Energy Efficient Virtual Network Embedding for Path Splitting

Xiaohua Chen ^{1,3} 1 Software Engineering Institute East China Normal University, Shanghai, China 200062 3 School of Information Engineering Huzhou Teachers College, Huzhou, China 313000 Email: 97122046@sina.com

Abstract—Multicommodity flow-based virtual network embedding algorithm does not consider link energy, which causes waste of energy. And its high time complexity can not meet real-time requirements of online virtual network embedding. In this paper, we find the dynamic inversion phenomenon, where revenue does not rely on embedding cost. Two novel link mapping algorithms are proposed for path splitting which based on the undirected network minimum cost flow. They enable link resource to consolidate and have low time complexity. Simulation results show that proposed algorithms reduce energy consumption and ensure real-time performance of online VN embedding.

Keywords—virtual network embedding, network virtualization, energy efficient, minimum cost flow, path splitting.

I. INTRODUCTION

Current networks are designed for peak loads, which cause a high resource under-utilization and unnecessary energy consumption. For example, the average link utilization in backbone networks of large ISPs is estimated to be around 30-40% [1]. The average server utilization in data centers is about 11-50% [2,3]. Low utilization causes a huge waste of energy.

Network virtualization is an important technology in future Internet, cloud computing and software-defined networks. It will be an enabler for intelligent energy-aware network deployment by means of resource consolidation. Little attention has been paid to reducing the energy consumption for path splitting. The link-embedding problem for path splitting can be reduced to the multicommodity flow (MCF) problem. However, multicommodity flow-based virtual network embedding (VNE) algorithm [4] does not consider link energy, which causes unnecessary energy consumption. Because of a high time complexity, it cannot meet real-time requirements of online VNE. In [5], Zhang et. present an algorithm of link embedding, which is based on greedy k-shortest paths for path splitting. However, it comprises revenue of substrate network.

We study dynamic characteristics of VNE, and find the dynamic inversion phenomenon, where revenue doesn't depend on mapping cost. We also propose two novel link mapping algorithms, which are based on the undirected network minimum cost flow (UNMCF) by means of link resource consolidation. Extensive simulation experiments show that the proposed algorithms outperform their counterparts in terms of acceptance ratio, revenue, running time and energy consumption.

Chunzhi Li 2,3

2 Department of Computer Science and Technology East China Normal University, Shanghai, China 200241 3 School of Information Engineering Huzhou Teachers College, Huzhou, China 313000 Email: lichunzhi82@126.com

We make the following major contributions in this work: i) We find the dynamic inversion phenomenon occurs in the dynamic process of VNE. ii) We propose two novel algorithms for embedding virtual links, which are based on UNMCF and optimization of link resource consolidation. iii) We carry out simulations to evaluate the performance of our algorithms and the state-of-the-art algorithm.

The rest of this paper is organized as follows. Section II discusses the related work. Section III describes the dynamic inversion phenomenon, and presents UNMCF and two link embedding algorithms. We evaluate our algorithms in Section IV. Section V concludes the paper.

II. RELATED WORK

VNE is a major issue in network virtualization. Multiple VNE algorithms have already been discussed in [6]. Most of them are cost efficient algorithms. Performance is evaluated by a number of different parameters, e.g. cost, revenue, acceptance ratio, running time and load balance. There are several papers that focus on energy efficient embedding. Using the consolidation technology, network virtualization enables the energy to use network infrastructure efficiently [7-12]. In [13], Garroppo et al. provide an approach which spreads the traffic among all network resources and is especially suitable for load-sensitive equipment.In [14], Su et al. propose an energy cost model and design two energy-aware algorithms to reduce the energy cost. Our work is different from those early results. In contrast to the existed literatures, We study how to reduce energy spending in the context of substrate network that supports path splitting.

III. ENERGY EFFICIENT LINK EMBEDDING

A. Power Consumption Model

In server systems, Power consumption mainly includes processor, memory, disk I/O, fan, network ports and others. We assume energy consumption for substrate network by node and link energy consumption.

