TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 05/11 pp407–414
Volume 21, Number 4, August 2016

Efficient Algorithm for Energy-Aware Virtual Network Embedding

Shuxian Jia, Guiyuan Jiang, Peilan He, and Jigang Wu*

Abstract: Network virtualization is a promising approach for resource management that allows customized Virtual Networks (VNs) to be multiplexed on a shared physical infrastructure. A key function that network virtualization can provide is Virtual Network Embedding (VNE), which maps virtual networks requested by users to a shared substrate network maintained by an Internet service provider. Existing research has worked on this, but has primarily focused on maximizing the revenue of the Internet service provider. In this paper, we consider energy-aware virtual network embedding, which aims at minimizing the energy consumption for embedding virtual networks in a substrate network. In our optimization model, we consider energy consumption of both links and nodes. We propose an efficient heuristic to assign virtual nodes to appropriate substrate nodes based on priority, where existing activated nodes have higher priority for hosting newly arrived virtual nodes. In addition, our proposed algorithm can take advantage of activated links for embedding virtual links so as to minimize total energy consumption. The simulation results show that, for all the cases considered, our algorithm can improve upon previous work by an average of 12.6% on acceptance rate, while the consumed energy can be reduced by 12.34% on average.

Key words: virtualization technology; virtual network embedding; energy efficient; optimization algorithm

1 Introduction

With the development of data centers and cloud computing, network service requests from end users impose various requirements on Quality-of-Service (QoS) guarantees. The ever-increasing requirements placed on the network are fueling its evolution to architectures that make more efficient use of available network resources, and promote service innovations. Internet Service Providers (ISPs) have to satisfy individual needs for their customers, and hence are impelled to use different protocol stacks and provide custom services and network resources.

In recent years, the networking research community has become increasingly interested in efficient and flexible management approaches for resource allocation. Virtualization technology, which utilizes critical resources by creating virtual (rather than actual) versions of the corresponding resources, such as operating systems, servers, storage devices, and network resources, provides new opportunities for improving resource efficiency. In particular, the concept of network virtualization[1–3], allowing customized Virtual Networks (VNs) to be set up on a shared physical infrastructure, has been proposed as a promising approach for the future Internet, which aims to overcome the resistance of the current Internet to architectural change.

Virtualization is an abstract concept that hides the complexity of underlying substrate network details, thereby allowing ISPs to cope with the wide range of customer requirements. In network virtualization, Infrastructure Providers (InPs) divide their resources

• Shuxian Jia is with the School of Computer Software Technology, Zhejiang University, Ningbo 315048, China. E-mail: shuxianJia@outlook.com.
• Guiyuan Jiang and Peilan He are with the School of Computer Science and Engineering, Nanyang Technological University, 639798, Singapore. E-mail: gyjiang@ntu.edu.sg; phe002@ntu.edu.sg.
• Jigang Wu is with the State Key Laboratory of Computer Architecture, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China. E-mail: asjgwu@outlook.com.
∗ To whom correspondence should be addressed.
Manuscript received: 2016-02-21; accepted: 2016-03-22

into chunks, called VNs, which are allocated to SPs. Thanks to virtualization, the resource chunks are isolated from each other, so the service networks behave as if they were independent, though they share the same substrate infrastructure. Large enterprises like Google, Facebook, and Amazon actually utilize virtualized data centers for large-scale computation and storage[4].

Due to the advantages of the technique, it has received significant attention, since it makes it possible to set up a cost-effective data center infrastructure for storing large volumes of data and to host large-scale service applications. However, this technology requires efficient algorithms, commonly called "Virtual Network Embedding (VNE)" algorithms, to instantiate virtualized networks on a substrate infrastructure, and optimize the layout for service-relevant metrics.

Previous work has shown that the VNE problem is NP-hard, as it can be reduced to a multi-way separator problem[5–7]. The VNE problem has received extensive investigation from different perspectives such as load balancing in the substrate network[8] and efficient use of resources[9]. The challenges are principally caused by the high computational complexity[10, 11]. Most proposals have focused on a two-phase embedding approach, first by preselecting the mapping of virtual nodes, and second by assigning virtual links to substrate paths. This approach ignores the benefit of joint design of node mapping and link mapping through efficiency coordination. On the other hand, many proposals in the literature have therefore focused on restricting the problem to particular scenarios, such as assuming infinite network capacities in order to gain obvious admission control, ignoring either virtual link or node QoS requirements[12], relaxing embedding path-splitting constraints[13], focusing on specific VN topologies[12], and designing greedy embedding algorithms[12, 14].

