)} = 1$ represents that n_i^V is embedded to n_x^S, and 0 otherwise. $distance(n_i^V, n_x^S)$ states the distance between n_i^V and n_x^S, and r_x^S indicates the residual CPU capacity of n_x^S. In the proposed VNE algorithm, all the VNs that arrive in a certain time window will be embedded together in the nonincreasing order of InP income. For each VN, all of its virtual nodes will be first ranked according to their CPU demands and the network connectivity. For example the VN 1 in Fig. 1 (a), where the numbers beside the capital letters indicate the CPU

Algorithm 1 VNE in FiWi access network

Input: $G^V, G^S, \alpha, \beta, \gamma, \rho^n, \rho^l, \Delta s^q, \Delta d$.
Output: $\chi_{(i,x)}, s_i^q, s_{(i,j)}^q, \forall i, j \in \{1, 2, \cdots, M\}$, $x \in \{1, 2, \cdots, N\}$.

1: **Initialization:** $\chi_{(i,x)} \leftarrow 0, \xi_i \leftarrow 0$, $i \leftarrow 0$.
2: Rank the virtual nodes in G^V accordingly;
3: **for** all the adjacent links of node n_i^V **do**
4: Find the links $l_{(i,j)}^V$ that satisfy $e_{(i,j)}^V = 1$ and $\xi_j = 1$, which will be embedded together with n_i^V;
5: **end for**
6: $s_i^q \leftarrow 1$, $s_{(i,j)}^q \leftarrow 1, \forall l_{(i,j)}^V$ above;
7: **for** all n_x^S with $r_x^S \geq c_i^V \cdot s_i^q$ and $distance(n_i^V, n_x^S) < \Delta lct_i$ **do**
8: Find the shortest paths for all $l_{(i,j)}^V$ with the bandwidth demands of $b_{(i,j)}^V \cdot s_{(i,j)}^q$ and the path length constraint of $len_{(i,j)}^V$ using Dijkstra;
9: **end for**
10: **if** one or more feasible embedding solutions exist **then**
11: Choose the paths with the maximum InP revenue;
12: $\xi_i \leftarrow 1$, $\chi_{(i,x)} \leftarrow 1$, $r_x^S \leftarrow r_x^S - c_i^V \cdot s_i^q$, $i \leftarrow i + 1$;
13: **if** $i \leq M$ **then**
14: Go to line 3;
15: **else**
16: Finish the embedding of G^V;
17: **end if**
18: **else if** the embedding of n_i^V fails **then**
19: $s_i^q \leftarrow s_i^q - \Delta s^q$;
20: **if** $s_i^q \geq s^q$ **then**
21: Go to line 7;
22: **else**
23: Reject G^V;
24: **end if**
25: **else**
26: $s_{(i,j)}^q \leftarrow s_{(i,j)}^q - \Delta s^q$;
27: **if** $s_{(i,j)}^q \geq s^q$ **then**
28: Go to line 7;
29: **else**
30: $s_{(i,j)}^q \leftarrow 1$, $s_i^q \leftarrow s_i^q - \Delta s^q$, go to line 20;
31: **end if**
32: **end if**

demands of virtual nodes. Node A is given as the access node that should be embedded to OLT firstly. If the residual CPU capacity of OLT cannot afford s^q with the requested amount of CPU resource of A, the VN will be rejected directly. Otherwise, all the nodes that are directly connected with A will be embedded subsequently in the decreasing order of their CPU demands, i.e., first D and then B. Then the nodes connected to D and B will be ranked using the same strategy and so do other nodes. Thus, the order of embedded nodes in VN 1 is $A \to D \to B \to F \to E \to C$. In each embedding stage, the links between the node n_i^V to be embedded and the nodes already embedded will be embedded simultaneously and the feasible solution with the maximum InP revenue will be chosen. In case no feasible solution is available for n_i^V and its related links, the type of failure will be judged. According to the judgment, the QoS satisfaction level will be decreased by Δs^q and the embedding program will be implemented again until the feasible solution is found or the QoS satisfaction level of n_i^V is lower than s^q in which case the VN will be rejected.