Node power consumption model. The CPU utilization is the main contributor to the power consumption variations of a server. We define node i power consumption model PN^i as:

$$PN^{i} = \begin{cases} P_{b} + P_{l} \cdot \mu, & \text{if } s(i) = 1\\ 0, & \text{if } s(i) = 0 \end{cases},$$
(1)

where P_b is the server's baseline power, P_m is the total power when serving at the maximum capacity, μ is CPU utilization, $P_l = P_m - P_b$ represents the energy proportion factor for μ , s(i) = 1 means node *i* active, and s(i) = 0 means node *i* turned off.

Link power consumption model. For current power consumption insensitiveness of network equipment to traffic load, the power consumption of an engine, denoted by P_n , is nearly a constant regardless whether the ports are idle or carrying full speed traffic. Whenever a link is turned off, energy is saved in the pair of interfaces/ports on its endpoints. So we define link *j* power consumption model PL^j as:

$$PL^{j} = \begin{cases} P_{n}, \text{ if } s(j) = 1\\ 0, \text{ if } s(j) = 0 \end{cases},$$
(2)

where P_n is the link's baseline power, s(j) = 1 means link j active, and s(j) = 0 means link j turned off.

B. Dynamic Inversion Phenomenon

VN dynamical characteristics include arrivals and duration of VN, the number of VN nodes and links, node CPU and link bandwidth requirements of VN. With arrival and department of VN, substrate network topology, node remaining CPU and link remaining bandwidth will change dynamically.

Dynamic inversion phenomenon occurs in VNE. It's common that lower VNE cost saves more room for the future VNs, which produces higher revenue and accepts more VNs. However, because bigger VNs are refused for saving more room for smaller VNs, higher VNE cost produces higher acceptance ratio and more revenue. This is the dynamic inversion phenomenon.

C. Notations

Undirected Networks. Let N = (V, E) be an undirected network, where V is the set of nodes and E is the set of links. Each link $(u, v) \in E$ has an associated capacity $c_{u,v}$.

Undirected Network Flow. In N = (V, E), there are a pair of nodes $s, t \in V$, called *source* and *sink* respectively. f(u, v) is a flow on the link (u, v), where $\forall (u, v) \in E:$ (i) $f(u, v) \leq c(u, v)$, (ii)f(u, v) = -f(v, u), and (iii) $\sum_{u \in V, v \in V} f(u, v) = 0$ iff. $(u \notin \{s, t\}, v \notin \{s, t\})$. f_s^t is the undirected network flow from s to t, where $|f_s^t| = \sum_{(s, v) \in E} f(s, v) = \sum_{(v, t) \in E} f(v, t)$.

UNMCF. In N = (V, E), each link $(u, v) \in E$ has an associated cost b(u, v) that denotes the cost per unit flow on the link (u, v). f_s^t is the undirected network flow. The undirected network minimum cost flow problem is an optimization model which is formulated as follows:

$$\operatorname{Minimize}_{(u,v)\in E} f(u,v) \cdot b(u,v) \tag{3}$$

Subject to

$$\sum_{(s,v)\in E} f(s,v) = \sum_{(v,t)\in E} f(v,t) = |f_s^t|,$$
(4)

$$\sum_{(u,v)\in E} f(u,v) = 0, if(u,v \notin \{s,t\}).$$
(5)

D. Energy Efficient Link Mapping

We use UNMCF and propose two energy efficient link embedding algorithms with substrate network that supports path splitting.

Algorithm 1 and 2 create UNMCF model and find one minimum cost flow for a virtual link, until all virtual links have been mapped successfully. Based on the link resource consolidation, the cost per unit flow b_{ij} on link (i, j) is defined by the residual bandwidth bw_{ij} on the substrate link (i, j). Hence, substrate link fragmentation should be used as much as possible. In Algorithm 1 and 2, virtual link selected method is different. Algorithm 1 randomly selects a virtual link to be mapped. Algorithm 2 selects a virtual link with maximum bandwidth to be mapped, where the ordering strategy helps to decrease link mapping cost.

