Computer Communications xxx (2015) xxx–xxx

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

A feedback control approach for energy efficient virtual network embedding☆

Q1 Xiaohua Chen [a,b], Chunzhi Li [a,*], Yunliang Jiang [a]

[a] The School of Information Engineering, Huzhou University, Huzhou, Zhejiang Province 313000, China
[b] The School of Computer Science and Software Engineering, East China Normal University, Shanghai 200062, China

ARTICLE INFO

Article history:
Received 7 January 2015
Revised 17 October 2015
Accepted 24 October 2015
Available online xxx

Keywords:
Network virtualization
Virtual network embedding
Feedback control
Energy efficient
Resource consolidation

ABSTRACT

Network virtualization is an enabler for intelligent energy-aware network deployment. The existing research usually searches the subset of resources in the whole substrate network passively for the virtual networks (VNs), where resource consolidation achieves the minimization of energy consumption by switching off or hibernating as many network nodes and interfaces as possible. However, the stable active resources for accommodating the VNs help enhance the number of the hibernated nodes and links, which can reduce the energy consumption. A novel method for energy efficient virtual network embedding (EEVNE) is proposed in this paper, which controls the mappable area of the substrate network actively and can find the minimal consolidation of the network resources. In our proposed method, a controller, a check device and an actuator are designed for finding the stable consolidated subset of the substrate resources. Besides, two feedback-control-based EEVNE algorithms are devised to minimize the energy consumption of the substrate network for embedding VNs. Simulation results show that our algorithms significantly save greater energy than the existing algorithms.

1. Introduction

The reduction of energy consumption has become a key issue in the future Internet. In 2011, the energy consumption of the Internet amounted 2% of the overall energy consumption approximately [1,2]. And a decrease in greenhouse gases emission volume of 15–30% is required before year 2020 to keep the global temperature increase below 2 °C [3]. Various studies have been started to research the technology of the energy consumption reduction of the Internet. The resources of the Internet are often supplied for the peak load, which are under-utilized in normal operation. The above problems leave a large room for energy savings [4].

Network virtualization is an important technology for the future Internet, the cloud computing and the software-defined networks [5–8]. In the research of the future Internet, some network virtualization platforms, such as PlanetLab [9], Trelis [10], GeENI [11], and GENI [12], combine general-purpose servers with the high performance network processor subsystems. Since the current power consumption of the network equipment is insensitive to the traffic load [13], switching off or hibernating as many network nodes and links as possible without compromising the network performance is the best approach to minimize the energy consumption [14]. Generally, resource consolidation is an enabler for the intelligent energy-aware network deployment.

Virtual network embedding (VNE) is a key technology in the network virtualization environment[8]. The shared substrate network (SN) is managed by the infrastructure providers, while virtual networks (VNs) are created by the service providers. VNs with the constraints on both nodes (e.g., CPU) and links (e.g., bandwidth) are embedded into the same shared SN, which is known as VNE. The energy efficient VNE (EEVNE) reduces the energy consumption of SN when VN requests are embedded into the shared physical SN. In this paper, we consider the EEVNE in the IP network over the Wavelength Division Multiplexing (WDM) optical network. Energy consumption can be reduced by embedding VNs in a smaller set of substrate resources. Some EEVNE literatures have explored to minimize the energy consumption for accommodating VNs in the future Internet [14–20]. The general optimization models (such as mixed integer program, integer linear program) of EEVNE are proposed to reduce the energy consumption. However, the implementation of those models are not scalable for large scenarios. Many heuristic algorithms of EEVNE are proposed, which try to find the consolidated subset of active substrate nodes and links for the VN requests. As those algorithms search the subset in the area of a whole SN, we call those algorithms as

☆ The work of this paper is supported in part by the National Natural Science Foundation of China (No.61501184 and 61370173), and in part by Science and Technology of Zhejiang Province (No.2014C31084).

* Corresponding author. Tel.: +86 13819215406.
E-mail address: lichunzhi82@126.com (C. Li).

http://dx.doi.org/10.1016/j.comcom.2015.10.010
0140-3664/© 2015 Published by Elsevier B.V.

passive EEVNE algorithms. Passive EEVNE algorithms have two outstanding disadvantages, listed as:

- Since the VN requests arrive and depart dynamically over time, VNE brings about the dynamical allocation and recycling of the substrate resources, and the consolidated subset of the active physical nodes and links are easily changed. More substrate resources are switched between the hibernated and active states by the dynamical changes. Hence, the more resources enter into the active state, and the unnecessary energy is consumed.
- Searching the feasible solution in the whole SN may require exploring a high number of available resources, which will lead to unnecessary high processing times. There is a small set of the consolidated resources that can meet the resource requirements of VNs in the light loads. Actively controlling the substrate resources for VNs makes the mappable area smaller, and can help reduce the time cost of VNE.

In this paper, a feedback control based approach is proposed for EEVNE, which can minimize the energy consumption effectively. The controller, actuator, check device and control object are designed for finding the minimal active resources. The controller calculates the minimal number of the hibernated links, which is the parameter of the actuator to set the mappable area for the current VNs. The mappable area of SN is the control object, which determines the energy consumption of SN. Check device is used to check whether or not the virtual nodes and the virtual links are embedded in the current mappable area. If the VN is not accepted successfully, the mappable area will be extended gradually by the feedback control method. The minimal stable consolidated subset of the substrate nodes and links can be found actively in the feedback control process of embedding VNs. In our proposed method, the following technologies and justifications are employed.

- An algorithm is proposed to set the mappable area, in which a mappable flag (such as 1) or an unmappable flag (such as 0) may be set for each link and node of the SN. The mappable flagged area is composed of all the nodes and links with a mappable flag. We search one feasible solution for a VN in the mappable flagged links and nodes of SN.
- If the VNE fails to find the solution in the process of embedding the virtual nodes or links, a feedback control approach will be used to extend the range of the mappable flagged area step by step. The feedback control process will stop if the VN is embedded successfully or no additional resources of substrate links and nodes can be extended to be set a mappable flag.
- The number of the mappable flagged links is a global variable, which is used for the subsequent VNE and can improve the efficiency of finding the stable minimal consolidated subset. The relationships among different VN requests are established by the number of the mappable flagged links.

The distinct difference between the passive and active approaches is how to find the minimization of the consolidated subset. The passive approaches search the feasible solution in the whole SN, while our proposed approach finds the feasible solution in the mappable resources of the SN for VNs. The proposed feedback-control-based approach controls the mappable area actively to find the minimization of the stable consolidated subset. Inspired by the existing heuristic algorithms of EEVNE and cost-based VNE, two feedback control based heuristic algorithms are proposed, which can increase the number of the hibernated nodes and links to reduce the energy consumption significantly.

Our main contributions are listed as follows.

- We present a feedback control approach for EEVNE, which controls the mappable substrate resources for embedding VNs actively. Whatever the substrate resources are frequently allocated and recycled, the subset of the substrate resources will always remain stable in the hibernated or active states.
- Due to the NP-hardness of the exact VNE algorithms [21], two feedback control based heuristic EEVNE algorithms are presented to find the minimal stable consolidated subset of the SN actively, where the energy consumption of the SN can be reduced greatly. Compared to the prior arts, our algorithms PR-FB (based on the page rank and feedback control approach) and EA-FB (based on the energy-aware VNE and feedback control approach) save up to 56% and 60% of energy consumptions, respectively.
- Extensive simulations are done to evaluate the performances of our algorithms. The simulation results demonstrate that our algorithms perform better than the existing heuristic algorithms.

The remainder of this paper is organized as follows. In Section 2, we introduce the related work. In Section 3, the VNE problem and the power consumption model are introduced. We propose a feedback control approach and two algorithms of EEVNE in Section 4. In Section 5, we detail the performance evaluations of the solutions and their comparisons with the heuristic solutions. Finally, we conclude this paper in Section 6.

2. Related work

VNE is a key technology in the network virtualization [8], and EEVNE is a hot research field in VNE. Various exact EEVNE models have been proposed. Botero *et al.* propose a mixed integer program (MIP) to minimize the active substrate links and nodes, which provides optimal EEVNE [14]. The MIP formulates the resource consolidation and provides an optimal bound to evaluate the heuristics solutions. However, they only consider the number of the activated nodes and links instead of considering the energy consumption. In [15], the energy consumption minimization of the SN and load balancing are introduced, which result in better embeddings with regard to energy savings and acceptance ratio. Su *et al.* propose an integer linear program (ILP), which minimizes the number of working nodes and intermediate nodes [16]. Rosario *et al.* [17] propose a general optimization model for the load-sensitive equipment. However, those MIP and ILP apply exact algorithms (such as GLPK), which lead to too high time complexity and inapplicability for online VNE.

Since the exact EEVNE models (such as MIP and ILP) are NP-hard problems, some heuristic approaches are proposed. Botero and Hesselbach [15] propose an heuristic approach to reconfigure the allocation of embedding VNs. The relocation implies the migration of the virtual nodes and links from one place to another, which produces the undesirable effects on the virtual routers and links migration for the QoS of working transmissions. Su *et al.* devise an EEVNE algorithm by using the consolidation technique [16]. However, the heavy weight on the difference of remaining CPU of the substrate nodes and requested CPU of virtual nodes is given to the node rank, and more energy is consumed. They observe that the electricity price varies over both location and time, and propose two EEVNE algorithms: a heuristic algorithm and a particle-swarm-optimization-based algorithm [18], which reduce the energy cost. Wang *et al.* [19] minimize the energy consumption by coordinating the power-aware node mapping and the power-aware link mapping. However, the dynamical characteristics are not considered, and the energy is wasted. Chang *et al.* [20] propose an ant colony optimization based algorithm to minimize the energy consumption. However, the energy consumption depends on the population of ants and the number of iterations. In [22], a minimal cost flow based EEVNE model for path splitting is proposed to reduce the energy consumption. For the cloud data centers, the EEVNE algorithms are proposed to reduce the energy consumption [23–27]. Tarutani *et al.* [23] propose a method that immediately reconfigures the VN, which reduces the energy consumption with the constraints on the bandwidth and delay among the servers (based on optical communication paths in data center networks). Wang *et al.* [24] propose an unified framework by exploring the communication patterns of the applications and the regularities of the network topology, and devise two efficient algorithms to save energy for the green data center networks. Nonde *et al.* [25] consider EEVNE in IP over O-OFDM cloud networks. Guan *et al.* [26] develop a heuristic EEVNE algorithm, where demands of VNs change over time and are predictable. Nguyen *et al.* [27] assume an environment-aware paradigm for the virtual slices that allows improving energy efficient and dealing with intermittent renewable power sources. Besides, they formulate an optimal solution for the virtual slice assignment problem.

Most of the above technologies on the load-insensitive equipment for the future Internet research try to find the minimization of the consolidated subset for the VNs passively. Minimization of the energy cost is added to the EEVNE [18]. However, the founded subset of the active resources are easily interfered by the dynamical characteristics of VNE, and the unnecessary energy consumption is produced. In this paper, we propose a feedback control approach for EEVNE, which can find the stable consolidated subset for the VNs by controlling the mappable resources of the SN actively. Two heuristic EEVNE algorithms are proposed, where the stable minimal consolidated subset of the substrate resources can be found gradually by our proposed feedback control approach.

3. Network model and problem description

3.1. Virtual network embedding problem

Nowadays, network virtualization can be thought as an inherent component of the future Internet architecture [28]. In this paper, we consider the EEVNE in the future Internet architecture, which can also be extended to the cloud data centers. The application of network virtualization brings about some issues, such as, how to model VN and SN and how to embed the virtual resources into the shared physical resources effectively. Typically, a VN/SN is composed of the network elements (nodes and links), where the nodes are interconnected through the links. For simplicity, we consider the network resources are homogeneous with regard to their energy consumption, and the SN is reduced to just one ISP segment (access, transport or core), where the network equipments share the similar characteristics. The VN and SN are modeled by the following descriptions.

SN. Let $G^s = (N^s, L^s, C_N^s, C_L^s)$ be a SN, where N^s is the set of nodes and L^s is the set of links. C_N^s and C_L^s are the attributes of the substrate nodes and links, respectively. We consider the available CPU as the node attribute, and the available bandwidth as the link attribute.

VN. Let $G^v = (N^v, L^v, C_N^v, C_L^v)$ be a VN, where N^v is the set of the virtual nodes and L^v is the set of the virtual links. C_N^v and C_L^v are the attributes of the virtual nodes and links, correspondingly. Be consistent with the SN, the CPU and bandwidth are the C_N^v of the node N^v and C_L^v of the link L^v.

The VNE solves the efficient mapping of the VN requests to the nodes (routers or switches with the virtualization capabilities) and links of the SN with the constraints of processing power and bandwidth demands. VNE can be naturally decomposed into two components: *node mapping* and *link mapping*. A virtual node can be hosted by any available substrate node with the sufficient processing power resources, furthermore, a single substrate node can host several virtual nodes. A virtual link can span several links with the sufficient bandwidth resources in the SN, and each substrate link may be a part of the several virtual links. In this paper, we consider that different nodes of an identical VN cannot be embedded in the same substrate node, and a single virtual link can be embedded in a path in the SN. The VNE is defined by the following descriptions.

VNE. The VNE is defined as mapping M^v, which is from G^v to the subset of G^s, i.e., $M^v : (N^v, L^v) \longrightarrow (N_r^s, P_r^s)$, where $N_r^s \subset N^s, P_r^s \subset P^s$ and P^s denotes the set of all the loop-free paths of the SN.

Many VNE approaches [8] try to realize the minimization of embedding cost, link bandwidth and processing power. Currently, EEVNE is an important issue in the research of VNE, which searches the minimization of the energy consumption. In this paper, we propose a feedback control method of EEVNE, where the VNs are embedded in the consolidated subset actively. For minimizing the overall network consumption, the unused interfaces and nodes can be deactivated by switching them off. If a link is switched off, the pair of the interfaces on its ends are switched off and energy is saved.

3.2. Energy consumption modeling

An accurate energy consumption model is essential to evaluate the energy savings of the green solutions. Yet obtaining energy consumption figures for the real network infrastructures is a very challenging task (Because the inconsistency of the different models further become quickly out-of-date) [29]. We assume that the energy management strategy can be integrated in the network management platforms. These platforms allow the centralized and remote control of all the devices and the changes of their configuration (change of routing settings, switch between the active/sleep models).

