Journal of Theoretical and Applied Information Technology
10th February 2013. Vol. 48 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

HIDDEN HOPS AWARE LOAD BALANCING BASED ON GREEDY APPROACH

[1]BIN-BIN HUANG, [2]RONG-HENG LIN, [3]HONG-XIA ZHANG, [4]HUA ZOU, [5]FANG-CHUN YANG

State Key Laboratory of Networking and Switching Technology,
Beijing University of Posts and Telecommunications, Beijing 100876 China
Email: { huangbinbin[1], rhlin[2] , zouhua[4], fcyang[5]}@bupt.edu.cn, zhanghx103@gmial.com

ABSTRACT

Virtual network mapping is one of the main problems in network virtualization. At present, virtual networking mapping aims at minimal resource consumption at substrate network, but ignores the resource demanded by the hidden hops, making bottlenecks due to the resource shortage at the hidden hops. This paper aims at the simultaneous loading balance of the substrate node and the substrate link, mathematically formulates the virtual networking mapping problem constrained by hops, and solves it by using greedy algorithm. Our experiments show that this algorithm eliminates resource bottleneck efficiently, provides a more balanced substrate network for the request of the consequential virtual network request, thus improving the constructing success rate of virtual network, the availability of network resources and the profits of the infrastructure providers.

Keywords: *Virtual Network Mapping, Hidden Hops, Load Balancing, Greedy Algorithm.*

1. INTRODUCTION

Virtualization of network resource has been identified as a key technology for Future Internet research [1] and is actively used in current research testbeds [2][3]. By virtualizing both nodes and link resources of a substrate network, multiple virtual network topologies with widely varying characteristics can be created and cohosted on the same physical hardware. It makes cloud providers to gain economical revenues from underutilized physical resources [4]. However, applying virtualization of network resources leads to the problem of mapping virtual resources to physical resources, known as “virtual network mapping”. The so-called virtual network mapping can be viewed as the assignment of the resources from a virtual network to the components of a physical network.

Existing researches [5-12] classify it as a constraint optimization problem with the objective of minimizing mapping cost. But they neglect the resource consumption of all hidden hops on the paths, which results in bottleneck because of insufficient hidden hops’ resource. Because mapping a virtual link to a path in the Substrate obviously uses resources of the substrate links on the path. However, there are also physical nodes on the path that will be traversed by the virtual link. These are called “Hidden Hops” here. The virtual link will also consume resources of all Hidden Hops on the paths. Furthermore, several virtual links can use the same physical link. The bottleneck influences the performance of the whole substrate network and the request success rate of the consequential virtual network. In previous work [13], the important and realistic concept of hidden hops is introduced in virtual network mapping. The required demand of the hidden hops helps to carry out a more realistic virtual network mapping, because more virtual networks can be mapped taking into account the CPU demand of hidden hops. However, this work consideres the offline version of the algorithm, and the virtual node mapping is not considered.

In [14], the objective is to balance the load of substrate links. This paper formulates it to be a Multicommodity-Flow problem and proposes an algorithm to solve the virtual network mapping problem based on optimization theory. But this paper neglects the CPU resource that must be assigned to an intermediate node and doesn’t focus on the nodes’ load balancing.

In [15], balanced link load and balanced node load virtual network construction algorithm are given, respectively. Based on these, two algorithms

Balanced Adaptive virtual network Construction Algorithm (BACA) is proposed. But, in this paper, there are two issues as follows: firstly, the virtual nodes mapping in the virtual network request is determined in advance, which simplifies the virtual network mapping problem to the virtual link mapping problem; Secondly, this paper does not consider the CPU resource that must be assigned to an intermediate node on the path that will be traversed by the virtual link.