Network virtualization techniques can also lead to significant energy savings by means of resource consolidation[15]. Existing Energy-Efficient VNE (EEVNE) algorithms use different approaches to minimize energy consumption. Razzaq et al.[16] proposed a Baseline Approach (BLA) to deal with assigning virtual nodes to any substrate nodes that satisfy the demand. A path-based model called P-VNE was presented by Hu et al.[17] for the VNE problem. Xu et al.[18] proposed an efficient embedding algorithm, which explores periodic resource demands of virtual networks and employs a new embedding metric. Guan et al.[19] built a novel model of a VNE problem to minimize energy usage in data centers. Oliveira et al.[20] presented an Opportunistic Resilience Embedding (ORE), which protects virtual links against substrate network disruptions.

Most existing EEVNE algorithms try to minimize energy consumption by resource consolidation. However, due to the relationship between the energy consumption of nodes and CPU utilization, virtual node embedding has great impact on the energy consumption of substrate networks, which is not considered in existing EEVNE algorithms. In this paper, we investigate the problem of energy-efficient virtual network embedding and propose an efficient algorithm to obtain minimization of energy consumption of virtual node embedding under the constraints of virtual link embedding. Specifically, we formulate virtual network embedding as an optimization problem for minimizing energy consumption and propose an efficient heuristic algorithm to reduce energy consumption for the substrate network. We conduct extensive experiments to evaluate the performance of our proposed algorithm.

The rest of this paper is organized as follows. We introduce related definitions and important notations, and formulate the EEVNE problem in Section 2. The proposed algorithm is introduced in Section 3, and Section 4 presents the experimental results with analysis. We conclude our work in Section 5.

2 Preliminaries

In this section, we present the definitions and important notations, then formulate virtual network embedding as an optimization problem for minimizing energy consumption.

2.1 Notation

A substrate network is modeled as an undirected weighted graph $G_{\mathrm{P}} = (N_{\mathrm{P}}, L_{\mathrm{P}})$, where N_{P} is the set of physical nodes and L_{P} is the set of physical links. Similar to the substrate network, a virtual network is modeled as an undirected graph, which is denoted as $G_{\mathrm{V}} = (N_{\mathrm{V}}, L_{\mathrm{V}})$, where N_{V} is the set of virtual nodes and L_{V} is the set of virtual links. Let m_{SN} be the number of physical nodes, i.e., $m_{\mathrm{SN}} = |N_{\mathrm{P}}|$, and n_{VN} is the number of virtual nodes, i.e., $n_{\mathrm{VN}} = |L_{\mathrm{P}}|$. $|L_{\mathrm{V}}|$ is the total number of the set of virtual links.

Each physical node $n_{\mathrm{p}} \in N_{\mathrm{P}}$ is associated with $\mathrm{CP}(n_{\mathrm{p}})$ units of CPU resources, and $\mathrm{ACP}(n_{\mathrm{p}})$ units are currently available for accepting new computation tasks.

Let $C_{\rm b}$ denote the base energy consumption, $C_{\rm m}$ be the total energy consumption, and $C_{\rm curr}$ represents the total energy consumption for all the nodes that have been allocated into the physical networks. Let ABR$(n_{\rm s})$ be the amount of unoccupied bandwidth of all the outgoing links of substrate node $n_{\rm s}$. Each substrate link $(i,j)((i,j)\in L_{\rm P})$ from physical node i to j is associated with the amount of energy consumption $C_{\rm n}(l_{\rm p})$.

For every virtual node $n_{\rm v}$, the demanding CPU resource is denoted as DCP$(n_{\rm v})$, while the required bandwidth for virtual link $l_{\rm v}$ is denoted as DBR$(l_{\rm v})$. The amount of required bandwidth of all outgoing links of a given virtual node $n_{\rm v}$ is denoted as DBR$(n_{\rm v})$.

2.2 Problem description

In this paper, we focus on the impact of virtual node mapping on the energy consumption of a substrate network. We keep the same model and assumption as previous work[21]. The physical network is quantified as follows.