Algorithm 2 ABA in FiWi access network

Input: G^S.
Output: $b^S_{(x,y)} \forall x, y \in \{1, 2, \cdots, N\}$.

Initialization: $\psi^S_{(x,y)} \leftarrow 0, flag_{(x,y)} \leftarrow 0, \forall x, y \in \{1, 2, \cdots, n\}$.
while $\sum_{x=1}^{N} \sum_{y=1}^{N} flag_{(x,y)} \neq \sum_{x=1}^{N} \sum_{y=1}^{N} e^S_{(x,y)}$ **do**
  for all $l^S_{(x,y)} \in L^S$, $flag_{(x,y)} = 0$ **do**
    Compute $\psi^S_{(x,y)}$ and find the link $l^S_{(x,y)}$ with the minimum $\psi^S_{(x,y)}$;
  end for
  for all $l^S_{(p,q)} \in D_{(x,y)}$, $\forall p, q \in \{1, 2, \cdots, N\}$, $flag_{(p,q)} = 0$ **do**
    $b^S_{(p,q)} \leftarrow f^S_{(p,q)} + \psi^S_{(x,y)}/|D_{(x,y)}|$, $flag_{(p,q)} \leftarrow 1$;
  end for
end while
if $\sum_{l^S_{(p,q)} \in D_{(x,y)}} f^S_{(p,q)}/b^S_{(p,q)} \leq 1, \forall l^S_{(x,y)} \in L^S$ **then**
  return true; // Reconfiguration of G^S succeeds.
else
  return false; // Reconfiguration of G^S fails.
end if

B. Advanced VNE Algorithm Based on QoS Satisfaction and Network Reconfiguration

As FiWi access network is an integration of wired and wireless subnetworks, the flexibility of substrate network resources offers an opportunity to improve the virtual network acceptance ratio through the resource reallocation in SN. However, it causes high cost (e.g. energy and time consumption) to switch the channels of wireless radios and the bandwidth of each link depends on not only the maximum transmission rate of occupied channel but also the collision domain [11]. Thus, the bandwidth allocation, in which no channel switch is involved, may be more preferably adopted to avoid network congestion. In this paper, an Adaptive Bandwidth Allocation (ABA) algorithm is proposed to reconfigure the substrate network by reallocating the bandwidth when the embedding of VN is congested. One important problem in ABA is the order of allocating links. As in Algorithm 2, the average residual bandwidth capacity $\psi^S_{(x,y)}$ is used to determine the order in case the prior allocated links conflict with the subsequent links. For example in Fig. 2 (a), there are four links with the number in the parenthesis denoting the link load. The collision domains of $l^S_{(a,b)}$ and $l^S_{(c,d)}$ are $D_{(a,b)} = \{l^S_{(a,b)}, l^S_{(c,d)}, l^S_{(e,f)}\}$ and $D_{(c,d)} = \{l^S_{(a,b)}, l^S_{(c,d)}, l^S_{(g,h)}\}$, respectively. We assume $C^R = 10$ such that $\psi^S_{(a,b)} = (10-7)/3 = 1$ and $\psi^S_{(c,d)} = (10-4)/3 = 2$. If $l^S_{(c,d)}$ is first chosen to allocate bandwidth, then we can get $b^S_{(a,b)} = b^S_{(c,d)} = 2 + 2 = 4$ which conflicts with the bandwidth capacity of $l^S_{(c,d)}$ with $4 + 4 + 3 > 10$. Thus, in this case the link $l^S_{(a,b)}$ whose $\psi^S_{(a,b)}$ is lower should be chosen first. The resulting bandwidth allocation is shown as Fig. 2 (b) with the numbers on the links indicating the allocated bandwidth, which are determined by link loads and $\psi^S_{(x,y)}$.

With regard to the online VNs, the embedding results of the prior embedded VNs usually leave an influence on the subsequent ones. Thus, it is necessary to rank the VNs arrived in a specific time window accordingly, which has been considered in the VNE algorithm based on QoS satisfaction, and change the embedding order by releasing and re-embedding

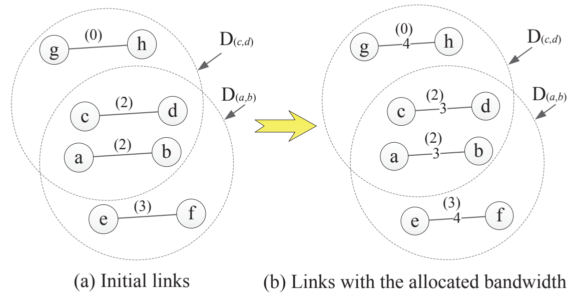

Fig. 2. Illustration of link allocation order in ABA.