Be similar to the previous work in [30], we define the energy consumption of the physical nodes and links. The *i*th node energy consumption PN^i can be calculated as

$$PN^i = \begin{cases} P_b + P_l \cdot \mu^i, & \text{if node } i \text{ is } \textit{active} \\ 0, & \text{otherwise} \end{cases}, \tag{1}$$

where P_b is the chassis baseline power, P_m denotes the total power which comes into being at the maximum capacity, $P_l = P_m - P_b$ represents the energy proportion factor, and μ^i denotes CPU utilization of the *i*th node. When the node is powered off or in the hibernated state, the energy consumption of the node is 0.

Four IP-Over-WDM transport network architectures are introduced in [31], where the energy consumption models are different. Several components, such as receiving/transmitting equipment, electronic traffic processing, 3R-regenerators, electronic or optical switching devices, optical amplifiers and network control, can give contributions to the total power consumed by an optical transport network. There are two possible ways to implement the IP-Over-WDM networks, i.e., lightpath *non-bypass* and *bypass* [32]. Under lightpath non-bypass, both switching and grooming of traffic are accomplished in the electronic domain. The lightpath bypass approach allows IP traffic, whose destination is not the intermediate node, to directly bypass the intermediate router via a cut-through lightpath. Hence, we define the energy consumption PL^j of the physical link j under lightpath non-bypass as Eq. 2.

$$PL^j = \begin{cases} 2P^{card} + \left(\left\lceil \frac{J_j}{P^s} - 1 \right\rceil + 2\right) P^{CA}, & \text{if link } j \text{ is powered on} \\ 0, & \text{otherwise} \end{cases}, \tag{2}$$

where P^{card} is the energy consumption of the line card in each endpoint of the link, J_j is the length of link j, P^s is the span distance which determines every P^s kilometers (km) a new amplifier required to properly propagate the signal, and P^{CA} is the energy consumption of each amplifier. Typically, erbium doped fiber amplifiers (EDFAs) are deployed on each fiber of physical link, and the value

Please cite this article as: X. Chen et al., A feedback control approach for energy efficient virtual network embedding, Computer Communications (2015), http://dx.doi.org/10.1016/j.comcom.2015.10.010

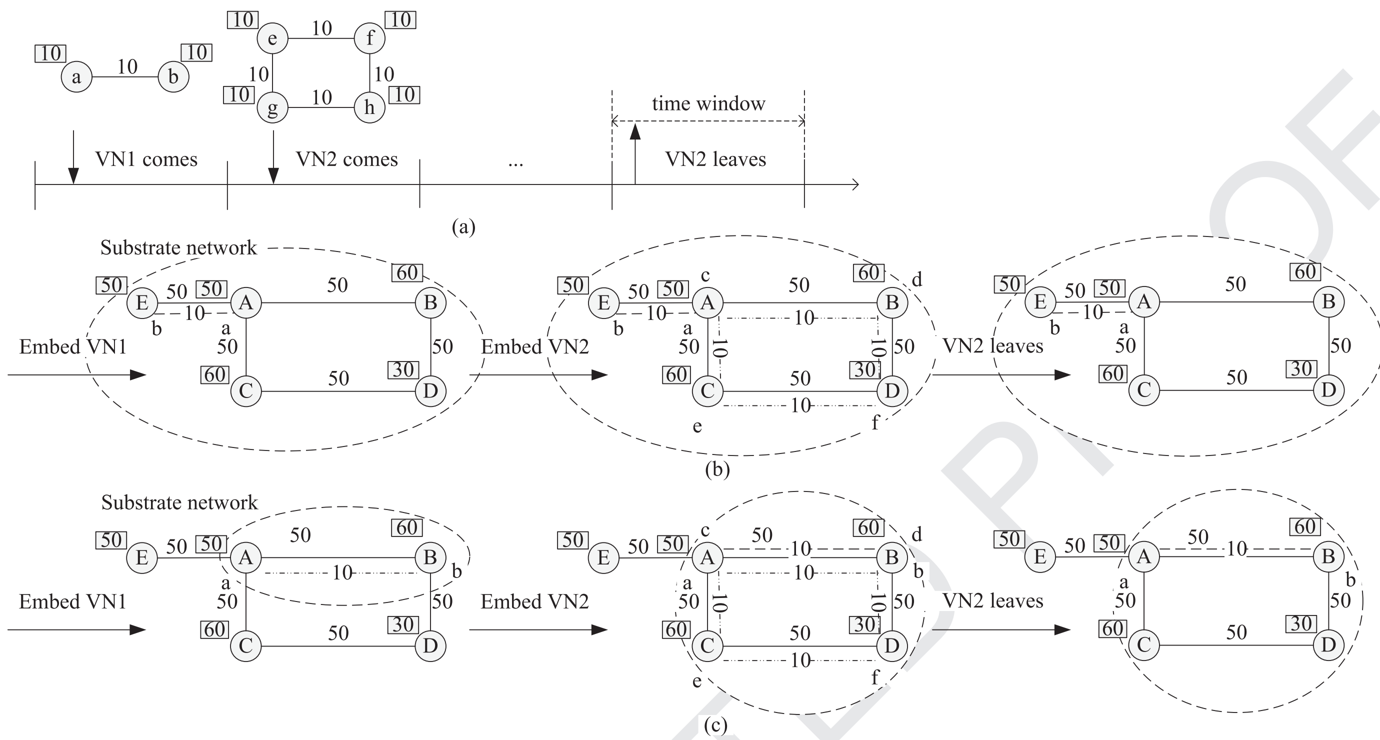


Fig. 1. The benefit of the feedback-control-based VNE.

of P^s is 80 [32]. In this paper, it is assumed that the links have a single fiber. Specially, $\lceil \frac{J_j}{P^s} - 1 \rceil + 2$ is the number of in-line EDFAs required on the link, and "2" counts a post-amplifier and pre-amplifier respectively at the two ends of a fiber link [32,33]. When the link is powered off or in the hibernated state, the energy consumption of the link is 0. We assume that the energy consumption of a dedicated offload engine will be widely deployed in network virtualization [9,34,35], where such engine can sustain high packet processing rates and incur low processing latency. It is nearly a constant regardless whether the ports are idle or carrying full speed traffic. Although the link rate has an influence on the energy consumption of the interfaces (such as the number of ports that are used to aggregate the data traffic), we apply the P^{card} in this model to reduce the influence.

In the transparent/translucent IP-Over-WDM, the traffic flows can bypass IP routers. Associated with each wavelength, a pair of transponders are connected for data transmission. Due to the OEO processing capability of each transponder, full wavelength conversion can be ensured. Referring to the literature [32], we define the energy consumption PL^j of the physical link j under lightpath bypass as Eq. 3.

$$PL^j = \begin{cases} 2P^{card} + 2P^{tr} + (\lceil \frac{J_j}{P^s} - 1 \rceil + 2)P^{CA}, & \text{if link } j \text{ is powered on} \\ 0, & \text{otherwise} \end{cases} , \quad (3)$$

where P^{tr} is the energy consumption per transponder.

4. Feedback control approach and algorithms of EEVNE

4.1. An example

In this section, a feedback control method of EEVNE is proposed to get the stable consolidated subset. Fig. 1 illustrates the dynamical process and the benefit of feedback-control-based VNE, where the mappable area of SN is surrounded by the dashed oval. The area is controlled actively for the VNs, which is named as the mappable area/subset.

Fig. 1(a) describes the dynamic process of the VN requests, including their arrival and department. In VN1, each node requires 10 units of CPU and each link requires 10 units of bandwidth at one time window. At the next time window, VN2 with 4 nodes and 4 links comes for requesting the resources (each node requires 10 units of CPU, and each link requires 10 units of bandwidth). VN2 departs after running some time. Fig. 1(b) tries to find the minimization of the consolidated resources in the whole SN, which are included in the dashed oval for VN1 and VN2. Because the dynamical characteristics are not considered, the nodes of VN1 are mapped to the substrate nodes A and E based on the resource consolidation firstly. Then the nodes of VN2 are mapped to the substrate nodes A, B, C and D. In this case, all the substrate nodes and links get into the active state. Our proposed approach actively controls the mappable area in the dashed oval for VN requests, shown in Fig. 1(c). When VN1 comes for requesting the resources, we design one algorithm to search the small subset (i.e., the mappable subset/area) of the substrate resources firstly. Then we apply the existing VNE algorithms to embed VN1 in the mappable subset. If the mappable subset cannot be enough to embed VN, the feedback control approach is proposed to extend the subset. Repeat the steps until VN1 is embedded successfully or the subset cannot be extended further. Considering the dynamical characteristics, we actively control the mappable area with more resources to embed the VNs in the minimal active resources. Therefore, a and b of VN1 are embedded in A and B, and the link (a,b) is embedded in (A,B). The mappable subset, i.e., two nodes (A and B) and a link (A,B) in the dashed oval, can be used for the VN2, which can reduce the running time of VNE. Since the subset in embedding VN1 is not enough to embed VN2, we repeat the above steps to extend the area. In Fig. 1(c), since the nodes of VN1 are embedded in A and B with more resources, the chances of embedding VN2 in the small range of the active substrate resources are increased. After accepting VN2, only four nodes and links of the SN are active. Compared to Fig. 1(b), energy is saved in Fig. 1(c). Besides, this phenomenon will occur again after the leaving of VN2.

Please cite this article as: X. Chen et al., A feedback control approach for energy efficient virtual network embedding, Computer Communications (2015), http://dx.doi.org/10.1016/j.comcom.2015.10.010

4.2. Feedback control EEVNE approach

From the above example, we find that the active EEVNE approach encounters two important problems, i.e., how to find the mappable area and how to build a relation between VNE and the mappable area. In this section, an algorithm and a feedback control approach are proposed to solve those problems.

The mappable flags for the substrate nodes and links are set by Algorithm 1, where the mappable area is consists of the nodes and links with the mappable flags. For example, the mappable area (such as nodes A, B and the link (A,B)) is surrounded by the dashed oval shown in Fig. 1(c). The mappable flags are set for all the nodes and links in the mappable area, and the unmappable flags are set for the nodes and links outside the mappable area.

Algorithm 1 sets the mappable flags for all links and nodes at the beginning (**Step 4**). For finding the minimization of the mappable area, the connectivity and the abundant resources in the mappable area must be ensured, which can improve the probability of accepting VNs and reduce the energy consumption of SN. The degree $d(k)$ of the substrate node k is calculated (**Step 2 and 3**), where the degree $d(k)$ is the sum of the links that are connected to the node k. For example, $d(E) = 1$, $d(B) = d(C) = d(D) = 2$, and $d(A) = 3$ in Fig. 1. The mappable flagged node u with the minimum degree is selected at each loop (**Step 6**), and an unmappable flag is set for the link l_{uv} that is connected the node u. If $d(v) == 0$, the node v will be set an unmappable flag. These steps can guarantee the connection and more resources in the mappable area. For example, the node E with minimal degree is selected firstly, and the link (E, A) is set the unmappable flag in Fig. 1. The value of $nosleep^l$ is the number of links with the mappable flags. $linkSum$ is the number of substrate links. $sleep^l = linkSum - nosleep^l$, which is the number of links with unmappable flags. If $sln >= sleep^l$, the process of setting mappable flags for nodes and links will stop (**Step 5**). The set of nodes and links with the mappable/unmappable flags will be returned. The time complexity of Algorithm 1 is $O(linkSum \cdot nodeSum \cdot nosleep^l)$, where $linkSum$, $nodeSum$ and $nosleep^l$ are the number of substrate links, nodes and mappable links, respectively.

Accommodating the VNs is an essential prerequisite for reducing the energy consumption of SN. The value "true" in Fig. 2 is the expectation of EEVNE. Every VN request hopes to get the required resources. For reducing the energy consumption, the consolidated subset should be minimized under the condition of accommodating the VNs. We use the feedback control to build the relation between the consolidated subset and EEVNE. The feedback control system includes several control variables and a feedback control loop, which continuously monitor the system behaviour. In this paper, a novel feedback control approach is developed to search the minimization of the stable consolidated subset of the substrate resources for VNs (shown in Fig. 2). When any virtual node or link of one VN request fails to be embedded, the feedback control will be used to expend the mappable area, so that the probability of accepting the VN request can be improved in the next loop. The feedback control approach includes four parts: controller, actuator, check device and control object, which collaborate with each other to complete EEVNE jointly.

Control object. The mappable area of SN is the control object. According to the minimal principle, the actuator controls the range of the substrate resources for the ith VN. If any virtual node/link fails to be embedded in one loop, the mappable area will be extended step by step for the next loop.

Check device. Check device includes $NodeEm(i)$ and $LinkEm(i)$, which are the node mapping and link mapping for the ith VN (shown in Fig. 2). It is worth noting that $NodeEm(i)$ and $LinkEm(i)$ are only embedded in the mappable area of the control object. If one VN is embedded successfully, $NodeEm(i)$ or $LinkEm(i)$ will return the value *true*, which means the minimal consolidated subset of the SN is found for current VN.

Algorithm 1
Set mappable/unmappable flags for nodes and links

Input: The number of mappable flagged links $nosleep^l$,
Output: The set of nodes and links with mappable/unmappable flags.

1: $sln = 0$; $sleep^l = linkSum - nosleep^l$;
2: **foreach**(each node k of the substrate network)
3: Calculate node degree $d(k)$;
4: Set a mappable flag for each node and link of substrate network.
5: **while**($sln < sleep^l$) {
6: **if**(find a mappable node u with minimum $d(k)$){
7: **foreach**(each link l_{uv} that is connected to u){
8: Set l_{uv} as a unmappable flag; $sln++$;
9: $d(u)--$; $d(v)--$;
10: **if** ($d(u) == 0$) set u as an unmappable flag;
11: **if** ($d(v) == 0$) set v as an unmappable flag;
12: **if** ($sln >= sleep^l$) **break**;
13: }
14: } else { **break**; }
15: }
16: **return** the set of mappable flagged nodes and links.