Let η_i be the rate of CPU utilization of a physical node. Since the CPU utilization is a main contributor to the energy consumption of a server, the i-th node energy consumption E_i can be calculated as $C_{\rm b}+C_{\rm l}\cdot\eta_i$, where $C_{\rm b}$ is the base energy consumption of the substrate node, $C_{\rm m}$ is the total energy consumption of a substrate node.

Let the binary variable $a(e)$ be 1 if the node or link e is activated, and $a(e)=0$ otherwise. Thus, if a node or link is not activated, its energy consumption is 0. The energy consumption model of the nodes and links of a physical network is calculated as follow.

$$E_{\rm SN}=\sum_{n_{\rm s}\in N_{\rm S}} a(n_{\rm s})\cdot(C_{\rm b}-(C_{\rm m}-C_{\rm b})\times\eta_{n_{\rm s}})+\sum_{l_{\rm p}\in L_{\rm P}} a(l_{\rm p})\cdot C_{\rm n}(l_{\rm p}) \tag{1}$$

In order to reduce the energy consumption, we should consider the energy consumption of both nodes and links. On the other hand, if the available CPU resource ACP$(n_{\rm p})$ is less than the required CPU resource DCP$(n_{\rm v})$, or the available bandwidth ABR$(n_{\rm s})$ is less than required DBR$(n_{\rm v})$, then the mapping is not successful. Otherwise we denote the energy cost per CPU of mapping n_i into n_j as follows. Let cost$_{ij}$ be the energy cost per CPU of embedding i into j. Note that if a physical node is not active, we need to set the cost$_{ij}$ to a large value. Hence, cost$_{ij}$ is calculated as

$$\text{cost}_{ij}=\begin{cases}\dfrac{1}{\text{CP}(n_{\rm p})}, \text{if physical node } n_{\rm p} \text{ is active;}\\ \dfrac{1}{\min\limits_{n_i\in N_{\rm P}}\{\text{CP}(n_{\rm p})\}-0.0001}, \text{otherwise}\end{cases} \tag{2}$$

According to the aforementioned theory, the key to reducing the energy cost per CPU of mapping i to j is allocating priority in the VEN problem model. In this section, we propose a novel way to calculate the priority for selecting a physical node.

$$W_{\rm cost}(n_{\rm p})=\alpha\times\text{CP}(n_{\rm p})+\beta\times\text{ACP}(n_{\rm p})+(1-a(n_{\rm p}))\times\text{MAX} \tag{3}$$

where α and β are weight factors, and MAX is a constant integer. Here, we set MAX to be the biggest capacity of all substrate nodes. It is obvious that increasing the parameter α can improve the efficiency for embedding a computation-intensive virtual network, while increasing the parameter β is more useful for embedding communication-intensive virtual networks.

Through the above analysis, the EEVEN problem can be formulated as follows.

Objective:

$$\sum_{n_{\rm s}\in N_{\rm S}} a(n_{\rm s})\cdot(C_{\rm b}-(C_{\rm m}-C_{\rm b})\times\eta_{n_{\rm s}})+\sum_{l_{\rm p}\in L_{\rm P}} a(l_{\rm p})\cdot C_{\rm n}(l_{\rm p}) \tag{4}$$

s.t.

$$\sum_{n_{\rm v}\in N_{\rm V}}\text{DCP}(n_{\rm v})x(n_{\rm v},n_{\rm p})\leqslant\text{CP}(n_{\rm p}) \tag{5}$$

$$\sum_{l_{\rm v}\in L_{\rm V}}\text{DBR}(l_{\rm v})x(n_{\rm v},n_{\rm p})\leqslant\text{ABR}(n_{\rm p}) \tag{6}$$

$$\sum_{i\in N_{\rm V}} x(i,u)\leqslant 1 \tag{7}$$

$$\sum_{u\in N_{\rm P}} x(i,u)=1 \tag{8}$$

where the binary variant $x(i,u)$ is defined as follows.

$$x(i,u)=\begin{cases}1, \text{if node } i \text{ is embedded to node } u;\\ 0, \text{otherwise}\end{cases} \tag{9}$$

Constraint (5) guarantees that the total required CPU capacity of virtual nodes mapped to physical nodes always be equal to or less than the total required CPU capacity of virtual nodes mapped to physical nodes, which must always be equal to or less than the available computational resources on it. Similarly, another constraint (6) ensures that all of the bandwidth demanded by the virtual links will not exceed the unoccupied bandwidth of all the outgoing links of a substrate node. Constraint (7) is used to ensure that every i will be linked to a substrate node u. Constraint (8) is employed to ensure that each substrate node allows no more than one virtual node.