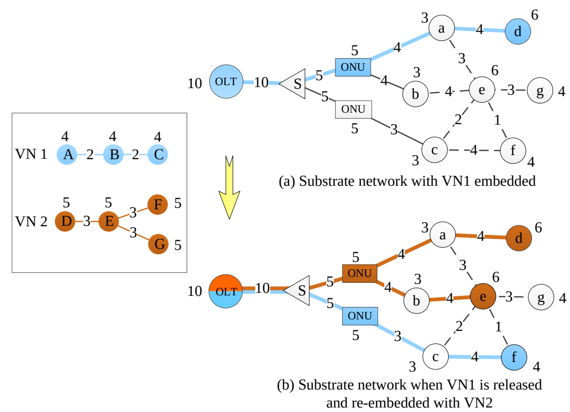

Fig. 3. Illustration of virtual network reconfiguration.

Algorithm 3 Advanced VNE in FiWi access network

Input: $G^V, G^S, \alpha, \beta, \gamma, \rho^n, \rho^l, \Delta s^q, \Delta d$.
Output: $\chi_{(i,x)}, s_i^q, s_{(i,j)}^q, \forall i, j \in \{1, 2, \cdots, M\}$, $x \in \{1, 2, \cdots, N\}$.

1: **Initialization:** $\chi_{(i,x)} \leftarrow 0, \xi_i \leftarrow 0, \forall i \in \{1, 2, \cdots, M\}$ $x \in \{1, 2, \cdots, N\}$, $i \leftarrow 0$;.
2: Rank the virtual nodes in G^V accordingly;
3: Embed n_i^V and its related links on G^S by Algorithm 1;
4: **if** node n_i^V and its related links are successfully embedded **then**
5: $\quad i \leftarrow i + 1$;
6: $\quad$ **if** $i \leq M$ **then** go to line 3;
7: $\quad$ **else** finish the embedding of G^V;
8: $\quad$ **end if**
9: **else if** there exists substrate node with $r_x^S \geq c_i^V \cdot s^q$ **then**
10: $\quad$ **for all** $l_{(x,y)}^S \in L^S$ **do**
11: $\quad\quad$ Compute $p_{(x,y)}^S$ and build the expanded network $G^{S'}$ by replacing $b_{(x,y)}^S$ of G^S with $p_{(x,y)}^S$;
12: $\quad$ **end for**
13: $\quad$ Embed n_i^V and related links on $G^{S'}$ by Algorithm 1;
14: $\quad$ **if** node n_i^V and its related links are successfully embedded **then**
15: $\quad\quad$ Reconfigure the substrate network $G^{S'}$ using ABA;
16: $\quad\quad$ **if** $G^{S'}$ is reconfigured successfully **then**
17: $\quad\quad\quad$ $G^S \leftarrow G^{S'}$;
18: $\quad\quad$ **else**
19: $\quad\quad\quad$ Go to line 23;
20: $\quad\quad$ **end if**
21: $\quad$ **end if**
22: **else**
23: $\quad$ Release the VNs that are already embedded with InP income lower than G^V and residual duration time longer than the given threshold Δd;
24: $\quad$ Embed G^V and all the released VNs by Algorithm 1 with the ABA mechanism in the nonincreasing order of InP income;
25: $\quad$ **if** G^V and all the released VNs are embedded successfully **then**
26: $\quad\quad$ Finish the embedding of G^V ;
27: $\quad$ **else**
28: $\quad\quad$ Recover the released VNs and reject G^V;
29: $\quad$ **end if**
30: **end if**

the existing VNs to accept more subsequent VNs. For example in Fig. 3, the numbers beside nodes and on the links indicate the CPU demands of virtual nodes (or residual CPU capacity of substrate nodes) and the bandwidth demands of virtual links (or residual bandwidth capacity of substrate links). We assume two VNs labeled by VN 1 and VN 2, and VN 1 has already been successfully embedded as shown in Fig. 3 (a). When we attempt to embed VN 2, it is found that there is not any substrate node satisfying the CPU demand of node G (put aside the QoS satisfaction here). However, if we release VN 1 and then embed VN 2 and VN 1 in sequence again, the feasible embedding solution can be found as shown in Fig. 3 (b). Thus, the virtual network reconfiguration can improve the acceptance ratio of VNs by migrating the existing VNs.