Since the time complexity of UNMCF is $O(n^2)$, the time complexity of Algorithm 1 and 2 are $O(e \cdot n^2)$, where n is the number of substrate nodes and e is the number of virtual links.

gorithm 1 Energy efficient link of	6
put: Virtual network G^v , substrate	e network G^s ,
utput: Link embedding output.	
foreach virtual link $(n_1^v, n_2^v) \in I$	L^v do
Create UNMCF model from G	's as
$N:V = N^s, E = L^s;$	
$C: \forall i, j \in N^s, c_{ij} = bw_{ij};$	
$B: \forall i, j \in N^s, b_{ij} = bw_{ij};$	
Find mapped substrate node n_1^s	$n_1^s, n_2^s \text{ of } n_1^v, n_2^v;$
if (Find minimum cost flow f)	
Update remaining link bandy	width;
else return LINK_FAIL; end if	n [
end foreach	
return LINK_SUCC;	

Algorithm 2 Energy efficient sorted link embedding		
Input: Virtual network G^v , substrate network G^s ,		
Output: Link embedding output.		
1: while (Find $(n_1^v, n_2^v) \in L^v$ with max bandwidth) do		
2: Create UNMCF model from G^s as		
$N:V = N^s, E = L^s;$		
$C: \forall i, j \in N^s, c_{ij} = bw_{ij};$		
$B: \forall i, j \in N^s, b_{ij} = bw_{ij};$		
3: Find mapped substrate node n_1^s, n_2^s of n_1^v, n_2^v ;		
4: if (Find minimum cost flow f)		
5: Update remaining link bandwidth;		
6: else return LINK_FAIL; end if		
7: end while		

8: return LINK SUCC;

IV. PERFORMANCE EVALUATION

A. Saturated and Non-saturated State

The performance varies in saturated and non-saturated state. In non-saturated state, the remaining resources of substrate network are enough to accept all VNs. Minimizing Energy consumption is the major object. In saturated state, the remaining resources of substrate network are limited and some VNs will be refused. Maximizing the revenue is the major object.

TABLE I. COMPARED ALGORITHMS

Notation	Algorithm Description
CB-MM-MCF	It is a two-stage embedding algorithm. It employs greedy method to map virtual nodes, then embeds the virtual links by MCF model.
CB-MM-UNMCF	It is a two-stage embedding algorithm. It employs greedy
	method to map virtual nodes, then embeds the virtual links
	by UNMCF model (Algorithm 1).
CB-SM-UNMCF	It is a two-stage embedding algorithm. It employs greedy
	method to map virtual nodes,, then embeds the sorted virtual
	links by UNMCF model (Algorithm 2).

B. Performance Metrics

The metrics considered include (i)the long-term energy consumption of substrate network in a time unit, given by $\lim_{T\to\infty} \frac{\sum_{t=0}^{t=T} (\sum PN^i + \sum PL^j)}{T \cdot T_n}$, where T is a time window and T_n is time unit in a time window. (ii)The long-term average number of hibernating links of substrate network in a time window. (iii)The long-term average number of hibernating nodes of substrate network in a time window. (iv)the long-term revenue, given by $\lim_{T\to\infty} \frac{\sum_{t=0}^{t=T} R(G^v(t))}{T}$, where $R(G^v(t))$ is the revenue of serving VNs at time t. And $R(G^v(t)) = \sum bw(l^v) + \sum CPU(n^v)$, where $bw(l^v)$ and $CPU(n^v)$ are the bandwidth and CPU requirements for the virtual link l^v and the virtual node n^v , respectively. (v)The long-term VN acceptance ratio. (vi)The long-term revenue to cost ratio, given by $r/c = \lim_{T\to\infty} \frac{\sum_{t=0}^{t=T} R(G^v(t))}{\sum_{t=0}^{t=T} C(G^v(t))}$, where $C(G^v(t))$ is the cost of serving VNs at time t. And $R(r^v)$, $r/c = \lim_{T\to\infty} \frac{\sum_{t=0}^{t=T} R(G^v(t))}{\sum_{t=0}^{t=T} C(G^v(t))}$, where $C(G^v(t))$ is the cost of serving time t. And $C(G^v(t)) = \sum bw(l^v) + \sum CPU(n^v)$, $r/c = \lim_{T\to\infty} \frac{\sum_{t=0}^{t=T} R(G^v(t))}{\sum_{t=0}^{t=T} C(G^v(t))}$, where $C(G^v(t))$ is the cost of serving time t. And $C(G^v(t)) = \sum bw(l^v) + \sum CPU(n^v)$, $r/c = \lim_{T\to\infty} \frac{\sum_{t=0}^{t=T} R(G^v(t))}{\sum_{t=0}^{t=T} C(G^v(t))}$.