3 Proposed Algorithms

In this section, we proposed an efficient heuristic algorithm shown in Algorithm 1, called `EH_Alg`, to minimize the energy consumption of virtual node embedding under the constraints of virtual link embedding.

In our approach, `EH_Alg` determines how to assign a virtual node according to the following principles.

(1) It uses $W_{\text{cost}}(n_{\text{p}})$ to measure the priority of a substrate node of the physical network, to accommodate target virtual nodes, according to the total CPU resources and unoccupied CPU capacities. In particular, the proposed $W_{\text{cost}}(n_{\text{p}})$ allows embedding more than one virtual node into the same physical node.

(2) After finding a substrate node j, we consider that a virtual node i is mapped into j. If the substrate node j still has the unoccupied CPU resources, then we try to embed the neighbor node of node i into substrate node j. If the virtual node i has more than one neighbor node, the algorithm chooses a node according to the amount of required bandwidth between i and its neighbors. The node with large bandwidth requirements takes higher priority over those with low bandwidth needs. The process repeats until substrate node j is overloaded. This process minimizes the use of communication resources.

Algorithm 1 EH_Alg

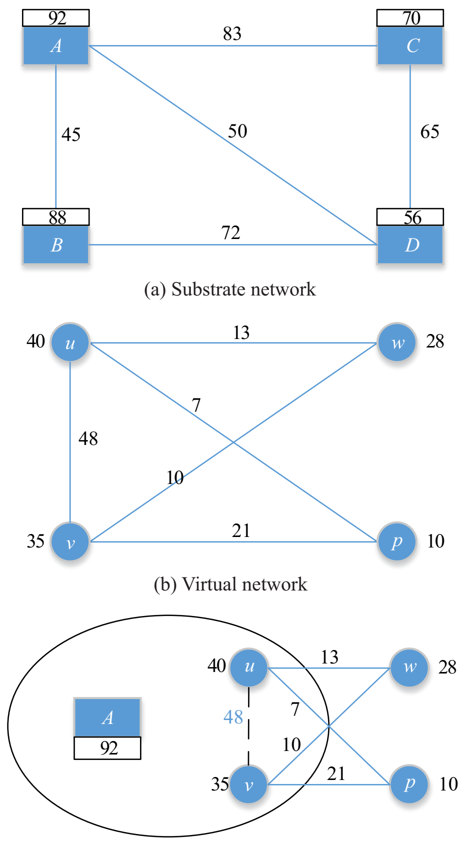
Input: A set of virtual nodes at a each time slot and the physical network $G_{\text{P}} = (N_{\text{P}}, L_{\text{P}})$;

Output: Map as many virtual nodes as possible into substrate network G_{P} while simultaneously minimizing the total energy cost;

```
Begin
for each time slot do
    for each unmapped virtual node i do
        compute W_cost(n_p) for each substrate node n_p;
        j ← arg min_{n_p ∈ V_P} {W_cost(n_p)};
        Map virtual node i to substrate node j;
        while node j can accommodate more nodes do
            MS ← the set of nodes mapped to node j.
            NS ← unmapped neighbors of nodes in MS;
            c ← the virtual node in NS demand maximal bandwidth;
            Map this virtual node c to substrate node j;
        end while
    end for
end for
```

The detailed algorithm is shown in algorithm `EH_Alg`. For simplicity, we assume that all virtual nodes arriving within a particular timeframe are assigned to the substrate network, otherwise the virtual network will be rejected. The algorithm runs are simulated over 500 time slots, with 10 virtual nodes arriving in each time slot.

We employ an example to show the substrate network, the virtual network, and the process of embedding. Figure 1a illustrates a substrate network, where the integer in each rectangle represents the available CPU resources that the physical nodes can offer, and the value of each link indicates the available bandwidth. Figure 1b shows a virtual network consisting of 4 virtual nodes u, v, w, and



(a) Substrate network

(b) Virtual network

(c) Embedding a neighbor node

Fig. 1 Example of virtual network embedding.

p. Each virtual node requires a certain amount of CPU resources, denoted by the value inside the nodes symbol. The value of the link between two virtual nodes indicates the amount of required communications.