In addition, since the bandwidth reallocation in substrate network only changes the bandwidth capacity of substrate links but takes no effect on the CPU capacity of substrate nodes, ABA can only solve link congestion while the node congestion should be avoided by virtual network reconfiguration. The procedure of advanced VNE based on QoS satisfaction and network reconfiguration is shown in Algorithm 3. While the VNE algorithm fails to embed a given virtual node and its related links, the type of congestion (i.e., link or node congestion) would be first checked. For link congestion, the VNE algorithm will be implemented again on the expanded substrate network $G^{S'}$ whose link bandwidth capacity is replaced by the potential bandwidth capacity of each link. If node n_j^V and its related links can be successfully embedded on $G^{S'}$, and the substrate network can be successfully reconfigured by ABA algorithm, then the embedding process is valid. Otherwise, if link congestion cannot be resolved by ABA or the node congestion exists, the virtual network reconfiguration will be executed (in line 23). When it comes to the reconfiguration of VNs, the following tips should be considered. First, we should not migrate the VNs whose duration time is coming into end because their embedded resources will be released soon. Thus, the VNs whose residual duration time is longer than the given threshold Δd can be released and re-embedded. Second, it is obvious that the VNs with higher InP income will be harder to embed for their high resource demands, thus the VNs whose InP income is lower than the VN G^V to be embedded currently can be migrated. After the VNs to be migrated are determined, the VNE embedding algorithm will be executed again. Only if all the released VNs as well as G^V can be embedded successfully, the migration would be implemented on SN. Otherwise, the VN would be rejected and the embedding of all the released VNs would be recovered.

IV. Performance Evaluations

A. Simulation Settings and Scenarios

A simulator is developed for virtualized FiWi access network in VC++ environment. In the substrate FiWi access network, one OLT, 4 ONUs and 80 wireless routers are randomly deployed in a square $500\text{m} \times 500\text{m}$ area. Each ONU drives 2 or 3 wireless gateways and there are 2 or 3 radio interfaces on each wireless router. The CPU capacity of OLT and ONUs is uniformly distributed between 200 and 500, and the CPU capacity of wireless routers is uniformly distributed between 50 and 100. The bandwidth capacity of optical fibers, cables and wireless links is set to 1000Mbps, 54Mbps and 54Mbps, respectively. The arrival of VN requests follows Poisson process with an average rate of 0.05 per time unit. The VN duration is exponentially distributed in an average of 100 time units. The VN size is uniformly distributed between 2 and 10 and s^q is uniformly distributed

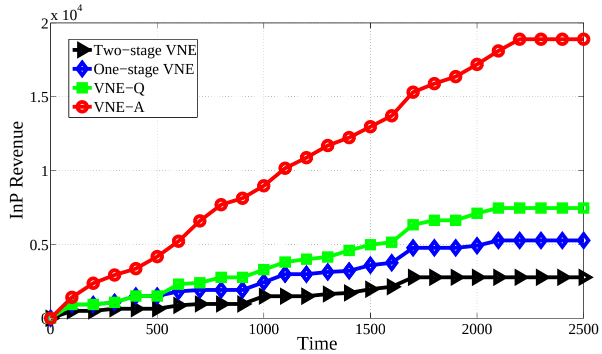

Fig. 4. Comparison of InP revenue over time.