Addressing such a problem in online scenario, we consider the CPU resource that must be assigned to an intermediate node. To remove efficiently the resource bottlenecks due to the insufficient hidden hops' resource, we aim at the simultaneous loading balance of the substrate node and the substrate link, formulate the virtual networking mapping problem constrained by hops, and propose a Load balancing Greedy (LB-Greedy) algorithm. Moreover, we conduct a numerical comparison between our algorithm and BACA. The simulation experiment shows that there are efficiencies about LB-Greedy as follows: first, after getting rid of the resource expense of all intermediate nodes, it is able to remove efficiently the resource bottleneck with the goal of balanced link load and balanced node load. It will provide a more balanced substrate network for the consequential virtual network request, thereby improving the constructing success rate of virtual network, the availability of network resources and the profits of the infrastructure providers. Second, limiting the hop counts on the path that will be traversed by the virtual link will use as less substrate resource as possible. So, it can map as many virtual network requests as possible in the limited substrate networks and maximize the profit of infrastructure providers.

The rest of this paper is organized as follows. Following that, section 2 formalizes the virtual network mapping problem and presents the performance metrics. In Section 3, we present the LB-Greedy algorithm. In section 4 we describe the simulation setting and present simulation results that evaluate the proposed algorithms, and we conclude and identifying future research directions in section 5.

2. VIRTUAL NETWORK MAPPING AND PERFORMANCE METRICS

2.1 Virtual Network Mapping Model

For convenience, we abstract the virtual network mapping problems to be the graph theory problems, and use a weighted undirected graph $G_s=(N_s,E_s)$ to present the substrate network. The whole CPU resource of the physical node $n_s^i \in N_s$ is denoted by $C\left(n_s^i\right)$, and the available resource is denoted by $c\left(n_s^i\right)$. The whole bandwidth of the substrate link $e\left(n_s^i,n_s^j\right) \in E_s$ is denoted by $B\left(e\left(n_s^i,n_s^j\right)\right)$, and the available bandwidth is denoted by $b\left(e\left(n_s^i,n_s^j\right)\right)$.

Similarly, a weighted undirected graph $G_v=(N_v,E_v)$ presents the virtual network which is a sub-graph of the substrate network. The CPU resource demand of virtual node $n_v^i \in N_v$ is denoted by $c\left(n_v^i\right)$, the bandwidth demand of virtual link is denoted by $b\left(e\left(n_v^i,n_v^j\right)\right)$. Mapping a virtual link to a path in the Substrate obviously uses resources of the substrate links on the path, but the virtual link will also consume resources of all Hidden Hops on the path. Here we denote the CPU resource expense of an intermediate node by $m(n_v^i,n_v^j)$, which makes the virtual network mapping more reasonable.

2.1.1 Resource consumption model of hidden hops

The intermediate nodes on the substrate path need to be configured and correctly forward the packets passing the virtual link, thereby they will have a CPU expenditure. And how much CPU resource will be expended depends on the bandwidth demand $b\left(e\left(n_v^i,n_v^j\right)\right)$ of virtual link and characteristic (like frequency of node, etc) of the substrate node. Then, we will formulate the CPU resource demand of an intermediate node as follows:

Suppose Φ Gbps=100 BW units, Γ GHz=100 CPU units, the packet's size is PS bytes, CPU cycle which is needed to forward such size packet is ω Cycle. The CPU resource expense of an intermediate node can be calculated as follows:

$$m(n_v^i,n_v^j)=\frac{\mathrm{b}\left(e\left(n_s^i,n_s^j\right)\right)\times\Phi\times10^9 \Big/ 8\times100\times PS}{\Gamma\times10^9 \Big/ \omega} \quad (1)$$

Subject to:

$$c\left(n_s^l\right) > \sum_{m=1}^{N} x_l^m c(n_v^m) + \sum_{e\left(n_v^i, n_v^j\right)\in E_v} \delta\left(n_v^i, n_v^j, n_s^l\right) m\left(n_v^i, n_v^j\right) \quad (2)$$

$$f_{ij}^{mt} R\left(p\left(n_v^m, n_v^t, n_s^i, n_s^j\right)\right) - b\left(e\left(n_v^m, n_v^n\right)\right) > 0 \quad (3)$$

$$\delta\left(n_v^i, n_v^j, n_s^l\right) m\left(n_v^i, n_v^j\right) > c(n_s^l) \quad (4)$$