Figure 1c illustrates how to allocate the virtual network into the substrate network using the proposed approach. In the first step, we select physical node A in the substrate network according to the calculated priority. Then, the algorithm assigns node u to the substrate node A. At this stage, node A has enough CPU resources to host additional virtual nodes, whether to assign one of u's neighbors to node A. Node v is mapped to substrate node A because there are many communication requests between node u and node v, so mapping them to the same substrate node can significantly reduce energy consumption. After virtual nodes u and v have been mapped to substrate node A, node A still has enough CPU resources to host other virtual nodes, as shown in Fig. 1c. The algorithm repeats this process and tries to map other nodes into substrate node A. If node A cannot host any more nodes, the algorithm will try to find another substrate node until all the virtual nodes are successfully embedded or the virtual network is rejected.

We now analyze complexity of algorithm `EH_Alg`. Since the algorithm `EH_Alg` is an online algorithm, we only analyze the time complexity of embedding a given virtual network $G' = (V', E')$, which contains $|V|$ virtual nodes and $|E|$ virtual links, into a given substrate network $G(V, E)$ consisting of $|V|$ physical nodes and $|E|$ physical links. Clearly, procedure `MAP` (i, j), as shown in Algorithm 2, runs in at most $O(|V'| \cdot (|E| + |V|))$ time. This is because, when mapping node i into node j, there are at most $|V'|$ edges needed to be mapped, as node i has at most $|V'| - 1$ neighbors, and mapping one edge takes at most $O((|E| + |V|))$ time. For each virtual network, there are $O(|V'|)$ nodes in total, and thus the time complexity is $O(|V'|^2 \cdot (|E| + |V|))$.

Algorithm 2 MAP(*i*, *j*)

Input: Virtual node i and the physical network $G_P = (N_P, L_P)$;
Output: Try to map each of node i to the substrate node j;
Begin
if j can host node i **then**
  map i into j, and update node j;
else
  reject the virtual network and update substrate network then break;
end if
while there is an unmapped link (i, k) for each mapped virtual node k **do**
  find a shortest path between host(i) and host(j);
  if there is no feasible path **then**
    reject the virtual network and update substrate network then break;
  else
    map the link (i, k);
    update information for all links on the path;
  end if
end while
End

4 Simulation

In this section, we evaluate the proposed approach by comparing it to existing algorithms. Same as the previous work, we use the GT-ITM tool to generate the substrate network topology consisting of 100 physical nodes and 570 physical links, which corresponds to a medium-sized ISP. For each substrate node, the CPU resources are set following a uniform distribution from 50 to 100. Similarly, for each substrate link, the bandwidth also follows this range from 50 to 100. Both algorithms take the same input substrate network and virtual networks, and both repeat over 500 time slots.

In order to evaluate our proposed algorithm, we compare the `EH_Alg` with algorithm `TR-CL` from Ref. [21]. Referring to the previous research, we set the values of C_b, C_m, and $CP(n_p)$ to be 150, 300, and 15, respectively. When collecting simulation results, both α and β are set to 0.5.

To compare the existing algorithms, we consider a certain status, named a "closely-saturated" state, which means the CPU resources of physical nodes are close enough to accommodate all the virtual nodes at a certain time slot. To evaluate this state, the CPU resources of the virtual nodes and the bandwidth of the virtual links are randomly distributed between 0 and 12. The evaluation results are shown in Figs. 2 – 6.

(1) The revenue is defined as $DCP(n_v) + DBR(l_v)$, which can be seen in Fig. 2. Our algorithm has great performance in terms of long-term average energy consumption, shown in Fig. 3.