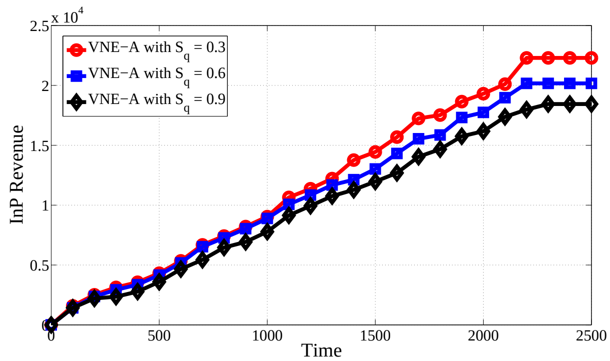

Fig. 5. Comparison of InP revenue under different s^q.

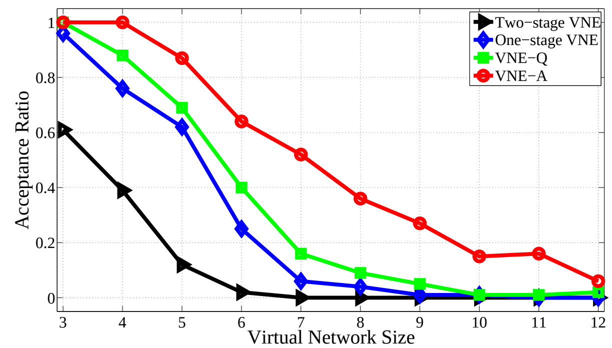

Fig. 6. Comparison of acceptance ratio on different VN sizes.

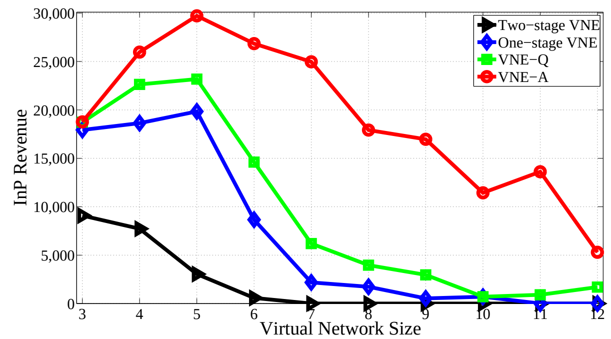

Fig. 7. Comparison of InP revenue on different VN sizes.

between 0.5 and 1. The average connectivity ratio between two virtual nodes is set to 0.5. The CPU demand of each virtual node and the bandwidth demand of each link are uniformly distributed between 10 and 20 and between 1 and 10, respectively. The distance constraint of virtual node is set to 50m with the preferred location uniformly distributed in the network area, and the average path length constraint is set to 5 hops. The simulation is carried out based on the settings of $\alpha = 5, \beta = 1, \gamma = 2, \rho^n = 1, \rho^l = 1, \Delta s^q = 0.1$, and $\Delta d = 50$ time units, and there are 50 time units in each time window. The simulation results are evaluated among the one-stage VNE algorithm which implements the embeddings of node and link simultaneously but ignores the QoS satisfaction, the VNE algorithm based on QoS satisfaction (i.e., VNE-Q) and the advanced VNE algorithm based on QoS satisfaction and network reconfiguration (i.e., VNE-A). We compare these algorithms with the traditional two-stage VNE algorithm [18]–[20] in which the node embedding and link embedding are separated.

B. Simulation Results and Analysis

Figure 4 shows the comparison of InP revenue along with time. It can be observed that the InP revenue grows up with the increase of time because more VNs have been accepted. Particularly, InP revenue of VNE-A is obviously higher than that of two-stage VNE, one-stage VNE and VNE-Q for the reason that in one-stage VNE, each node and its related links are embedded simultaneously such that the defect of seperated link and node embedding in two-stage VNE is avoided. Moreover, in VNE-Q, each VN is equipped with a QoS satisfaction requirement and InP is conferred with the right to provide best-effort service with its residual resource instead of all the required resource to VNs. Thus, the acceptance ratio will increase as well as the InP revenue. In addition, VNE-A results in higher InP revenue through the reconfigurations of substrate network and virtual network compared with VNE-Q.

Figure 5 shows the comparison of InP revenue under different average QoS satisfaction requirements. It can be concluded that lower average QoS satisfaction requirement brings about higher InP revenue because InP can embed VNs with more flexible resource allocation which contributes to its acceptance of more VN requests.