Controller. The function of the controller is to calculate the number $LNum$ of the mappable substrate links for current VN. According to the results of $NodeEm(i)$ and $LinkEm(i)$, the value *true* is expected, and the global variable $LNum$ is saved for the next VNE. The em is a boolean variable, which is calculated by $em = NodeEm(i)$ and $em = LinkEm(i)$. If $em == false$, the node or link of ith VN fails to be embedded. $LNum$ is defined as

$$LNum = LNum + 1 \text{~~if}(em == false) \tag{4}$$

With the increasing $LNum$, the mappable area is expanded gradually. Finally, the minimal consolidation of the substrate resources can be found. The parameter $LNum$ in Eq. 4 determines the mappable area of the control object. We try to replace Eq. 4 by $LNum = LNum + STEP$, where $STEP$ is a constant. The large value of $STEP$ can reduce the loop times, but it affects the energy consumption. Specially, when the value of $STEP$ is close to the value of $linkSum$, our algorithms degenerate into the original algorithms. Therefore, $STEP$ is set to be 1.

Actuator. The actuator sets the mappable flags for the substrate nodes and links, which is realized by Algorithm 1. The VNs will be only embedded in the returned mappable area of the control object.

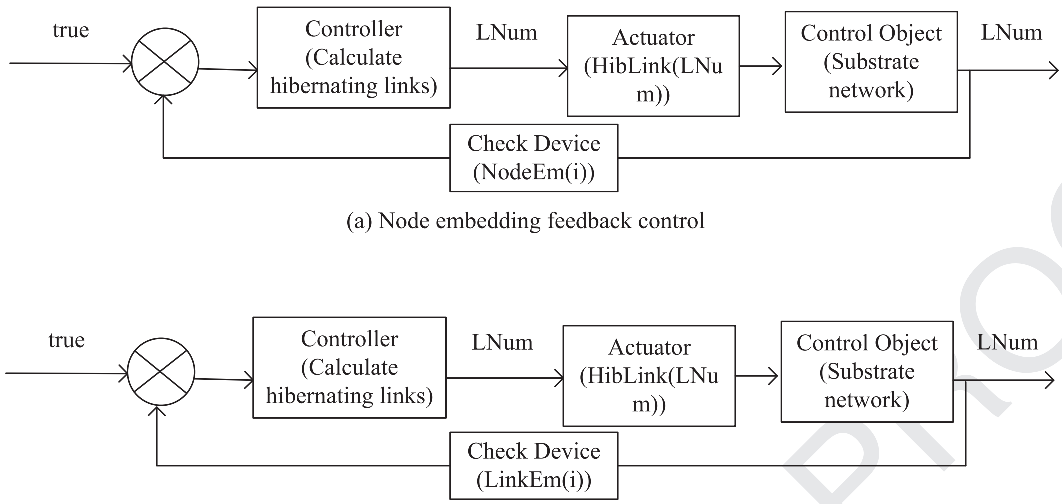
Using feedback control theory, the proposed approach aims to get the minimization of the energy consumption by shrinking the size of the mappable area.

4.3. Feedback Control EEVNE Algorithms

The feedback control approach gives the rough guide to embed the VNs actively in the mappable area. Based on the feedback control VNE approach, Algorithm 2 is proposed for EEVNE, which is a two-stage VNE algorithm. We try to reduce the energy consumption by searching the minimum consolidated subset of the SN for embedding a VN. The algorithm consists of two parts: node mapping and link mapping. Algorithm 1 is used to control the mappable area, where $NodeEm(i)$ and $LinkEm(i)$ allocate the resources in the mappable area. If the required resources of the ith request is satisfied, the result of VNE will be returned. $LNum$ is a global variable. When current VN is accepted, the value of $LNum$ is utilized for next VN, which can reduce the times of feedback and the running time for VNE. The variable $LNum$ is initialized to 1, and $linkSum$ is the maximum value of $LNum$. If one virtual node or link is not accepted, $LNum++$ and Algorithm 1 will be executed again to extend the mappable area.

Algorithm 2 tries to reduce the energy consumption by finding the minimization of the consolidated subset. After node mapping is completed successfully, the link embedding is started (**Step 12**). $NodeEm(i)$ and $LinkEm(i)$ can be realized by the current existing algorithms. In this paper, we propose two feedback-control-based

Please cite this article as: X. Chen et al., A feedback control approach for energy efficient virtual network embedding, Computer Communications (2015), http://dx.doi.org/10.1016/j.comcom.2015.10.010



(a) Node embedding feedback control

(b) Link embedding feedback control

Fig. 2. A Feedback Control Approach for Virtual Node and Link Embedding.

Algorithm 2
Feedback-control-based EEVNE algorithm

Input: The ith VN request.
Output: The result of VNE.

```
Get LNum from the previous VNE.
while(1) {
  Get the mappable area by calling Algorithm 1, where LNum is an
   input parameter,
  if((em = NodeEm(i))==true){
    Set NODE_SUCC flag for the ith VN request;
    break;
  }
  LNum++;
  if (LNum > linkSum) return NODE_FAILED;
}
while(1) {
  if((em = LinkEm(i)) == true){
    Set LINK_SUCC flag for the ith VN request;
    return VNE solution;
  }
  LNum++;
  if (LNum > linkSum) return LINK_FAILED;
  Release the embedded resources of nodes and links.
  Goto 3;
}
```

algorithms (i.e., PR-FB and EA-FB) to heighten the effectiveness of the feedback control approach. In PR-FB, $NodeEm(i)$ and $LinkEm(i)$ use the RW-MaxMatch node mapping algorithm and the RW-MaxMatch link mapping algorithm in [36] respectively. In EA-FB, $NodeEm(i)$ and $LinkEm(i)$ use the energy-aware node mapping algorithm and the energy-aware link mapping algorithm in [16] respectively. The biggest difference between PR-FB and EA-FB is the method of ranking the nodes. In PR-FB, the rank of a node i is determined by its CPU and its collective bandwidth of outgoing links. Let $H(i) = CPU(i) \cdot \sum_{l \in L(i)} BW(l)$, where $CPU(i)$ is the remaining CPU of the substrate node i, $L(i)$ is the set of all the links of i and $BW(l)$ is the unoccupied bandwidth of the link l. The initial NodeRank value for node u is computed by $NR_u^{(0)} = \frac{H(u)}{\sum_{v \in N^s} H(v)}$. Let $p_{uv}^J = \frac{H(v)}{\sum_{w \in N^s} H(w)}$, and $p_{uv}^F = \frac{H(v)}{\sum_{w \in nbr_1(u)} H(w)}$, where $nbr_1(u) = \{v | (u, v) \in L^s\}$. For any node u, let $NR_v^{(t+1)} = \sum_{u \in N^s} p_{uv}^J \cdot p_u^J \cdot NR_u^{(t)} + \sum_{u \in nbr_1(v)} p_{uv}^F \cdot p_u^F \cdot NR_u^{(t)}$, where $p_u^J + p_u^F = 1$, $p_u^J \geq 0$, $p_u^F \geq 0$, and $t = 0, 1, \ldots$. Let $NR^{(t+1)} = T \cdot NR^{(t)}$, where $NR^{(t)} = (NR_1^{(t)}, NR_2^{(t)}, \ldots, NR_n^{(t)})$, n is the number of the nodes, and T is a one-step transition matrix of the Markov chain, defined by

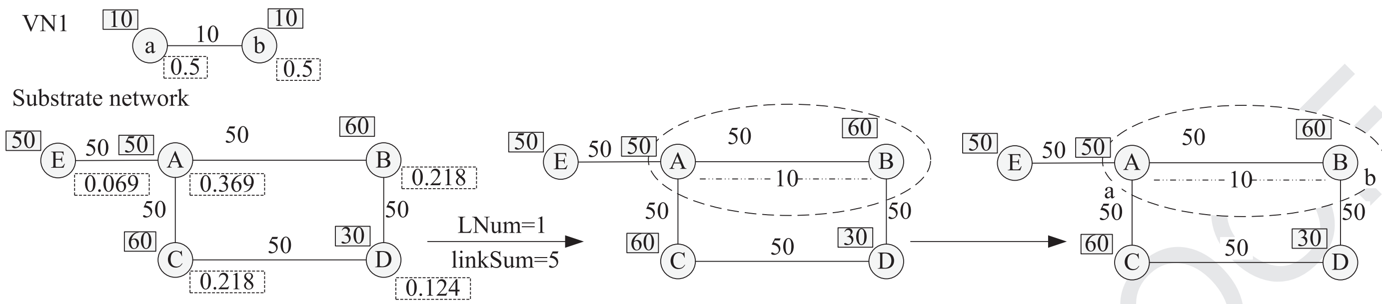
$$T = \begin{pmatrix} p_{11}^J & p_{12}^J & \ldots & p_{1n}^J \\ p_{21}^J & p_{22}^J & \ldots & p_{2n}^J \\ \ldots & \ldots & \ldots & \ldots \\ p_{n1}^J & p_{n2}^J & \ldots & p_{nn}^J \end{pmatrix} \cdot \begin{pmatrix} p_1^J & 0 & \ldots & 0 \\ 0 & p_2^J & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & \ldots & p_n^J \end{pmatrix} + \begin{pmatrix} 0 & p_{12}^F & \ldots & p_{1n}^F \\ p_{21}^F & 0 & \ldots & p_{2n}^F \\ \ldots & \ldots & \ldots & \ldots \\ p_{n1}^F & p_{n2}^F & \ldots & 0 \end{pmatrix} \cdot \begin{pmatrix} p_1^F & 0 & \ldots & 0 \\ 0 & p_2^F & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & \ldots & p_n^F \end{pmatrix}. \quad (5)$$

Finally, a positive value ϵ is given, and the node rank $NR(i) = NR_i^{(t)}$ can be computed using the iterative scheme [36], in which $\|NR^{(t+1)} - NR^{(t)}\| < \epsilon$ is used to end the loop. RW-MaxMatch node mapping procedure firstly computes the NodeRank values of all the nodes in N^s and N^v, and then map virtual nodes to the substrate nodes using the L2S2 mapping (which stands for "large-to-large and small-to-small" mapping) procedure. RW-MaxMatch link mapping procedure maps the virtual links using the k-shortest path algorithm. In EA-FB, the substrate node i is ranked as follows,

$$NR(i) = \alpha \cdot NR_{cpu}(i) + (1 - \alpha) \cdot NR_{bw}(i), \quad (6)$$

where $NR_{cpu}(i) = CPU(i) - CPU(j)$, $CPU(j)$ is the required CPU of the virtual node j, and $\alpha = 0.5$. $NR_{bw}(i)$ is the NodeRank for the substrate node i, which is computed by the above measure of PR-FB. In EA-FB, $H(i) = \sum_{l \in L(i)} BW(l)$, which is different from PR-FB. EA-FB maps the virtual nodes to the substrate node with the highest NodeRank, and uses the shortest-path-based algorithm for the link mapping, which selects the active nodes firstly before turning inactive nodes to active. We only consider the unsplittable path to map a virtual link in PR-FB and EA-FB. The time complexity of PR-FB and EA-FB depend on $NodeEm(i)$ and $LinkEm(i)$.

A typical example is used to illustrate our algorithms, where VN1 is embedded in SN in Fig. 1. Based on the two-stage algorithm in [36], the process of PR-FB is shown in Fig. 3. In the first step, the NodeRank values of all the nodes in SN and VN are computed in the light of the method [36], which are the numbers in the dotted rectangles in Fig. 3(a). The node rank measures the resource availability of a node. The rank of a given node u is not only determined by its CPU power and its collective bandwidth of outgoing links, but also affected by the ranks of the nodes that can be reached from u. The node rank is computed by the method in [36]. Then we use Algorithm 1 to get the



(a) Compute the NodeRank values of the nodes in SN and VN [44]

(b) Get the mappable area using Algorithm 1

(c) Embed VN1 using RW-MaxMatch [44] in the mappable area

Fig. 3. An example of the feedback-control-based EEVNE.

mappable area, which is surrounded in the dashed oval of Fig. 3(b), where $LNum = 1$ and $linkSum = 5$. Finally, VN1 is embedded in the mappable area by using the RW-MaxMatch in Fig. 3(c). The $LNum$ is a global variable, which can be used for the next VN. EA-FB computes the NodeRank of the substrate nodes by Formula 6, which is different from PR-FB. In EA-FB, the NodeRank values of A, B, C, D and E are approximately 0.316, 0.209, 0.209, 0.183, 0.081, respectively.

5. Performance evaluation

5.1. Saturated scenarios, non-saturated scenarios and performance metrics

To compare the different VNE algorithms, the non-saturated and saturated scenarios are defined. In the non-saturated scenarios, the resources of the SN are enough to accept all VNs. Conversely, in the saturated scenarios, the resources are not enough to accept all VNs.

The metrics to measure performances in these experiments are energy consumption, the number of the hibernated nodes and links, acceptance ratio, revenue, and revenue over cost.

- The long-term average energy consumption of SN in one time unit is given by $\lim_{T->\infty} \frac{\sum_{t=1}^{t=T} (\sum_{i\in N^s} PN^i(t) + \sum_{j\in L^s} PL^j(t))}{T \cdot T_n}$, where T is one time window and T_n is the time unit in a time window. $PN^i(t)$ and $PL^j(t)$ correspond to the energy consumptions of substrate node i and link j in the time window t.
- The long-term average number of the hibernated substrate nodes in a time window is given by $\lim_{T->\infty} \frac{\sum_{t=1}^{t=T} (\sum_{i\in N^s} HibNode^i(t))}{T}$, where $HibNode^i(t)$ is a binary variable. If the node i is in the hibernated state at the time window t, $HiberNode^i(t) = 1$. Otherwise, $HiberNode^i(t) = 0$.
- The long-term average number of the hibernated substrate links in a time window is given by $\lim_{T->\infty} \frac{\sum_{t=1}^{t=T} (\sum_{j\in L^s} HibLink^j)}{T}$, where $HibLink^j(t)$ is a binary variable. If the link j is in the hibernated state at the time window t, $HiberLink^j(t) = 1$. Otherwise, $HiberLink^j(t) = 0$.
- The long-term average acceptance ratio is given by $\lim_{T->\infty} \frac{AccVN(T)}{SumVN(T)}$, where $AccVN(T)$ and $SumVN(T)$ correspond to the number of the accepted VNs and all VNs in the time window T.
- The long-term average revenue in a time window is given by $\lim_{T->\infty} \frac{\sum_{t=1}^{t=T} (Rev(t))}{T}$, where $Rev(t)$ corresponds to the revenue of accepting the VNs at time t. $Rev(t) = \sum_{p\in AccS(t)} AccRev(p)$, where $AccS(t)$ is the set of the accepted VNs at the time window t and $AccRev(p)$ is the revenue of the pth VN. $AccRev(p) = \sum_{i\in N^v(p)} CPU^i + \sum_{j\in L^v(p)} BW^j$, where CPU^i, BW^j, $N^v(p)$ and $L^v(p)$ are the CPU of virtual node i, bandwidth of virtual link j, the set of virtual nodes and virtual links of the pth VN, respectively.
- The long-term average revenue over cost is given by $\lim_{T->\infty} \sum_{t=1}^{t=T} \frac{Rev(t)}{Cost(t)}$, where $Cost(t)$ is the cost of accepting the VNs at the time window t. $Cost(t) = \sum_{p\in AccS(t)} Cost(p)$, where $Cost(p)$ is the cost of accepting the pth VN. $Cost(p) = \sum_{i\in N^v(p)} CPU^i + \sum_{j\in L^v(p)} BWCost^j$, where $BWCost^j$ is a cost for embedding the virtual link j.