(2) Our algorithm `EH_Alg` significantly exceeds the `TR-CL` algorithm in terms of its acceptance ratio. In Fig. 4, the acceptance ratios of `EH_Alg` and `TR-CL` are 100% and 89%, respectively. Specifically, when a certain substrate node is still superior to the other substrate nodes after mapping virtual node

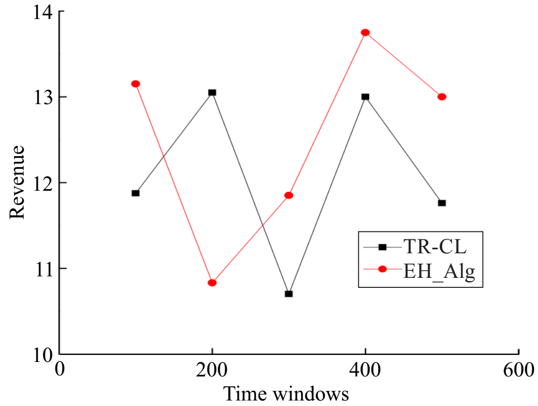


Fig. 2 Revenue of each time window.

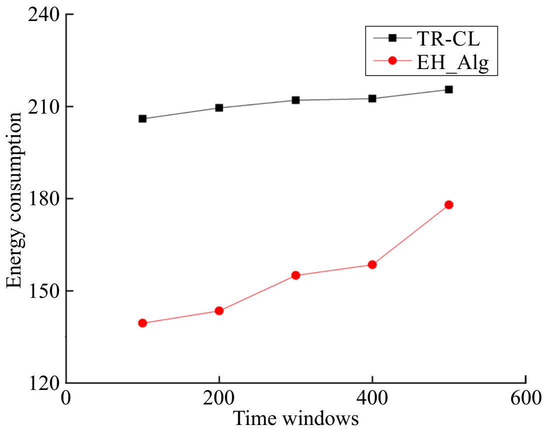


Fig. 3 Energy consumption of each time window.

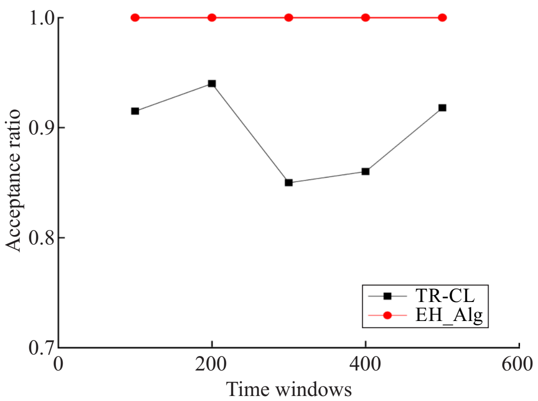


Fig. 4 Comparison of the two algorithms in terms of acceptance ratio.

u, our algorithm allows multiple virtual nodes to be mapped to the same substrate node. Compared to the existing algorithm, our proposed approach can significantly improve upon the previous work in terms of acceptance rate.

(3) In Fig. 5, we measure energy consumption of the proposed algorithm `EH_Alg` on substrate networks with varying energy consumption of substrate links,

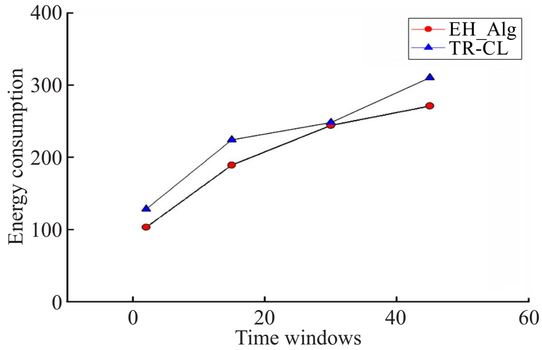


Fig. 5 Comparison of energy consumption.

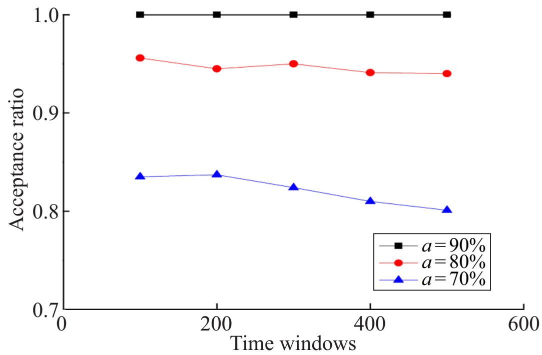


Fig. 6 Impact of changing parameter *a*.

i.e., link power. It's easy to see that the energy consumption of both algorithms increases with the value of $C_n(l_p)$. The total energy consumption contains both server (node) computation and data communication on certain links. As previously discussed, the energy consumption of data communication on links can be reduced by mapping closely related virtual nodes (i.e., nodes that have considerable communication) to the same substrate node. Keeping this in mind, our proposed approach considers the communication demand among virtual nodes when performing virtual network embedding, which clearly outperforms the existing algorithm in terms of energy consumption.