$\delta\left(n_v^i, n_v^j, n_s^l\right)$ is a binary variable. It is 0 if the substrate node $n_s^l \in N_s$ is an intermediate node of the path that will be traversed by the virtual link $e\left(n_v^i, n_v^j\right)$. It is 0 elsewhere. $p\left(n_v^m, n_v^t, n_s^i, n_s^j\right)$ presents the substrate path that will be traversed by the virtual link $e\left(n_v^m, n_v^t\right)$. And the residual bandwidth of $p\left(n_v^m, n_v^t, n_s^i, n_s^j\right)$ is denoted by $R\left(p\left(n_v^m, n_v^t, n_s^i, n_s^j\right)\right)$. Constrain set (2) and (3) contains the CPU capacity for substrate node and the bandwidth capability for substrate path $p\left(n_v^m, n_v^t, n_s^i, n_s^j\right)$. They assure the sum of the CPU and the bandwidths assigned to each virtual node and link, and it will not exceed substrate's node or link capacity. Constrain (4) constrains the CPU resource of an intermediate node of a path that will be traversed by a virtual link.

2.1.2 Load formulation for substrate node and substrate link

According to the statistics and analysis, it can be drawn that a more balanced substrate network can improve the constructing success rate for the consequential virtual network request, which can increase resource utilization for substrate network. So, during the mapping process, it needs to assure the balanced network load among substrate links, and also the balanced CPU load among substrate nodes, which can improve the constructing success rate for the consequential virtual network request and make efficient use of the underlying resources. The substrate node acts both the work node performing user's task and a hidden hop. A hidden hop will have a CPU expenditure because it has to be configured and it will have to correctly forward the packets passing through this virtual link. We denote the substrate node's CPU load as follows:

$$N(n_s^l) = \frac{\sum_{m=1}^{N} x_l^m c(n_v^m) + \sum_{e\left(n_v^i, n_v^j\right)\in E_v} \delta\left(n_v^i, n_v^j, n_s^l\right) m\left(n_v^i, n_v^j\right)}{c(n_s^l)} \quad (5)$$

The mean value among the M substrate nodes' CPU load is denoted by N_{avg}:

$$N_{avg} = \frac{\sum_{l=1}^{M} N\left(n_s^l\right)}{M} \quad (6)$$

In the same way, we denote the network load among substrate links as follows:

$$L\left(e\left(n_s^i, n_s^j\right)\right) = \frac{\sum_{e\left(n_v^m, n_v^t\right)\in E_v} f_{ij}^{mt} b\left(e\left(n_v^m, n_v^t\right)\right)}{b(e\left(n_s^i, n_s^j\right))} \quad (7)$$

The mean value for the network load among substrate links is denoted by L_{avg}:

$$L_{avg} = \frac{\sum_{e\left(n_s^i, n_s^j\right)\in E_s} L\left(e\left(n_s^i, n_s^j\right)\right)}{\left|E_s\right|}$$

(8)

The standard deviation for the CPU load among substrate nodes is denoted by N_σ:

$$N_\sigma = \frac{\sqrt{\sum_{i=1}^{M}\left(N\left(n_n^i\right) - N_{avg}\right)^2}}{M} \quad (9)$$

The standard deviation for the network load among substrate links is denoted by L_σ:

$$L_\sigma = \frac{\sqrt{\sum_{e\left(n_s^i, n_s^j\right)\in E_s}\left(L\left(e\left(n_s^i, n_s^j\right)\right) - L_{avg}\right)^2}}{\left|E_s\right|}$$

(10)

To avoid hot spots and improve the consequential virtual network request acceptance ratio, the objective of the virtual network mapping problem is to maintain balanced stress among all substrate nodes and substrate links:

$$Minimum\ \alpha N_\sigma + \beta L_\sigma \quad (11)$$

Where α and β are used to adjust the weight of link and node load balancing, respectively.

2.2 Virtual Network Mapping Model

To evaluate the stress balancing performance, we mainly define three performance metrics as follow:

Definition 1: The virtual network request acceptance ratio of the substrate network can be defined by

$$\partial_{success\_ratio} = \frac{Rq_{success}}{Rq_{total}} \tag{12}$$

Where $Rq_{success}$ is the number of virtual network requests successfully accepted by the substrate network Rq_{total} is the total number of virtual network requests.