5.2. Evaluation environment

Since the network virtualization is an emerging field, it is not rigorous claim that substrate networks and VN requests lack of comprehension in the literature. The high degrees of freedom are usually given to embed VN requests in the complex substrate network compositions (e.g., multi-technologies and multi-layer, such as the IP over WDM in this study). According to the existing VNE literatures [37], we use the synthetic network topologies to evaluate the proposed algorithms.

Substrate network. We consider two SN topologies to evaluate the performances of the above algorithms, where the CPU resources at nodes and the bandwidths at links follow a uniform distribution from 50 to 100 units.

- As in the existing VNE literatures [32], a 24-node 43-link USA backbone IP network (USNET, in short) is considered, where the physical distance (km) of the link is indicated.
- The SN topology is configured to have 100 nodes and around 570 links, which corresponds to a medium-sized ISP (MSNET, in short). The SN topology is generated by the GT-ITM tool [38]. Each pair of the substrate nodes are randomly connected with probability 0.5. The link length follows a uniform distribution from 160 to 700 km.

Referred to the previous literature [30], P_b and P_l are set to be 10920W and 996W, respectively, where the energy consumption of core is 166W and the number of cores per physical router is 6. We evaluate performances under lightpath non-bypass, where P^{card} and P^{CA} are set to be 450W and 15W in the bandwidth 10 Gbps of each physical link [30]. We also evaluate performances under lightpath bypass, where P^{card}, P^{tr} and P^{CA} are respectively set to be 1000W, 73W and 8W in the link rate 40 Gbps [32]. The parameters of the energy consumptions are listed in Table 2.

Virtual network request. In line with the existing VNE literatures [16,37], the VN requests are created by the GT-ITM tool [38], and each pair of the nodes are randomly connected with probability 0.5. The arrivals of VN requests are modeled by a Poisson process. Each VN request will wait for one time window if it cannot be served immediately. A time window is equal to 100 time units. We run all

Table 1
Evaluation environment.

Num	SN	Scenarios	Virtual networks
1	USNET	Non-saturated	2–4 nodes per VN, 2 duration time window, 10 VNs per time window, 0–6 CPU, 0–6 bandwidth
2	USNET	Saturated	2–4 nodes per VN, 2 duration time window, 10 VNs per time window, 0–30 CPU, 0–30 bandwidth
3	MSNET	Non-saturated	2–20 nodes per VN, 5 duration time window, 10 VNs per time window, 0–6 CPU, 0–6 bandwidth
4	MSNET	Saturated	2–20 nodes per VN, 5 duration time window, 10 VNs per time window, 0–20 CPU, 0–20 bandwidth

Table 2
Parameters of energy consumption.

Num	Lightpath category	Parameters of energy consumption (W)
1	Non-bypass	$P_b = 10920$, $P_l = 996$, $P^{card} = 450$ and $P^{CA} = 15$
2	Bypass	$P_b = 10920$, $P_l = 996$, $P^{card} = 1000$, $P^{tr} = 73$ and $P^{CA} = 8$

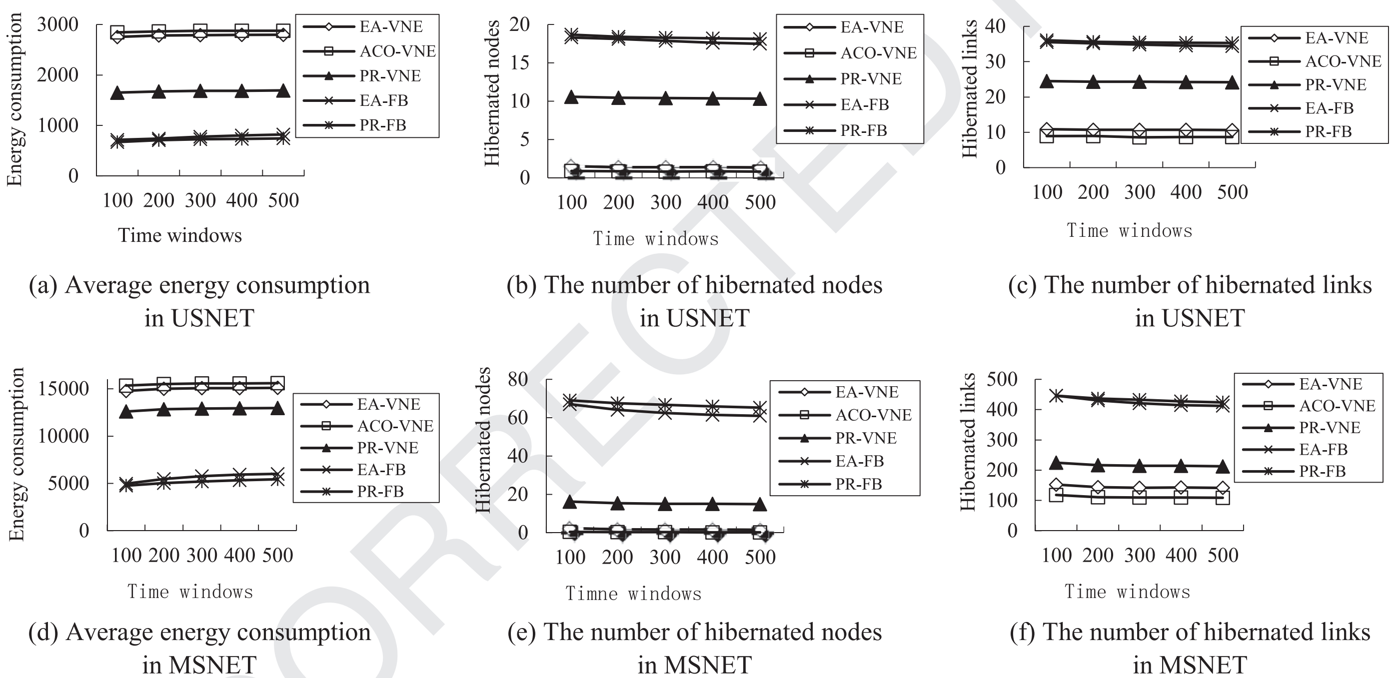


Fig. 4. Energy consumptions under lightpath non-bypass in the non-saturate scenarios.

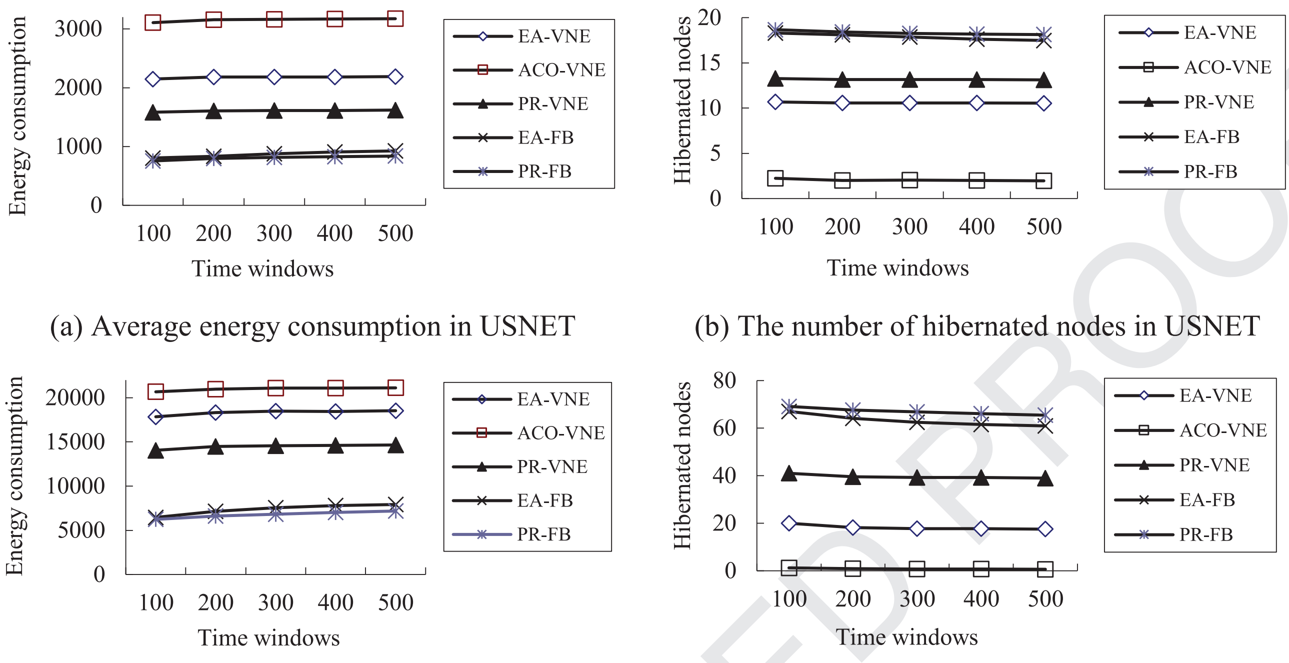
of our simulations in 500 time windows with average 10 VNs per time window, which amount to about 5000 VNs. To evaluate performances in the non-saturated scenarios and saturated scenarios, we construct four groups of VNs (shown in Table 1). For example, the state of “2–4 nodes per VN” in Table 1 means that the number of nodes per VN is randomly determined by a uniform distribution between 2 and 4. The state of “2 duration time window” means that the duration of the requests follows an exponential distribution with 2 time windows on average. The state of “10 VNs per time window” means that the arrivals of VNs follow a Poisson distribution with 10 VNs per time window on average. The states of “0–6 CPU” and “0-6 bandwidth” mean that the CPU and bandwidth requirements of the virtual nodes and links are real number distributed uniformly between 0 and 6, respectively. Ten different instances are run for the algorithms and the means of ten runs are recorded as the final results.

Comparison method. Due to the NP-hardness of the exact EEVNE approaches, we exclude them from comparisons. Our algorithms PR-FB and EA-FB have been described in Section 4.3. We compare them with the state-of-the-art algorithms: EA-VNE [16], ACO-VNE [19] and PR-VNE [36].

we evaluate the performances in the different loads of the following scenarios.

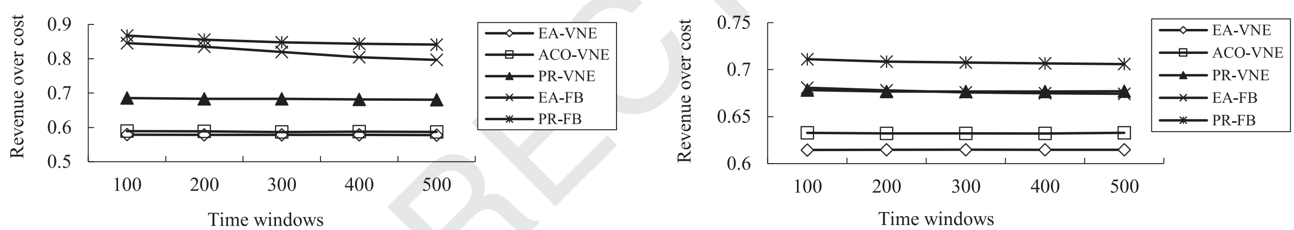
- The arrivals of the VNs follow a Poisson distribution with 2, 4, 6, 8, 10, 12, 14 and 16 VNs per time window on average, which amount to about 1000, 2000, 3000, 4000, 5000, 6000, 7000 and 8000 VNs in 500 time windows, respectively. The CPU and bandwidth of the virtual nodes and links are real number distributed uniformly in 0–30 units. Fig. 10(a–f) shows the results in the USNET, where the number of nodes per VN is randomly determined by a uniform distribution between 2 and 4, and the duration of the VNs follows an exponential distribution with 2 time windows on average. Fig. 10(g–l) shows the results in the MSNET, where the number of nodes per VN is randomly determined by a uniform distribution between 2 and 20, and the duration of the VNs follows an exponential distribution with 5 time windows on average.
- CPU and bandwidth of the virtual nodes and links are real number distributed uniformly in 0–6, 0–14, 0–32, 0–40, 0–48,

Please cite this article as: X. Chen et al., A feedback control approach for energy efficient virtual network embedding, Computer Communications (2015), http://dx.doi.org/10.1016/j.comcom.2015.10.010



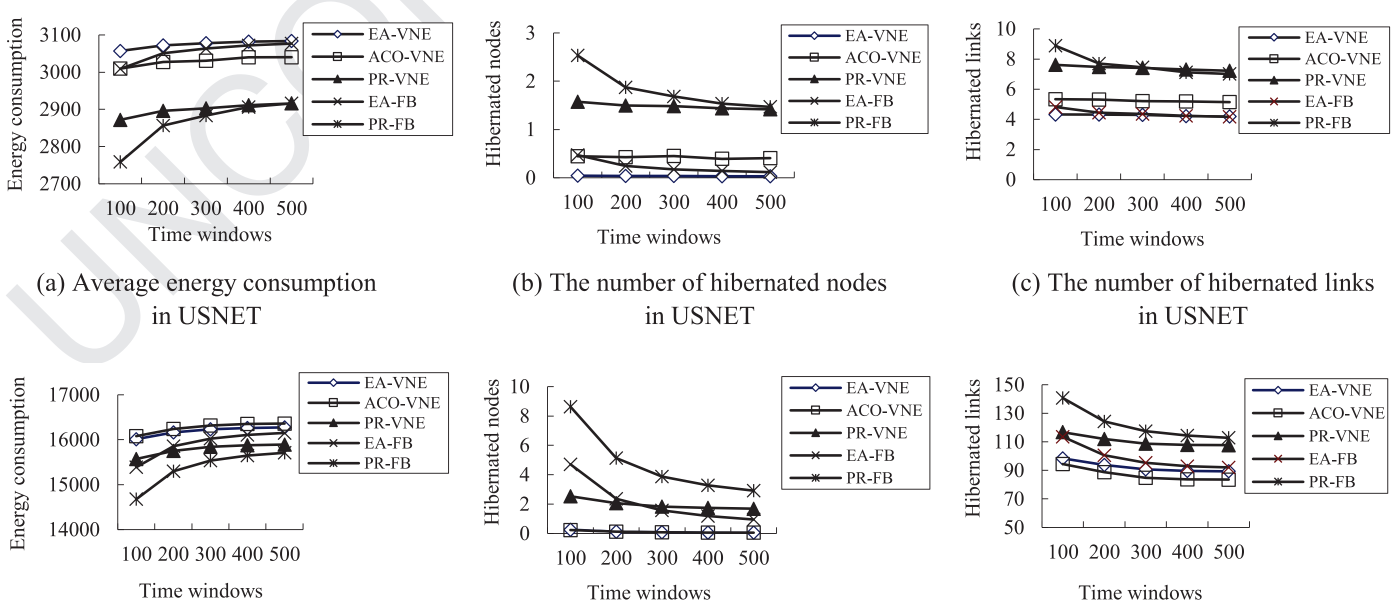
(a) Average energy consumption in USNET
(b) The number of hibernated nodes in USNET
(c) Average energy consumption in MSNET
(d) The number of hibernated nodes in MSNET

Fig. 5. Energy consumptions under lightpath bypass in the non-saturate scenarios..