(4) Our principle is that multiple virtual nodes can be mapped into a certain substrate node when the load of that physical node is less than a certain threshold a. a is set to 70%, 80%, or 90%, as shown in Fig. 6. For example, when a is set to 90%, the substrate network can hold all virtual nodes.

5 Conclusion

Virtual network embedding is a crucial problem in network virtualization. In this paper, the energy efficiency of virtual network embedding while simultaneously minimizing total energy consumption is investigated. We model the VNE optimization problem as an Integer Linear Program that considers the impact of energy consumption on both nodes and links of a substrate network. We also propose a new heuristic algorithm to produce an efficient embedding solution based on the defined priority of substrate nodes and virtual nodes. Our experimental results verify the efficiency of the proposed algorithm.

Acknowledgment

This work was supported by the Specialized Research Fund for the Doctoral Program of Higher Education of China (No. 20131201110002), and the Key Laboratory of Computer Architecture Opening Topic Fund Subsidization (No. CARCH201303).

References

[1] A. Fischer, J. F. Botero, M. Till Beck, H. De Meer, and X. Hesselbach, Virtual network embedding: A survey, *IEEE Communications Surveys & Tutorials*, vol. 15, no. 4, pp. 1888–1906, 2013.

[2] P. Papadimitriou, O. Maennel, A. Greenhalgh, A. Feldmann, and L. Mathy, Implementing network virtualization for a future Internet, in *Proceedings of the 20th ITC Specialist Seminar on Network Virtualization — Concept and Performance Aspects*, 2009, pp. 1–8.

[3] N. M. K. Chowdhury and R. Boutaba, A survey of network virtualization, *Computer Networks*, vol. 54, no. 5, pp. 862–876, 2010.

[4] M. F. Bari, R. Boutaba, R. Esteves, L. Granville, M. Podlesny, M. Rabbani, Q. Zhang, and M. F. Zhani, Data center network virtualization: A survey, *IEEE Communications Surveys & Tutorials*, vol. 15, no. 2, pp. 909–928, 2013.

[5] X. Cheng, S. Su, Z. Zhang, K. Shuang, F. Yang, Y. Luo, and J. Wang, Virtual network embedding through topology awareness and optimization, *Computer Networks*, vol. 56, no. 6, pp. 1797–1813, 2012.

[6] L. Gong, Y. Wen, Z. Zhu, and T. Lee, Toward profit-seeking virtual network embedding algorithm via global resource capacity, in *IEEE INFOCOM 2014 — IEEE Conference on Computer Communications*, 2014, pp. 1–9.

[7] D. Anderson, Theoretical approaches to node assignment, http://www.cs.cmu.edu/dga/papers/andersen-assignabstract.html, 2002.

[8] J. F. Botero and X. Hesselbach, Greener networking in a network virtualization environment, *Computer Networks*, vol. 57, no. 9, pp. 2021–2039, 2013.

[9] M. Yu, Y. Yi, J. Rexford, and M. Chiang, Rethinking virtual network embedding: Substrate support for path splitting and migration, *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 2, pp. 17–29, 2008.

[10] N. M. K. Chowdhury, M. R. Rahman, and R. Boutaba, ViNEYard: Virtual network embedding algorithms with coordinated node and link mapping, *ACM/IEEE Transactions on Networking*, vol. 20, no. 1, pp. 206-209, 2012.

[11] J. F. Botero, X. Hesselbach, M. Duelli, D. Schlosser, A. Fischer, and H. de Meer, Energy efficient virtual network embedding, *IEEE Communication Letters*, vol. 16, no. 5, pp. 756–759, 2012.

[12] J. Lu and J. Turner, Efficient mapping of virtual networks onto a shared substrate, Technical Report WUCSE-2006-35, Washington University in St Louis, 2006.

[13] M. Yu, Y. Yi, J. Rexford, and M. Chiang, Rethinking virtual network embedding: Substrate support for path splitting and migration, *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 2, pp. 17–29, 2008.