Definition 2: The loading balance degree among substrate network can be denoted by:

$$D = \alpha N_{\sigma} + \beta L_{\sigma} \tag{13}$$

Definition 3: The average of node load and the average of link load can be denoted, respectively, as follows:

$$\mu_{node} = \frac{\sum_{l=1}^{M} N\left(n_s^l\right)}{M} \tag{14}$$

$$\mu_{link} = \frac{\sum_{e\left(n_s^i, n_s^j\right)\in E_s} L\left(e\left(n_s^i, n_s^j\right)\right)}{\left|E_s\right|} \tag{15}$$

3. PROPOSAL: LOAD BALANCING GREEDY ALGORITHM

Since the virtual network mapping with the constraints has been proved to be a NP in [5]. And finding an optimal virtual network mapping for loading balance by using mixed integer linear programming (MILP) is computationally intractable. We propose efficient heuristics to solve the problem. LB-Greedy is a greedy algorithm with heuristics for mapping the virtual network requests to the substrate network, while trying to balance the load of the whole substrate network after getting rid of the resource expense of all intermediate nodes. Given below is the detail of LB-Greedy.

Before describing the detail of LB-Greedy, the resource demands of a virtual node and a virtual link will be firstly defined.

Definition 4: The resource demand of a virtual node can be defined as follows:

$$H\left(n_v^u\right) = c\left(n_v^u\right) \sum_{l\in L\left(n_v^u\right)} b(l)$$

Where, for a virtual network request, $L\left(n_v^u\right)$ is the set of all the unmapped outgoing links of n_s^u, $b(l)$ is the bandwidth demand of the virtual link l which is unmapped. $c\left(n_s^u\right)$ is the CPU demand of n_s^u. According to the definition above, we define the resource demand of a virtual link $e\left(n_v^i, n_v^j\right)$ as follows:

$$H\left(e\left(n_v^i, n_v^j\right)\right) = c\left(n_v^i\right) \sum_{l\in L\left(n_v^i\right)} b(l) + c\left(n_v^j\right) \sum_{l\in L\left(n_v^j\right)} b(l)$$

LB-Greedy is a virtual network mapping algorithm coordinating node and link. The strategy for virtual link mapping sees the Algorithm 1:

Algorithm 1 LB-Greedy

```
1: Parameter Initialization: initializes the parameter $M^{*} = \varnothing$ and $M \sim = M$ . $M^{*}$ and $M \sim$ are the sets of mapped links and unmapped links in $G_v$, respectively.
2: While $M \sim \neq \varnothing$ do
3: Choose the virtual link with the largest resource demand $H\left(e\left(n_v^i, n_v^j\right)\right)$ in $M \sim$
4: If both the source node $Snode$ and the destination node $Enode$ of the virtual link are unmapped, they will be mapped as follows:
    4.1: For $Snode$ , find the unallocated substrate node with the largest available resource $H\left(n_s^u\right)$ for it.
    4.2: Find a subset of candidate substrate nodes for $Enode$ . The candidate substrate node must meet with the conditions as follows:
        a) The substrate node is unallocated
        b) The substrate node is within the hop count limit.
        c) The available resource of the substrate node meets with the constraint (2).
        d) There is at least a path whose bandwidth meets with (3). And the constraint on each hidden hop on the path must meet with (4).
    4.3: Assign the candidate node with minimum mapping cost (see (11)) to $Enode$ .
5: Else if either of $Snode$ and $Enode$ is unmapped, the unmapped node will be mapped as follows:
    5.1 Find a subset of candidate substrate nodes for the unmapped node within the hop count limit of the
```

mapped node. And the candidate substrate node must be meet with the conditions as follows:

a) The available resource of the unallocated substrate node meets with (2).

b) There is at least a path whose bandwidth meets with (3). And the constraint on each hidden hop on the path must meet with (4).

5.2 Assign the candidate node with minimum mapping cost (see (11)) to the unmapped node.

6: Else if both of them are mapped, there is at least a path whose bandwidth must meet with (3), and the constraint on each hidden hop on the path must meet with (4).

7: If it is succeed, put the mapped virtual link into M^*, and remove it from $M\sim$. Or choose a virtual link in M^* randomly, remove this virtual link and the ones to relative to it from M^* to $M\sim$.