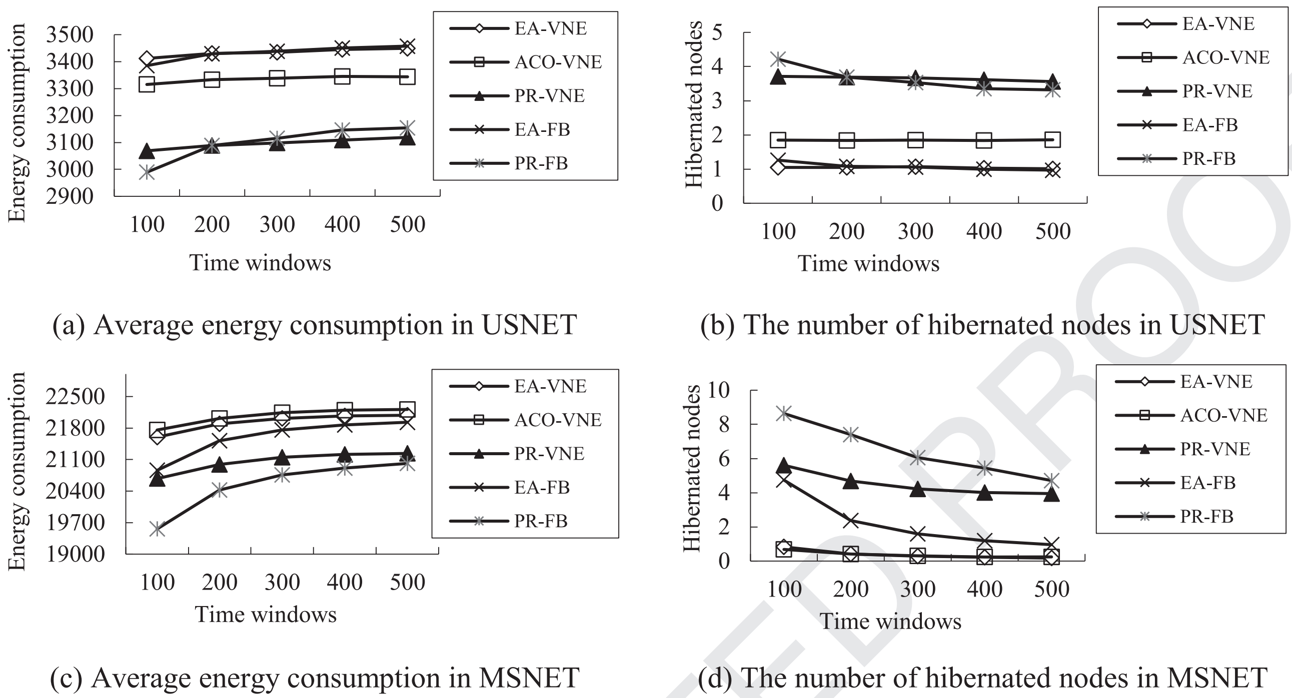
(a) Average revenue over cost in USNET
(b) Average revenue over cost in MSNET

Fig. 6. Revenue over cost in the non-saturate scenarios.



(a) Average energy consumption in USNET
(b) The number of hibernated nodes in USNET
(c) The number of hibernated links in USNET
(d) Average energy consumption in MSNET
(e) The number of hibernated nodes in MSNET
(f) The number of hibernated links in MSNET

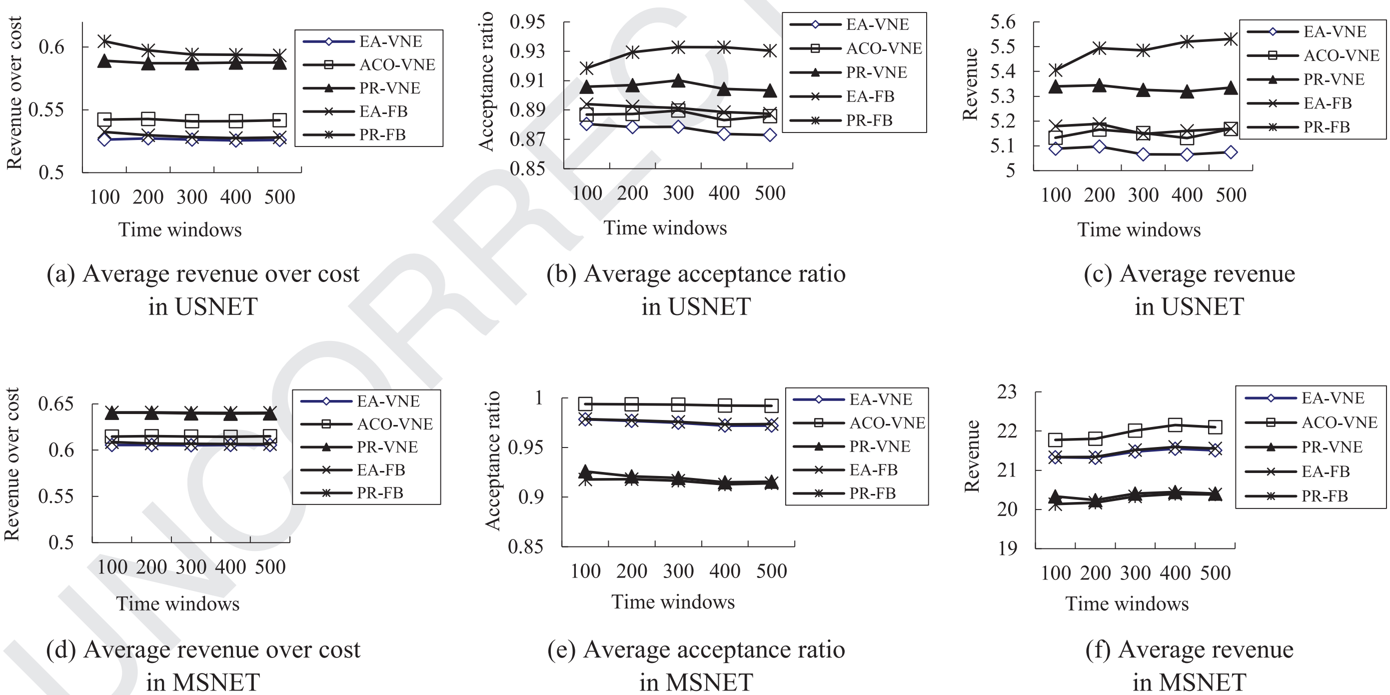
Fig. 7. The energy consumption under lightpath non-bypass in the saturated scenarios.



(a) Average energy consumption in USNET (b) The number of hibernated nodes in USNET

(c) Average energy consumption in MSNET (d) The number of hibernated nodes in MSNET

Fig. 8. The energy consumption under lightpath bypass in the saturated scenarios..



(a) Average revenue over cost in USNET (b) Average acceptance ratio in USNET (c) Average revenue in USNET

(d) Average revenue over cost in MSNET (e) Average acceptance ratio in MSNET (f) Average revenue in MSNET

Fig. 9. Other performances in the saturated scenarios.

0–54, 0–62, 0–70, 0–78, 0–86 and 0–94 units, and the arrivals of the VN requests follow a Poisson distribution with 10 VNs per time window on average, which amount to about 5000 VNs in 500 time windows. Fig. 12(a–f) shows the results in the USNET, where the number of nodes per VN is randomly determined by a uniform distribution between 2 and 4, and the duration of the VNs follows an exponential distribution with 2 time windows on average. Fig. 12(g–l) shows the results in the MSNET, where the number of nodes per VN is randomly determined by a uniform distribution between 2 and 20, and the duration of the VNs follows an exponential distribution with 5 time windows on average.

- Each pair of the substrate nodes are randomly connected with probability 0.1, 0.15, 0.2, 0.3, 0.4 and 0.5, respectively. The SN has 100 nodes, and the CPU resources at nodes and the bandwidths at links follow a uniform distribution from 50 to 100 units. CPU and bandwidth of the virtual nodes and links are real number distributed uniformly in 0–6 units, and the arrivals of the VN requests follow a Poisson distribution with 10 VNs per time window on average, which amount to about 5000 VNs in 500 time windows.

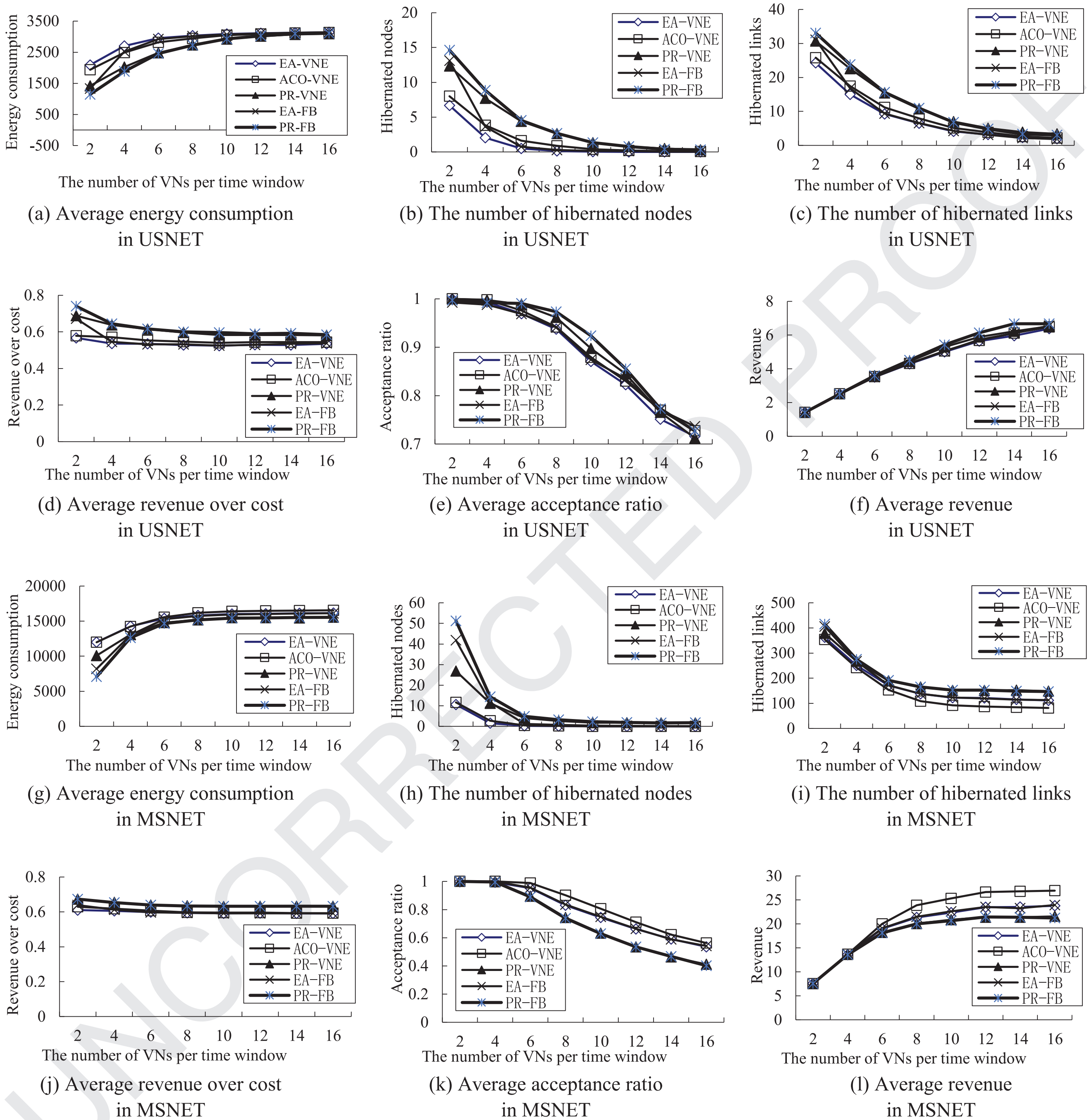


Fig. 10. Performances in the different number of VNs per time window under lightpath non-bypass.