[14] Y. Zhu and M. Ammar, Algorithms for assigning substrate network resources to virtual network components, in *Proceedings IEEE INFOCOM 2006. 25TH IEEE International Conference on Computer Communications*, 2006, pp. 1–12.

[15] S. Su, Z. Zhang, A. X. Liu, X. Cheng, Y. Wang, and X. Zhao, Energyaware virtual network embedding, *ACM/IEEE Transactions on Networking*, vol. 22, no. 5, pp. 1607–1620, 2014.

[16] A. Razzaq, P. Sjodin, and M. Hidell, Minimizing bottleneck nodes of a substrate in virtual network embedding, in *Proceedings of 2011 International Conference on the Network of the Future (NOF)*, 2011, pp. 35–40.

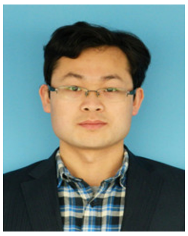
[17] Q. Hu, Y. Wang, and X. Cao, Resolve the virtual network embedding problem: A column generation approach, in *Proceedings of IEEE INFOCOM 2013*, 2013, pp. 410–414.

[18] Z. Xu, W. Liang, and Q. Xia, Efficient virtual network embedding via exploring periodic resource demands, in *Proceedings of 2014 IEEE 39th Conference on Local Computer Networks (LCN)*, 2014, pp. 90–98.

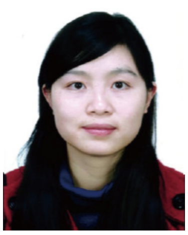
[19] X. Guan, B. Y. Choi, and S. Song, Energy efficient virtual network embedding for green data centers using data center topology and future migrationh, *Computer Communications*, 2015. doi:10.1016/j.comcom.2015.05.003.

[20] R. R. Oliveira, D. S. Marcon, L. R. Bays, M. C. Neves, L. P. Gaspary, M. P. Barcellos, and D. Medhi, Opportunistic Resilience Embedding (ORE): Toward cost-efficient resilient virtual networks, *Computer Networks*, 2015. doi:10.1016/j.comnet.2015.07.010.

[21] X. Chen, C. Li, Y. Jiang, Optimization model and algorithm for energy efficient virtual node embedding, *IEEE Communication Letters*, 2015. doi:10.1109/LCOMM.2015.2442575.

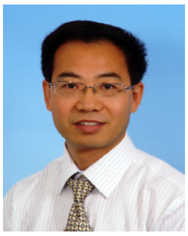


Guiyuan Jiang received the BS degree from Northwest University for Nationalities, China in 2008, the MEng degree from Tianjin Polytechnic University, China in 2011, and the doctoral degree from Tianjin University (TJU) in 2015, all in computer science and technology. He was an intern student at Nanyang Technological University (NTU) and National University of Singapore (NUS), respectively. Currently, he is working as a research fellow at School of Computer Science and Technology of NTU. His research interests include datacenter and cloud computing, fault-tolerant reconfiguration, hardware/software co-design, wireless sensor networks, and collaborative sensing for real-time traffic management.



Peilan He received the bachelor degree from Hainan University in 2012 and MEng degree in computer science and technology from Tianjin University, China in 2015. She is currently working towards the PhD degree at the School of Computer Science and Engineering, Nanyang Technological University, Singapore. Her research interests include intelligent transportation system, collaborative sensing for real-time traffic management, and data mining techniques for wireless sensor networks.

Shuxian Jia received the bachelor degree from Tianjin Polytechnic University, China, in 2015. Currently she is working towards the master degree at the School of Computer Software Technology, Zhejiang University (ZJU), China. Her research interests include datacenter and cloud computing, combinatorial optimization problems (designing efficient algorithm), and financial data analysis.



Jigang Wu received the BSc degree from Lanzhou University, China in 1983, and doctoral degree from University of Science and Technology of China (USTC) in 2000. He was with the Center for High Performance Embedded Systems, School of Computer Engineering, Nanyang Technological University, Singapore from 2000 to 2009, as a research fellow. He was with the School of Computer Science and Software Engineering as Dean and Tianjin distinguished professor, Tianjin Polytechnic University, China, from 2009 to 2014. Currently, he is with the School of Computer Science and Technology, Guangdong University of Technology. His research interests include reconfigurable VLSI design, hardware/software co-design, and combinatorial search.