8: If $M\sim=\varnothing$, put out the mapping scheme, or go back to 2.

4. PERFORMANCE EVALUATION

In this section, we will study the efficiency of our proposal. To achieve this, we will first describe the simulation environment, and then present our simulation results. The experiments focus primarily on the performance comparison of LB-Greedy with the BACA in [15].

4.1 Simulation Setup

As stated in [15], we set the substrate network topology with 100 nodes. The average substrate network connectivity is fixed at 0.02. The CPU and bandwidth resources of the substrate nodes and links are real numbers uniformly distributed between 50 and 100. We model the arrival of virtual network requests by a Poisson Process with rate 0.05 ; and each one has an exponentially distributed lifetime with an average of 400 time units. Furthermore, we set the virtual network size according to a discrete uniform distribution, using the values given in [2, 10]. The average virtual network connectivity is fixed at 0.5. The CPU and bandwidth demand of virtual nodes and virtual links are real numbers uniformly distributed between 0 and 50.

As stated in [14], we set the hop count within [4,8]. The expenses of Hidden hops on the path that will be traversed by the virtual link see [16], we set a substrate network transporting packet size $PS=1500$ bytes, the substrate node CPU capability $\Gamma=2.66$ GHz, the substrate node bandwidth capacity $\Phi=1$ Gbps, the number of cycles used to process a packet in the node is $\omega=40000$ Cycle. According to these parameter settings, the CPU resource that must be assigned to an intermediate node on the path that will be traversed by the virtual link is 1.2 times the bandwidth. The ranges of the weight coefficient α and β in our objective function (13) are both [0.2-0.8], and they meet with the constraint $\alpha+\beta=1$.

4.2 Evaluation Results

Our evaluation results quantify the efficiency of LB-Greedy we proposed. Several performance metrics for evaluation purposes are used, including the virtual network request acceptance ratio defined by Equation (12), the load balancing degree defined by Equation (13), the average of node load defined by Equation (14), the average of link load defined by Equation (15). We summarize the key observations from our simulation as follow.

Fig. 1 shows a comparison of the acceptance ratio of virtual network requests obtained by the mapping strategies LB-Greedy and BACA. We can see first of all that both of the two algorithms produce high acceptance ratio which achieves to 100%. Our proposal is consistently higher than BACA while the number of virtual network requests grows. An important factor is that BACA produce larger residual resource fragmentation on the substrate network than LB- Greedy, which influence the acceptance ratio of incoming virtual network requests.

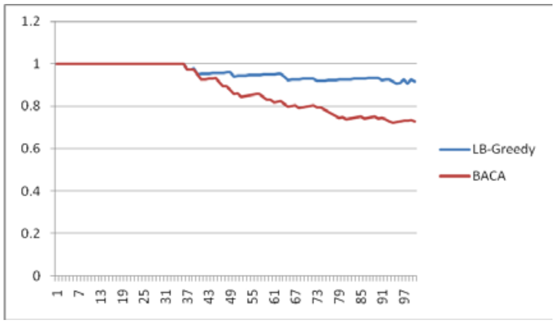

Fig. 1 The Virtual Network Request Acceptance Ratio Comparison.

Fig.2 evaluates the load balancing degree of LB-Greedy and BACA when the number of virtual network request increases. The less the value of load balancing degree is, the more balanced the substrate network is. So we can obviously find the LB-Greedy can produce a more balanced substrate network than BACA. The reason is that LB-Greedy scheme allocates the resource to the virtual network request with minimum mapping cost (see (11)), which constructs a more balanced substrate network.

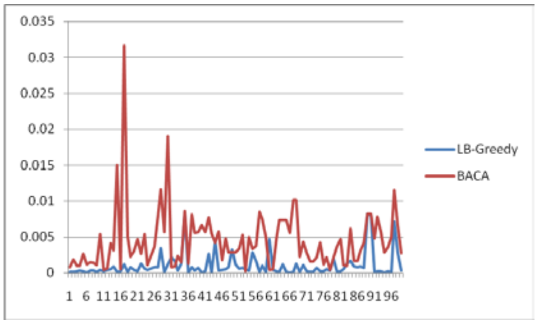

Fig. 2 The Loading Balance Degree Comparison.