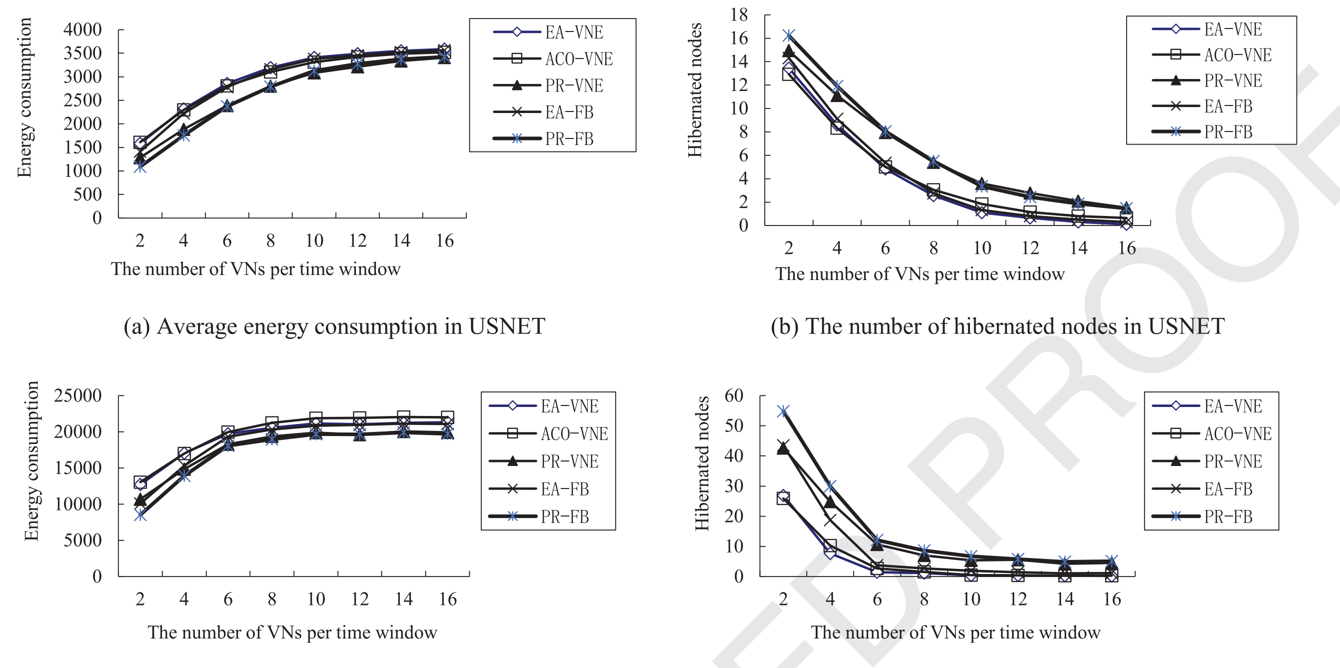
The number of nodes per VN is randomly determined by a uniform distribution between 2 and 20, and the duration of the VNs follows an exponential distribution with 5 time windows on average. The link length follows a uniform distribution between 200 and 700. Fig. 14 shows the results in the MSNET.

We also evaluate the energy consumption in the scenarios of the different maximum link length, where the link length follows a uniform distribution in 200–700, 200–1400, 200–2100, 200–2800, 200–3500 km, respectively. Fig. 15 shows the results.

5.3. Evaluation results in the non-saturated scenarios

In the non-saturated scenarios, all the VNs can be accepted. For all the algorithms, the acceptance ratio is 100%, and revenue is identical. We compare the performances of the energy consumption and revenue over cost.

(1) *Our algorithms significantly outperform the others in terms of the long-term average energy consumption under lightpath non-bypass and bypass*. Fig. 4(a–c) and (d–f) show the energy consumption and the number of the hibernated nodes and links of USNET and MSNET under lightpath non-bypass, respectively. Fig. 5(a–b) and (c–d) show



(a) Average energy consumption in USNET (b) The number of hibernated nodes in USNET

(c) Average energy consumption in MSNET (d) The number of hibernated nodes in MSNET

Fig. 11. Performances in the different number of VNs per time window under lightpath bypass.

the energy consumption and the number of the hibernated nodes of USNET and MSNET under lightpath bypass, respectively. The number of the hibernated links under lightpath bypass is identical to the lightpath non-bypass. From Fig. 4(a) and (d), we can see that PR-FB and EA-FB consume less energy than the others under lightpath non-bypass. In USNET, PR-FB and EA-FB respectively consumer the energy about 743W and 820W under lightpath non-bypass, which are less than PR-VNE (1694W), EA-VNE (2796W) and ACO-VNE (2876W). In MSNET, the energy consumptions of PR-FB and EA-FB are respectively about 5441W and 6016W under lightpath non-bypass, which are less than PR-VNE (12966W), EA-VNE (15099W) and ACO-VNE (15593W). Fig. 5(a) shows that the energy consumption of PR-FB and EA-FB are respectively about 837W and 925W in USNET, which consume less energy than PR-VNE (1619W), EA-VNE (3173W) and ACO-VNE (2188W) under lightpath bypass. Fig. 5(c) shows that the energy consumption of PR-FB and EA-FB are respectively about 7182W and 7921W in MSNET, which consume less energy than PR-VNE (14659W), EA-VNE (18538W) and ACO-VNE (21120W) under lightpath bypass. Since the feedback-control-based approach controls the mappable area of SN actively, hence PR-FB and EA-FB reduce the energy consumption and the number of the active links and nodes. Moreover, the intermediate nodes are bypassed, some algorithms achieve higher amounts of the hibernated nodes under lightpath bypath than that under lightpath non-bypass. For example, in the running of 500 time windows, the quantities of PR-FB, EA-FB, PR-VNE, EA-VNE and ACO-VNE in the MSNET are 65, 60, 38, 17 and 0.64 under lightpath bypass, and 65, 60, 14, 1.5 and 0.19 under lightpath non-bypass.

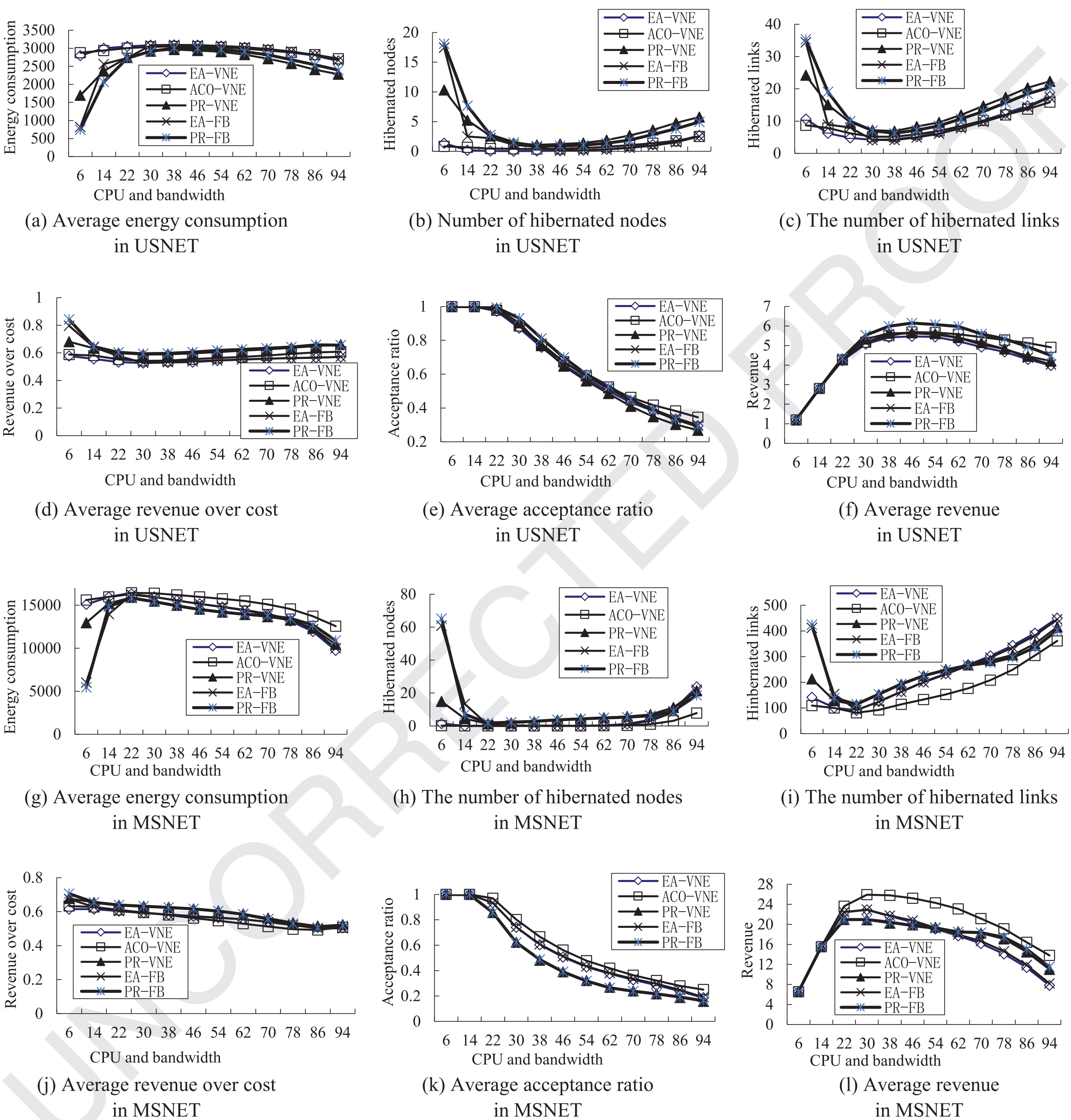
(2) *Our algorithms outperform the others in terms of the long-term average revenue over cost.* Fig. 6(a) and (b) shows the long-term average revenue over cost of USNET and MSNET, respectively. Our algorithms achieve higher revenue over cost than the other algorithms. In the USNET, PR-FB (0.841) and EA-FB (0.796) achieve higher revenue over cost than PR-VNE (0.68), EA-VNE (0.577), and ACO-VNE (0.587), respectively. In the MSNET, PR-FB (0.705) and EA-FB (0.674) achieve higher revenue over cost than PR-VNE (0.67), EA-VNE (0.614), and ACO-VNE (0.632). Since searching the solution of VNE in smaller mappable area of SN can increase the probability of reducing the length of path for embedding virtual links, as a result, PR-FB and EA-FB enhance the revenue over cost.

5.4. Evaluation results in the saturated scenarios

(1) *The long-term average energy consumptions of our algorithms are almost the same as the original algorithms.* Figs. 7 and 8 show the energy consumption and the number of the hibernated links and nodes of USNET and MSNET under lightpath non-bypass and bypass, correspondingly. We can see that with the increasing of time windows, the long-term energy consumptions and the number of the hibernated nodes and links of PR-FB and EA-FB are close to PR-VNE and EA-VNE, respectively. For example, in the running of 500 time windows, the energy consumptions of PR-FB, EA-FB, PR-VNE and EA-VNE in the USNET under lightpath non-bypass are 2915W, 3076W, 2915W and 3083W, respectively. The reason is that there is no space to extend the mappable area in the saturated scenarios.

(2) *The long-term acceptance ratio, revenue and revenue over cost of our algorithms are almost the same as the original algorithms.* In Figs. 9(d–f), we can see that the acceptance ratio, revenue and revenue over cost of PR-FB (0.914, 20.38 and 0.64) and EA-FB (0.973, 21.55 and 0.606) in the running of 500 time windows are almost the same as PR-VNE (0.915, 20.4 and 0.639) and EA-VNE (0.972, 21.5 and 0.605) in MSNET. When all the substrate resources are set the mappable flags, PR-FB and EA-FB are identical to PR-VNE and EA-VNE, respectively. The resources of the small size SN are vulnerable to the dynamical characteristics of VNE, and the feedback control approach can effectively reduce the influence on the number of the hibernated nodes and links caused by the dynamical characteristics. We can see Figs. 9(a–c) that the acceptance ratio, revenue and revenue over cost of PR-FB (0.93, 5.53 and 0.593) and EA-FB (0.887, 5.16 and 0.528) in the running of 500 time windows are slightly higher than PR-VNE (0.903, 5.33 and 0.587) and EA-VNE (0.872, 5.07 and 0.525) in USNET.

Please cite this article as: X. Chen et al., A feedback control approach for energy efficient virtual network embedding, Computer Communications (2015), http://dx.doi.org/10.1016/j.comcom.2015.10.010

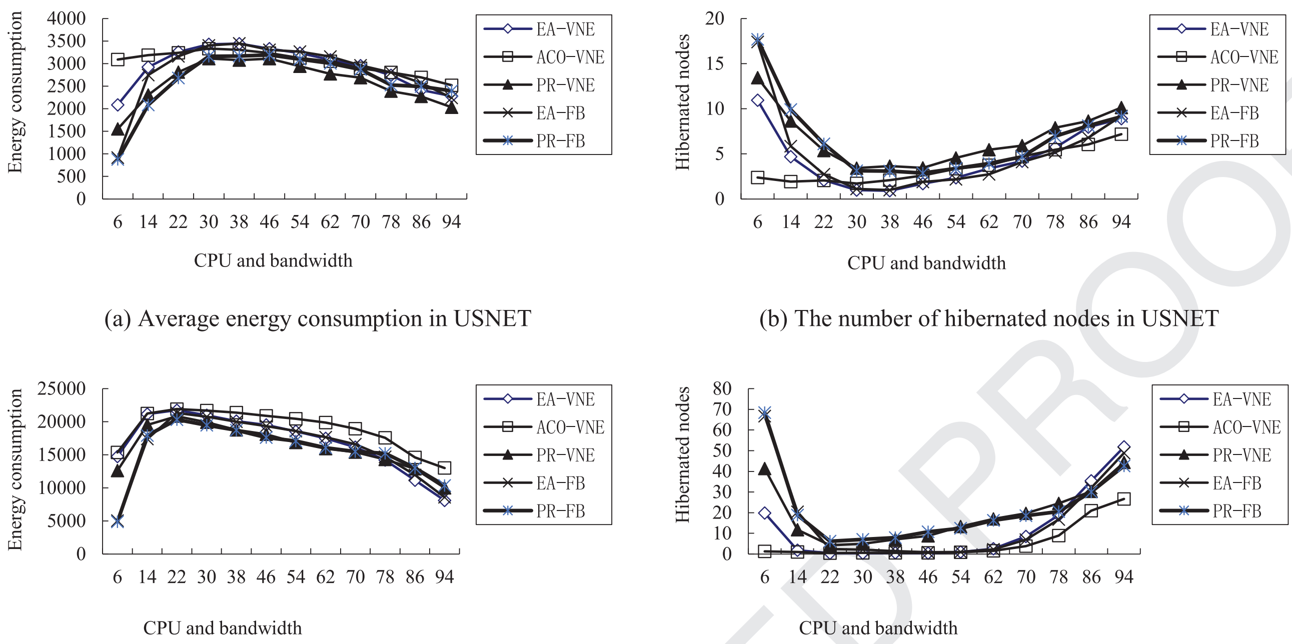


(a) Average energy consumption in USNET
(b) Number of hibernated nodes in USNET
(c) The number of hibernated links in USNET
(d) Average revenue over cost in USNET
(e) Average acceptance ratio in USNET
(f) Average revenue in USNET
(g) Average energy consumption in MSNET
(h) The number of hibernated nodes in MSNET
(i) The number of hibernated links in MSNET
(j) Average revenue over cost in MSNET
(k) Average acceptance ratio in MSNET
(l) Average revenue in MSNET

Fig. 12. Performances in the different CPU and bandwidth of VNs under lightpath non-bypass.