Figures 6 and 7 elaborate the comparisons of acceptance ratio and InP revenue on different VN sizes respectively. As larger size of VN makes it harder for InP to find out the feasible embedding solution, the VN acceptance ratio decreases while the average number of virtual nodes increases from 3 to 12. As for InP revenue, it will first increase discordantly with the decreasing VN acceptance ratio, which results from the higher InP revenue of each VN. However, as the size of VN grows extremely large, the acceptance ratio faces a serious decrease such that the InP revenue declines.

Figures 8 and 9 show the comparisons of acceptance ratio and InP revenue under different average VN arrival rates, respectively. It can be observed that, given the number of VN requests, both acceptance ratio and InP revenue of VNE-A

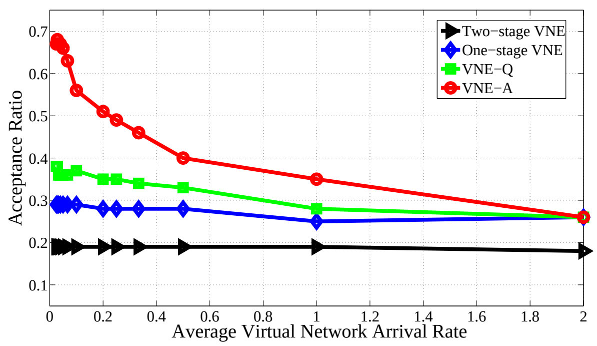

Fig. 8. Comparison of acceptance ratio under different VN arrival rates.

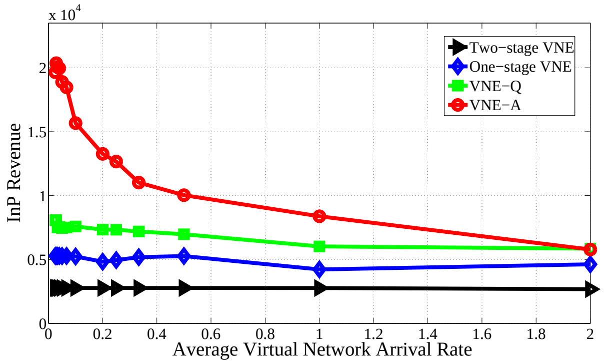

Fig. 9. Comparison of InP revenue under different VN arrival rates.

decrease gradually with the higher average VN arrival rate and go steady finally. However, owing to the adaptive bandwidth allocation, both acceptance ratio and InP revenue of VNE-A are higher than that of other three algorithms especially when the network is low loaded. Specifically, when the average VN arrival rate is 0.033 per time unit, the InP revenue of VNE-A is about 6.2, 2.8, and 1.7 times higher than that of two-stage, one-stage, and VNE-Q respectively. In summary, with the introduction of appropriate QoS satisfaction and network reconfiguration, the proposed VNE-A can achieve larger VN acceptance ratio and InP revenue than other algorithms.

V. Conclusion

In this paper, we studied the virtual network embedding in FiWi access network. With the objective of maximum InP revenue, a VNE algorithm based on QoS satisfaction was first proposed to enable InP to provide the best-effort VN service. Then, a network reconfiguration mechanism was further proposed for the advanced VNE algorithm to improve the VN acceptance ratio by means of the adaptive bandwidth allocation of substrate network and the virtual network reconfiguration. Simulation results suggested that our proposed VNE algorithm based on QoS satisfaction and network reconfiguration leads to higher acceptance ratio and InP revenue than the traditional VNE algorithms. In our future works, we will invest efforts to improve the utilization of substrate network resource and evaluate the performance of traffic transmission such as network delay and throughput.

References

[1] M. Maier and M. Levesque, "Dependable Fiber-Wireless (FiWi) access networks and their role in a sustainable third industrial revolution economy," *IEEE Transactions on Reliability*, vol. 63, no. 2, pp. 386–400, Jun. 2014.

[2] S. Sabino, N. Correia and A. Barradas, "Frequency assignment in multi-channel and multi-radio FiWi access networks," in *Proc. CISTI*, Jun. 2014, pp. 1–6.

[3] F. Aurzada, M. Levesque, M. Maier, and M. Reisslein, "FiWi access networks based on next-generation PON and gigabit-class WLAN technologies: a capacity and delay analysis," *IEEE/ACM Transactions on Networking*, vol. 22, no. 4, pp. 1176–1189, Aug. 2014.