VNs at time t. And $C(G^{v}(t)) = \sum bws(t^{v}) + \sum CPU(n^{v})$, where $bws(t^{v})$ is substrate bandwidth sum for the virtual link t^{v} . (vii)Running time.

C. Evaluation Environment

Although network virtualization is an emerging field, the actual characteristics of substrate networks and VN requests are still not well understood. Therefore, we use synthetic network topologies to evaluate the proposed algorithms.

Substrate network. The running time of MCF-based link mapping algorithm is affected by the size of virtual network and substrate network. Here, we use the GT-ITM tool to generate a small size of a substrate network topology that is configured to have 50 nodes and around 140 links. Each pair of substrate nodes are connected with probability 0.5. The CPU resources at nodes and the bandwidths at links follow a uniform distribution between 50 and 100.

Virtual network request. The arrivals of VN requests are modeled by a Poisson process. Each pair of virtual nodes are randomly connected with probability 0.5. The waiting time of each VN is set to one time window. The duration of VNs follows an exponential distribution with 5 time windows on average. A time window is equal to 100 time unit. In order to evaluate performance in the saturated and non-saturated state, two groups of VNs are created. In the non-saturated state, we run all the simulations in 250 time windows with mean 20 VNs per time window, which correspond to about 5000 VNs on average, the number of VN nodes is randomly determined by a uniform distribution between 2 and 8, and the CPU and bandwidth requirements of virtual nodes and links are real

number uniformly distributed between 0 and 4. In the saturated state, we run all the simulations in 500 time windows with mean 10 VNs per time window, which correspond to about 5000 VNs on average, the number of VN nodes is randomly determined by a uniform distribution between 2 and 20, and the CPU and bandwidth requirements of virtual nodes and links are real number uniformly distributed between 0 and 40. Ten different instances are run for the algorithms and we record the arithmetic mean of the ten runs as the final result.

Comparison method. Our simulation experiments evaluate three algorithms listed in Table 1. We primarily focus on quantifying the benefits of substrate support for path splitting, which is based on UNMCF model in the saturated and non-saturated state. And we compare the solutions by MCF model. Since the revenue and acceptance ratio of the algorithm proposed in [5] are lower than MCF-based algorithm, we do not compare our algorithms with the algorithm in [5].

We set P_b , P_m and P_n defined in Section II-B to 150W, 300W and 15W, respectively, which are similar as the previous work [10].

D. Evaluation Results in Non-saturated State

1) Our algorithms outperforms MCF-based in terms of the long-term average energy. From Fig. 1(a), we can see that UNMCF-based algorithms consume less energy than MCF-based algorithm in the long run. This is due to the fact that the UNMCF-based algorithms utilize the residual bandwidth fragmentation on the substrate link, and enable link resource consolidation which results in switching off or hibernating more links and nodes (shown in Fig. 1(b) and 1(c)).

2) Our algorithms decrease cost of the embedding. Fig.2(a) shows the result of different algorithms, where revenue to cost ratio of UNMCF-based increases 6% than MCF-based solution. This is due to sufficient resources in the non-saturated state, and UNMCF-based algorithms can minimize the embedding cost.

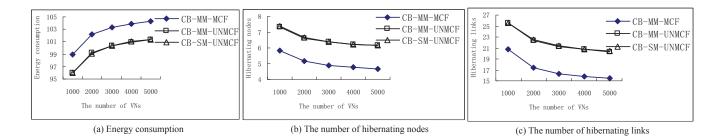
E. Evaluation Results in Saturated State

1) Dynamic inversion phenomenon occurs in saturated state. Fig.2(b), 3(b) and 3(c) show that the dynamic inversion phenomenon occurs in saturated state, where the UNMCF-based algorithms outperform the MCF-based algorithm in term of the long-term acceptance ratio and revenue, but the UNMCFbased algorithms decrease revenue to cost ratio.

2) Trade-off between energy savings and revenue. In the saturated state, the UNMCF-based algorithms increase energy consumption while increasing acceptance ratio and revenue (shown in Fig. 3(a), Fig. 3(b) and Fig. 3(c)). The UNMCF-based algorithms accept more VNs, which enhance the utilization of substrate links and nodes, and consequently consume more energy than MCF-based algorithm.