Table 3
Running time.

Num	SN	Scenario	Algorithms(Seconds)
1	USNET	Non-saturated	EA-VNE(1)/ACO-VNE(10)/PR-VNE(1)EA-FB(0.98)/PR-FB(0.97)
2	USNET	Saturated	EA-VNE(1)/ACO-VNE(10)/PR-VNE(2)/EA-FB(14)/PR-FB(28)
3	MSNET	Non-saturated	EA-VNE(139)/ACO-VNE(415)/PR-VNE(142)/EA-FB(137)/PR-FB(140)
4	MSNET	Saturated	EA-VNE(140)/ACO-VNE(412)/PR-VNE(170)/EA-FB(190)/PR-FB(220)



(a) Average energy consumption in USNET

(b) The number of hibernated nodes in USNET

(c) Average energy consumption in MSNET

(d) The number of hibernated nodes in MSNET

Fig. 13. Performances in the different CPU and bandwidth of VNs under lightpath bypass..

5.5. Other performances

(1) *The long-term running time used by our algorithms is slightly more than the original algorithms in the saturated scenarios, and is slightly less than the original algorithms in the non-saturated scenarios.* The long-term running time of all algorithms in 500 time windows is shown in Table 3. Since our feedback control algorithms set the mappable area in each iteration and the VNE may be solved after several feedback loops, PR-FB and EA-FB consume more time than PR-VNE and EA-VNE in the saturated scenarios, respectively. Furthermore, our proposed algorithms make a decrease in the mappable area, PR-FB and EA-FB consume less time than PR-VNE and EA-VNE in the non-saturated scenarios, respectively.

(2) *With the increasing of the time windows, the energy consumptions of all the algorithms increase slightly, and the number of the hibernated nodes and links of all the algorithms decreases slightly in the saturated scenarios.* Figs. 7 and 8 show the trends in the USNET and MSNET under lightpath non-bypass and bypass. For example, in the USNET under lightpath non-bypass, PR-FB, EA-FB, PR-VNE, EA-VNE and ACO-VNE consume the energy 2758W, 3008W, 2871W, 3057W and 3009W in the running of 100 time windows, and 2915W, 3076W, 2915W, 3083W and 3040W in the running of 500 time windows, respectively. The reason is that the substrate resources are dynamically allocated and recycled (due to the dynamical coming and leaving of the VNs), and more resource fragmentation of the SN is generated with the growing time windows.

(3) *The energy consumptions in the saturated and non-saturated scenarios are different.* From Figs. 4, 5, 7 and 8, we can see that the energy consumptions of all the algorithms in the non-saturated scenarios are less than in the saturated scenarios. Comparing the results of the non-saturated scenarios with the saturated scenarios, the CPU and bandwidth are increased 5 times in the USNET and 3.3 times in the MSNET comparing the non-saturated with the saturated scenarios. Under lightpath non-bypass, the average energy consumptions of our algorithms, PR-FB and EA-FB, increase 3.9 times and 3.7 times (2915W and 3076W in the saturated scenarios, and 743W and 820W in the non-saturated scenarios) in the USNET; 2.8 times and 2.7 times (15709W and 16151W in the saturated scenarios, and 5441W and 6016W in the non-saturated scenarios) in the MSNET respectively. Under lightpath bypass, the average energy consumptions of our algorithms PR-FB and EA-FB increase 3.7 times and 3.7 times (3154W and 3457W in the saturated scenarios, and 837W and 925W in the non-saturated scenarios) in the USNET; 2.9 times and 2.7 times (21014W and 21928W in the saturated scenarios, and 7182W and 7921W in the non-saturated scenarios) in the MSNET. The reason of the above phenomena is that the loads in the non-saturated scenarios are lighter than in the saturated scenarios, where the SN accommodates the more VNs in the saturated scenarios than in the non-saturated scenarios.

(4) *With the increasing number of VNs per time window, the energy consumption and revenue of all the algorithms increase, and the revenue over cost and acceptance ratio of all the algorithms decrease.* The trends under lightpath non-bypass and bypass are shown in Figs. 10 and 11. For example, in the USNET under lightpath non-bypass, PR-FB, EA-FB, PR-VNE, EA-VNE and ACO-VNE consume the energy 1146W, 1328W, 1419W, 2090W and 1936W under the environment of 2 VNs per time window, and 3109W, 3142W, 3095W, 3140W and 3135W under the environment of 16 VNs per time window, respectively. The increasing number of VNs per time window denotes the changes of the loads from light to heavy. When the loads change from light to heavy, the energy consumption and revenue increase accordingly, and the acceptance ratio and revenue over cost decrease inevitably. As described in [14] and [15], there is also a trade-off between the revenue and the energy efficiency in the heavy loads.

(5) *With the increasing CPU and bandwidth of VNs, the energy consumption and revenue of all the algorithms change from low to high firstly, and then from high to low. The revenue over cost of all the algorithms changes conversely.* In Figs. 12 and 13, our algorithms achieve lower energy consumption, more quantities of the hibernated nodes and links, and higher revenue over cost than the other algorithms in the light loads. For example, the energy consumptions of PR-FB, EA-FB, PR-VNE, EA-VNE and ACO-VNE under the environments of 6 CPU and 6 bandwidth of each VN, are 743W, 820W, 1694W, 2796W

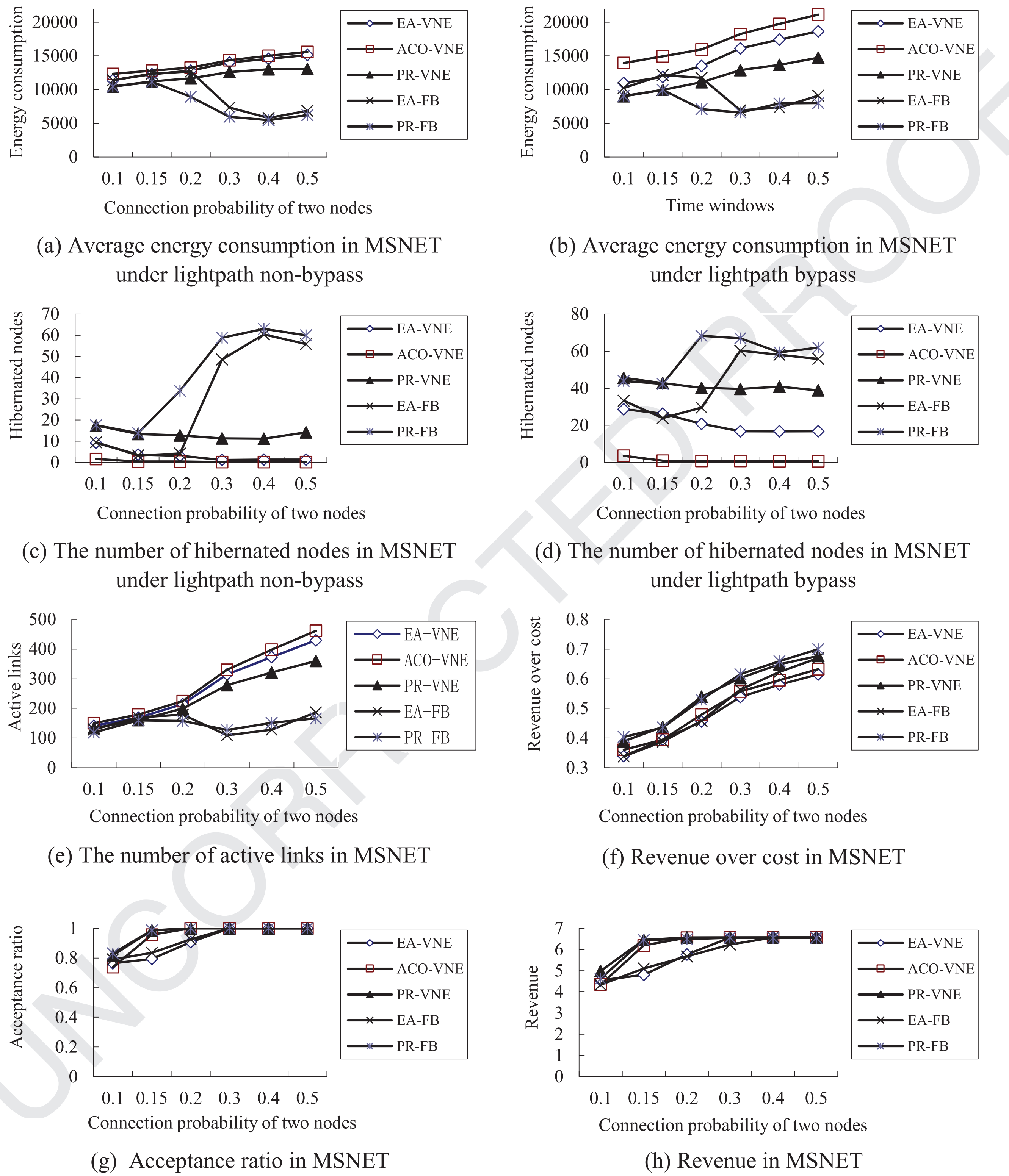
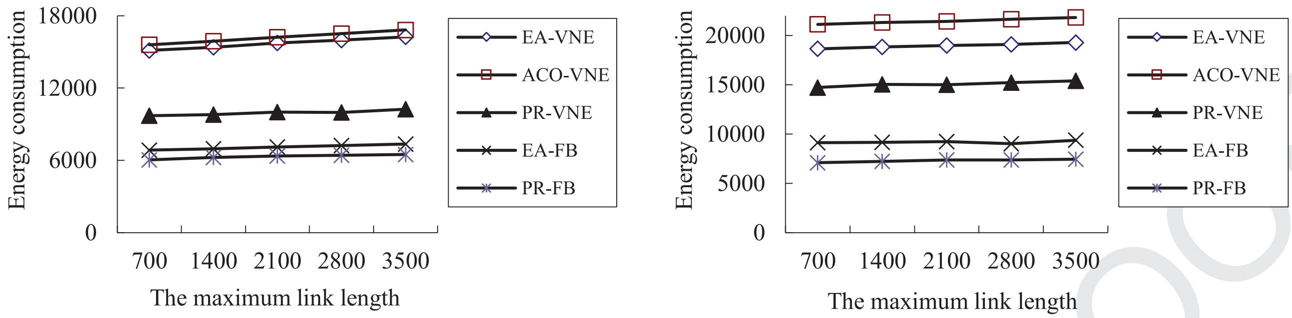


Fig. 14. Performances in the different connection probability of two substrate nodes..

and 2876W in the USNET under light non-bypass. When loads become heavy, the energy consumption and revenue will achieve maximal, and the revenue over cost will get minimal. For example, the energy consumptions of PR-FB, EA-FB, PR-VNE, EA-VNE and ACO-VNE under the environments of 46 CPU and 46 bandwidth of each VN, are 2989W, 3086W, 2936W, 3083W and 3045W in the USNET under light non-bypass. With the continual increasing loads, for the resource fragmentation becomes more and larger, the energy consumption and revenue begin to decrease, and the number of the hibernated nodes and links and revenue over cost begin to increase. For example, the energy consumptions of PR-FB, EA-FB, PR-VNE, EA-VNE and ACO-VNE under the environments of 94 CPU and 94 bandwidth of each VN, are 2394W, 2670W, 2277W, 2648W and 2711W in the USNET under light non-bypass.

Please cite this article as: X. Chen et al., A feedback control approach for energy efficient virtual network embedding, Computer Communications (2015), http://dx.doi.org/10.1016/j.comcom.2015.10.010



(a) Average energy consumption in MSNET under lightpath non-bypass

(b) Average energy consumption in MSNET under lightpath bypass

Fig. 15. The energy consumptions in the different link length in the non-saturated scenarios..

(6) *With the increasing connection probability of the two substrate nodes, the trends of the energy consumption of our algorithms change from low to high firstly, then from high to low, and finally from low to high.* At the beginning, with the continuous enhancement of the supplied resources, more VNs are accepted, and the energy consumptions of all the algorithms increase obviously in the saturated scenarios (shown in Fig. 14(a–h)). For example, the energy consumptions of PR-FB, EA-FB, PR-VNE, EA-VNE and ACO-VNE in the MSNET under lightpath bypass are 9063W, 10238W, 9010W, 10999W and 13942W in the connection probability 0.1 of two nodes, and 9964W, 12153W, 9965W, 11860W and 14917W in the connection probability 0.15 of two nodes, respectively. Then, when more amount of resources are supplied, there will be enough resources to accept all VNs. In the non-saturated scenarios, our algorithms keep a balance in the source supplies and the resource commands, and more substrate nodes and links enter into the hibernated state (shown in Fig. 14(c–e)), hence, the energy consumptions of our algorithms decrease. For example, in the MSNET under lightpath bypass, PR-FB and EA-FB consume the less energy (6598W and 6928W) in the connection probability 0.3 of two nodes than the energy consumption (7101W and 11767W) in the connection probability 0.2 of two nodes (shown in Fig. 14(b). When a growing connection probability of two substrate nodes comes, the number of the substrate links will increase. Since the quantities of the active links of our algorithms are increased (due to the dynamical characteristics of VNE), the energy consumptions of our algorithms will increase. For example, the energy consumptions of PR-FB and EA-FB in the MSNET under lightpath non-bypass are 5477W and 5793W in the connection probability 0.4, and 6211W and 6856W in the connection probability 0.5, respectively.

(7) *Since the enough supplies of resources can reduce the path length of embedding the virtual links, with the increasing connection probability of two substrate nodes, all the algorithms improve the revenue over cost.* With the increasing link resource supplies, all the algorithms enhance the revenue over cost (shown in Fig. 14(f)). For example, the revenue over costs of PR-FB (0.403), EA-FB (0.338), PR-VNE (0.389), EA-VNE (0.339) and ACO-VNE (0.36) in the connection probability 0.1 are higher than the values of PR-FB (0.699), EA-FB (0.669), PR-VNE (0.676), EA-VNE (0.614) and ACO-VNE (0.632) in the connection probability 0.5.