[4] N. Correia, G. Schutz, and A. Barradas, "Correlation-based energy saving approach for smart fiber wireless networks," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 7, no. 6, pp. 525–539, Jun. 2015.

[5] K. Saito, H. Nishiyama, N. Kato, H. Ujikawa, and K. I. Suzuki, "A MPCP-based centralized rate control method for mobile stations in FiWi access networks," *IEEE Wireless Communications Letters*, vol. 4, no. 2, pp. 205–208, Apr. 2015.

[6] J. van de Belt, H. Ahmadi, and L. E. Doyle, "A dynamic embedding algorithm for wireless network virtualization," in *Pro. VTC*, Sep. 2014, pp. 1–6.

[7] A. Tzanakaki, M. P. Anastasopoulos, G. S. Zervas, B. R. Rofoee, R. Nejabati, and D. Simeonidou, "Virtualization of heterogeneous wireless-optical network and it infrastructures in support of cloud and mobile cloud services," *IEEE Communications Magazine*, vol. 51, no. 8, pp. 155–161, Aug. 2013.

[8] Q. Dai, G. Shou,Y. Hu, and Z. Guo, "A general model for hybrid Fiber-Wireless (FiWi) access network virtualization," in *Proc. IEEE ICC*, Jun. 2013, pp. 858–862.

[9] Y. Wang, Q. Hu, and X. Cao, "Connectivity as a service: towards optical-based network virtualization," in *Proc. IEEE ICNC*, Feb. 2014, pp. 264–268.

[10] R. Mijumbi, J. Serrat, J. Gorricho, N. Bouten, F. D. Turck, and R. Boutaba, "Network function virtualization: state-of-the-art and research challenges," *IEEE Communications Surveys Tutorials*, vol. PP, no. 99, pp. 1–1, 2015.

[11] A. Jarray and A. Karmouch, "Decomposition approaches for virtual network embedding with one-shot node and link mapping," *IEEE/ACM Transactions on Networking*, vol. 23, no. 3, pp. 1012–1025, Jun. 2015.

[12] X. Gao, Z. Ye, J. Fan, W. Zhong, Y. Zhao, X. Cao, H. Yu, and C. Qiao, "Virtual network mapping for multicast services with max-min fairness of reliability," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 7, no. 9, pp. 942–951, Sep. 2015.

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

[14] X. Meng, G. Shou, Y. Hu, and Z. Guo, "Efficient load balancing multipath algorithm for fiber-wireless network virtualization," in *Proc. IEEE ICT*, May 2014, pp. 1–6.

[15] S. Su, Z. Zhang, A. X. Liu, X. Cheng, Y. Wang, and X. Zhao, "Energy-aware virtual network embedding," *IEEE/ACM Transactions on Networking*, vol. 22, no. 5, pp. 1607–1620, Oct. 2013.

[16] G. D. Stas, S. Avallone, and R. Canonico, "Virtual network embedding in wireless mesh networks through reconfiguration of channels," in *Proc. IEEE WiMob*, Oct. 2013, pp. 537–544.

[17] S. Avallone, "An energy efficient channel assignment and routing algorithm for multi-radio wireless mesh networks," *Ad Hoc Networks*, vol. 10, no. 6, pp. 1043–1057, Aug. 2012.

[18] M. Yuy, Y. Yiz, J. Rexfordy, and M. Chiang, "Rethinking virtual network embedding: substrate support for path splitting and migration," *ACM SIGCOMM*, vol. 38, no. 2, pp. 17–29, Apr. 2008.

[19] L. Nonde, T. E. H. El-Gorashi, and J. M. H. Elmirghani, "Energy efficient virtual network embedding for cloud networks," *IEEE/OSA Journal of Lightwave Technology*, vol. 33, no. 9, pp. 1828–1849, May 2015.

[20] Q. Chen, Y. Wan, X. Qiu, W. Li, and A. Xiao, "A survivable virtual network embedding scheme based on load balancing and reconfiguration," in *Proc. IEEE NOMS*, May 2014, pp. 1–7.