F. Running Time Comparison

The UNMCF-based algorithms take less time than the MCFbased algorithm, since the time complexity of UNMCF-based algorithms is much lower than MCF-based algorithm. Fig.2(c) depicts the average running time of these VN embedding algorithms in saturated state, which is on the same server with Intel 2.6GHz dual-core CPU and 8GB memory.





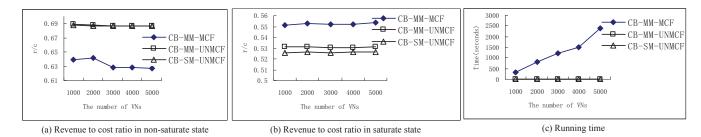


Fig. 2. Other performance

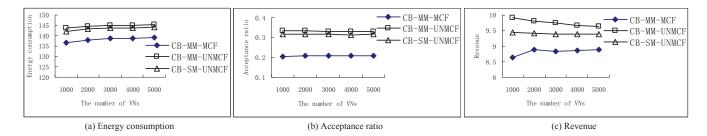


Fig. 3. Performance in the saturated state.

V. CONCLUSION

This paper studies the energy efficient VNE for path splitting. We analyze the dynamic characteristics and propose the UN-MCF model for embed virtual links with the substrate network supporting path splitting. Two UNMCF-based VNE algorithms are proposed to embed virtual links. The experimental results demonstrate that the inversion phenomenon occurs in saturated state, the proposed algorithms suit for online VNE and increase energy savings in the non-saturated state. In future work, we intend to develop energy-aware re-allocations of VNE for path splitting to increase energy savings. With the development of substrate energy saving technology, the energy-aware VNE is worthy of further study.

REFERENCES

- W. Fisher, M. Suchara, and J. Rexford. Greening backbone networks: reducing energy consumption by shutting off cables in bundled links. ACM SIGCOMM workshop on Green networking, 2010:29-34.
- [2] LA. Barroso, U. Holzle. The case for energy-proportional computing. IEEE Computer, 2007, 40(12):33-37.
- [3] P. Bohrer, EN. Elnozahy, T. Keller, et al. The case for power management in web servers. Power aware computing. Springer US, 2002:261-289.
- [4] M. Yu, Y. Yi, J. Rexford, et al. Rethinking virtual network embedding: substrate support for path splitting and migration. Proceedings of Computer Communication Review, 2008,38(2):17-27.

- [5] Z. Zhang, X. Cheng, S. Su, et al. A unified enhanced particle swarm optimization-based virtual network embedding algorithm. International Journal of Communication Systems, 2013,26:1054-1073.
- [6] A. Fischer, JF. Botero, BM. Till, et al. Virtual network embedding: A survey. Communications Surveys & Tutorials, IEEE, 2013,15(4):1888-1906.
- [7] JF. Botero, X. Hesselbach, M. Duelli, et al. Energy efficient virtual network embedding. IEEE Communications Letters, 2012,16(5):756-759.
- [8] XL. Chang, B. Wang, JQ. Liu, et al. Green cloud virtual network provisioning based ant colony optimization. Proc. of the 15th Annual Conf, 2013:1553-1560.
- [9] JF. Botero, X. Hesselbach. Greener networking in a network virtualization environment. Computer Networks, 2013,57(9):2021-2039.
- [10] S. Su, ZB. Zhang, X. Cheng, et al. Energy-Aware virtual network embedding through consolidation. Proc. of the Computer Communications Workshops, 2012:127-132.
- [11] B. Wang, XL. Chang, JQ. Liu, et al. Reducing power consumption in embedding virtual infrastructures. Proc. of the Globecom Workshops, 2012:714-718.
- [12] X. Chen, C. Li, L. Chen, Z. Zeng. Energy Efficient Virtual Network Embedding Based on Actively Hibernating Substrate Nodes and Links. Journal of Software, 2014, 25(7):1416-1431(in Chinese)
- [13] R. Garroppo, G. Nencioni, L. Tavanti, et al. Does traffic consolidation always lead to network energy savings. IEEE Communication Letters, 2013,17(9):1852-1855.
- [14] S. Su, Z. Zhang, A.X. Liu, et al. Energy-Aware Virtual Network Embedding. IEEE/ACM Transactions on Networking. 2014.10:1-14.