(8) *With the increasing connection probability of two substrate nodes, the quantities of the hibernated nodes of the proposed algorithms are different.* Fig. 14(c–d) shows the quantities of the hibernated nodes under lightpath non-bypass and bypass. Since the intermediate nodes are bypassed, the number of the hibernated nodes of all the algorithms under lightpath bypass are more than under lightpath non-bypass in the saturated scenarios. For example, the number of the hibernated nodes of PR-FB, EA-FB, PR-VNE, EA-VNE and ACO-VNE in the connection probability 0.1 of two nodes in the MSNET are 44, 33, 45, 28 and 3.5 under lightpath bypass, and 17, 9, 17, 9 and 1.6 under lightpath non-bypass, respectively. In the non-saturated scenarios, the quantities of the hibernated nodes of our algorithms under lightpath bypass are almost the same as the numbers under lightpath non-bypass. For example, in the connection probability 0.5 of the two nodes in the MSNET, the quantities of the hibernated nodes of PR-FB and EA-FB are 61 and 55.82 under lightpath bypass, and 60 and 55.8 under lightpath non-bypass, respectively. The above phenomena are caused as our feedback control can reduce the area of the active resources effectively but almost produce none of the intermediate nodes in the non-saturated scenarios.

(9) *With the increasing link length, the energy consumptions of all the algorithms are slightly enhanced.* For example, in the non-saturated scenarios of Table 1, the energy consumptions of PR-FB, EA-FB, PR-VNE, EA-VNE and ACO-VNE in the MSSNET are 6047W, 6855W, 9711W, 15130W and 15595W in 200–700 km of link length, and 6493W, 7358W, 10250W, 16246W and 16830W in 200–3500 km of link length, respectively (shown in Fig. 15(a–b)). Since the link length has effects on the energy consumption of the substrate links, the energy consumptions of all the algorithms increase predictively with the enhancement of the link length.

(10) *The targets in the saturated and non-saturated scenarios are different.* Since the resources are enough for accepting all VNs in the non-saturated scenarios, the minimization of the energy consumption is one important target (shown in Figs. 4 and 8). In the saturated scenarios, there are not enough resources, hence some VNs will be refused. As described in [14] and [15], there is a trade-off between the revenue and the energy efficiency (shown in Figs. 7, 8 and 9).

6. Conclusions

Due to the dynamic characteristics of VNE, the active resources of SN change frequently, more quantities of the nodes and links are activated and more energy consumptions are produced. In this paper, we presented a novel feedback control approach for EEVNE. The stable consolidated subset of the substrate resources can be found for current VNs. Two feedback-control-based algorithms were presented, which can increase the number of the hibernated links and nodes, as a result, the energy consumption can be reduced remarkably. Our algorithms in both theoretical analysis and simulations have shown their superiorities. The experiments demonstrate that a minimal subset of the substrate nodes and links for VNs can be found. The number of the hibernated nodes and links is enhanced, and the energy consumption is reduced significantly in the non-saturated scenarios.

The energy consumption of SN is closely related to the dynamic behavior of VNs, where the periodic and aperiodic dynamic changes of the loads usually coexist in the environment of VNE. In the future work, we focus on the state sensing and switching technology for energy savings and revenue maximization in the periodic and aperiodic dynamic changes of loads. Moreover, since there is a trade-off between the revenue and the energy efficiency in the saturated scenarios, it is worth exploring the pricing strategy to keep the balance among revenue, cost and energy saving.

References

[1] B. Raghavan, J. Ma, The energy and emergy of the Internet, in: Proceedings of the 10th ACM Workshop on Hot Topics in Networks, Cambridge, MA, USA, 2011, pp. 1–6, doi:10.1145/2070562.2070571.

[2] J. Giles, Internet responsible for 2 percent of global energy usage (2011). http://www.newscientist.com/blogs/onepercent/2011/10/307-gw-the-maximum-energy-the.html.

[3] D. Pamlin, K. Szomolanyi, Saving the climate @ the speed of lightfirst roadmap for reduced CO2 emissions in the EU and beyond, in: World Wildlife Fund and European Telecommunications Network Operators' Association, 2007.

[4] A.P. Bianzino, C. Chaudet, D. Rossi, J.L. Rougier, A survey of green networking research, IEEE Commun. Surv. Tutor. 14 (1) (2012) 1–18, doi:10.1109/SURV.2011.113010.00106.

[5] T. Anderson, L. Peterson, S. Shenker, J. Turner, Overcoming the internet impass through virtualization, IEEE Comput. Mag. 38 (4) (2005) 34–41, doi:10.1109/MC.2005.136.

[6] M. Sharkh, M. Jammal, A. Shami, A. Ouda, Resource allocation in a network-based cloud computing environment: design challenges, IEEE Commun. Mag. 51 (11) (2013) 46–52, doi:10.1109/MCOM.2013.6658651.

[7] D. Drutskoy, E. Keller, J. Rexford, Scalable network virtualization in software-defined networks, IEEE Internet Comput. 17 (2) (2013) 20–27, doi:10.1109/MIC.2012.144.

[8] A. Fischer, J. Botero, M. Beck, H. Meer, X. Hesselbach, Virtual network embedding: a survey, IEEE Commun. Surv. Tutor. 15 (4) (2013) 1888–1906, doi:10.1109/SURV.2013.013013.00155.

[9] J. Turner, B. Heller, J. Lu, P. Crowley, F. Kuhns, M. Wilson, J. DeHart, S. Kumar, C. Wiseman, A. Freestone, J. Lockwood, D. Zar, Supercharging PlanetLab - a high performance, multi-application, overlay network platform, in: SIGCOMM '07 Proceedings of the 2007 conference on applications, technologies, architectures, and protocols for computer communications, Kyoto, Japan, 2007, pp. 85–97, doi:10.1145/1282427.1282391.

[10] S. Bhatia, M. Motiwala, W. Muhlbauer, Y. Mundada, V. Valancius, A. Bavier, N. Feamster, L. Peterson, J. Rexford, Trellis: a platform for building flexible, fast virtual networks on commodity hardware, in: Proceedings of the ACM CoNEXT Conference, New York, NY, USA, 2008, doi:10.1145/1544012.1544084.

[11] D. Medhi, B. Ramamurthy, C. Scoglio, J.P. Rohrer, E.K. Cetinkaya, R. Cherukuri, X. Liu, P. Angu, A. Bavier, C. Buffington, J.P.G. Sterbenz, The GpENI testbed: Network infrastructure, implementation experience,and experimentation, Comput. Netw. 61 (2014) 51–74, doi:10.1016/j.bjp.2013.12.027.

[12] M. Berman, J.S. Chase, L. Landweber, A. Nakao, M. Ott, D. Raychaudhuri, R. Ricci, I. Seskar, GENI: a federated testbed for innovative network experiments, Comput. Netw. 61 (2014) 5–23, doi:10.1016/j.bjp.2013.12.037.

[13] J. Chabarek, J. Sommers, P. Barford, C. Estan, D. Tsiang, S. Wright, Power awareness in network design and routing, in: IEEE INFOCOM, Phoenix, AZ, 2008, pp. 457–465, doi:10.1109/INFOCOM.2008.93.

[14] J. Botero, X. Hesselbach, M. Duelli, D. Schlosser, A. Fischer, H. Meer, Energy efficient virtual network embedding, IEEE Commun. Lett. 16 (5) (2012) 756–759, doi:10.1109/LCOMM.2012.030912.120082.

[15] J. Botero, X. Hesselbach, Greener networking in a network virtualization environment, Comput. Netw. 57 (9) (2013) 2021–2039, doi:10.1016/j.comnet.2013.04.004.

[16] S. Su, Z. Zhang, X. Cheng, Y. Wang, Y. Luo, J. Wang, Energy-aware virtual network embedding through consolidation, in: IEEE Computer Communications Workshops (INFOCOM WKSHPS), Orlando, 2012, pp. 127–132, doi:10.1109/INFCOMW.2012.6193473.

[17] R. Garroppo, G. Nencioni, L. Tavanti, M. Scutella, Does traffic consolidation always lead to network energy savings? IEEE Commun. Lett. 17 (9) (2013) 1852–1855, doi:10.1109/LCOMM.2013.070913.131244.

[18] S. Su, Z. Zhang, A. Liu, X. Cheng, Y. Wang, X. Zhao, Energy-aware virtual network embedding, IEEE/ACM Trans. Netw. 22 (5) (2014) 1607–1620, doi:10.1109/TNET.2013.2286156.

[19] B. Wang, X. Chang, J. Liu, J. Muppala, Reducing power consumption in embedding virtual infrastructures, in: IEEE Globecom Workshops (GC Wkshps), Anaheim, CA, 2012, pp. 714–718, doi:10.1109/GLOCOMW.2012.6477662.

[20] X. Chang, B. Wang, J. Liu, J. Muppala, Green cloud virtual network provisioning based ant colony optimization, in: Proceeding of the 15th annual conference companion on Genetic and evolutionary computation, ACM, New York, 2013, pp. 1553–1560, doi:10.1145/2464576.2482735.

[21] D.G. Andersen, Theoretical approaches to node assignment (2002). http://repository.cmu.edu/compsci/86.

[22] X. Chen, C. Li, Energy efficient virtual network embedding for path splitting, in: The 16th Asia-Pacific Network Operations and Management Symposium (APNOMS), Hsinchu, 2014, pp. 1–4, doi:10.1109/APNOMS.2014.6996550.

[23] Y. Tarutani, Y. Ohsita, M. Murata, Virtual network reconfiguration for reducing energy consumption in optical data centers, J. Opt. Commun. Netw., IEEE/OSA 6 (10) (2014) 925–942, doi:10.1364/JOCN.6.000925.

[24] L. Wang, F. Zhang, A. Vasilakos, C. Hou, Z. Liu, Joint virtual machine assignment and traffic engineering for green data center networks, ACM SIGMETRICS Perform. Eval. Rev. 41 (3) (2013) 107–112, doi:10.1145/2567529.2567560.

[25] L. Nonde, T. El-Gorashi, J. Elmirghani, Green virtual network embedding in optical OFDM cloud networks, in: The 16th International Conference on Transparent Optical Networks (ICTON), IEEE, Graz, 2014, pp. 1–5, doi:10.1109/ICTON.2014.6876422.

[26] X. Guan, B. Choi, S. Song, Topology and migration-aware energy efficient virtual network embedding for green data centers, in: The 23rd International Conference on Computer Communication and Networks (ICCCN), IEEE, Shanghai, 2014, pp. 1–8, doi:10.1109/ICCCN.2014.6911768.

[27] K. Nguyen, M. Cheriet, Environment-aware virtual slice provisioning in green cloud environment, IEEE Trans. Serv. Comput. PP (99) (2014) 1–14, doi:10.1109/TSC.2014.2362544.

[28] A. Qureshe, R. Weber, H. Balakrishnan, J. Guttag, B. Maggs, Cutting the electric bill for internet-scale systems, in: Proceedings of the ACM SIGCOMM 2009 conference on Data communication, New York, NY, USA, 2009, pp. 123–134, doi:10.1145/1592568.1592584.

[29] A. Bianzino, C. Chaudet, F. Larroca, D. Rossi, J. Rougier, Energy-aware routing: a reality check, in: IEEE GLOBECOM Workshops (GC Wkshps), Miami, FL, 2010, pp. 1422–1427, doi:10.1109/GLOCOMW.2010.5700172.

[30] E. Rodriguez, G. Alkmim, D.M. Batista, N.L.S. da Fonseca, Green virtualized networks, in: IEEE International Conference on Communications (ICC), Ottawa, ON, 2012, pp. 1970–1975, doi:10.1109/ICC.2012.6364546.

[31] F. Musumeci, massimo Tornatore, A. Pattavina, A power consumption analysis for ip-over-wdm core network architectures, J. Opt. Commun. Netw. 4 (2) (2012) 108–117, doi:10.1364/jocn.4.000108.

[32] G. Shen, R. Tucker, Energy-minimized design for ip over wdm networks, IEEE/OSA J. Opt. Commun. Netw. 1 (1) (2009) 176–186, doi:10.1364/JOCN.1.000176.

[33] M.N. Dharmaweera, R. Parthiban, Y.A. Sekercioglu, Towards a power-efficient backbone network: the state of research, IEEE Commun. Surv. Tutor. 17 (1) (2015) 198–227, doi:10.1109/COMST.2014.2344734.

[34] D. Unnikrishnan, R. Vadlamani, Y. Liao, A. Dwaraki, J. Crenne, L. Gao, R. Tessier, Scalable network virtualization using FPGAs, in: Proceedings of the 18th annual ACM/SIGDA international symposium on Field programmable gate arrays, ACM, New York, 2010, pp. 219–228, doi:10.1145/1723112.1723150.

[35] G. Lu, C. Guo, Y. Li, Z. Zhou, T. Yuan, H. Wu, Y. Xiong, R. Gao, Y. Zhang, Serverswitch: a programmable and high performance platform for data center networks, in: Proceedings of the 8th USENIX conference on Networked systems design and implementation, Berkeley, CA, USA, 2011, pp. 15–28.

[36] X. Cheng, S. Su, Z. Zhang, H. Wang, F. Yang, Y. Luo, J. Wang, Virtual network embedding through topology-aware node ranking, ACM SIGCOMM Comput. Commun. Rev. 42 (2) (2011) 38–47, doi:10.1145/1971162.1971168.

[37] M. Chowdhury, M.R. Rahman, R. Boutaba, Vineyard: virtual network embedding algorithms with coordinated node and link mapping, IEEE/ACM Trans. Netw. 20 (1) (2012) 206–219, doi:10.1109/TNET.2011.2159308.

[38] E.W. Zegura, K.L. Calvert, S. Bhattacharjee, How to model an internetwork, in: Proceedings of the Fifteenth annual joint conference of the IEEE computer and communications societies conference on The conference on computer communications, San Francisco, CA, in: INFOCOM'96, vol. 2, 1996, pp. 594–602, doi:10.1109/INFCOM.1996.493353.

Please cite this article as: X. Chen et al., A feedback control approach for energy efficient virtual network embedding, Computer Communications (2015), http://dx.doi.org/10.1016/j.comcom.2015.10.010