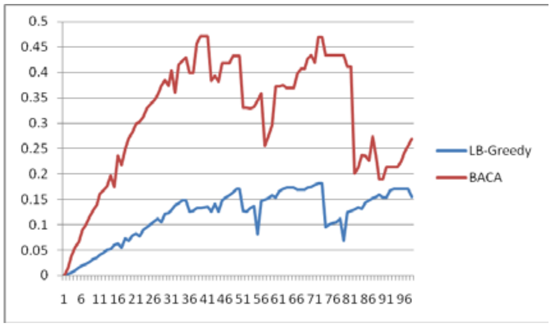

Fig. 3 The Average Of Node Load Comparison.

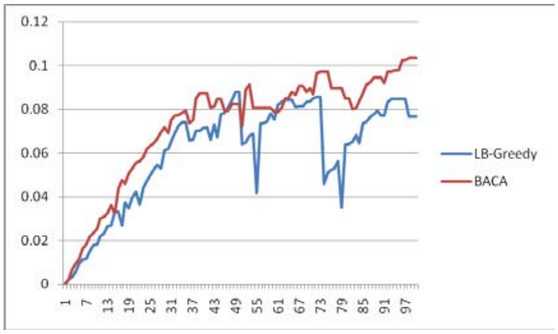

Fig. 4 The Average Of Link Load Comparison.

Fig. 3 and Fig. 4 show that LB-Greedy can produce less average of node load and link load than BACA for the same virtual network requests. To accept a virtual network request, LB-Greedy with hop count limit consumes less substrate resource than BACA. And also LB-Greedy constructs a more balanced substrate network with minimum mapping cost (see (11)) than BACA. All these efforts make LB-Greedy to produce less node load and link load than BACA in substrate network.

5. CONCLUSIONS

In this paper, we analyze the issues of existing researches, and consider the CPU resource that must be assigned to an intermediate node. To remove efficiently the resource bottlenecks due to the insufficient hidden hops' resource, this paper aims at the simultaneous loading balance of the substrate node and the substrate link, formulate the virtual networking mapping problem constrained by hops. Note that it is NP-hard and computationally intractable. In response, we proposed a LB-Greedy algorithm.

Based on extensive simulations, after getting rid of the resource expense of all intermediate nodes, it is able to remove efficiently the resource bottleneck with the goal of balanced link load and balanced node load. It will provide a more balanced substrate network for the consequential virtual network request, thereby improving the constructing success rate of virtual network, the availability of network resources and the profits of the infrastructure providers. Furthermore, limiting the hop count on the path that will be traversed by the virtual link will use as little substrate resource as possible. So, it can map as many virtual network requests as possible in the limited substrate networks and maximize the profit of infrastructure providers.

ACKNOWLEDGMENTS

This work is supported by the National Natural Science Fund China under Grant No. 2009CB320406, the National 863 High-tech Project of China under Grant No. 2011AA01A102, Funds for Creative Research Groups of China (60821001) and State Key Lab of Networking and Switching Technology.

REFERENCES

[1] A. Berl, A. Fischer, and H. de Meer, "Virtualisierung im future internet", Informatik-Spektrum, vol. 33, no. 2, pp. 186–194, 2010.

[2] J. Carapinha and J. Jiménez, "Network virtualization: a view from the bottom", in Proceedings of the 1st ACM workshop on Virtualized infrastructure systems and architectures, 2009, pp. 73–80.

[3] D. Schwerdel, D. Günther, R. Henjes, B. Reuther, and P. Müller, "German-lab experimental facility", Future Internet-FIS 2010, pp. 1–10, 2010.

[4] A. Lenk, M. Klems, J. Nimis, S. Tai, and T. Sandholm, "What's inside the Cloud? An architectural map of the Cloud landscape", in Proceedings of the 2009 ICSE Workshop on Software Engineering Challenges of Cloud Computing, 2009, pp. 23–31.

[5] N. M. M. K. Chowdhury, M. R. Rahman, and R. Boutaba, "Virtual network embedding with coordinated node and link mapping", in INFOCOM 2009, IEEE, 2009, pp. 783–791.

[6] M. Zhang, C. Wu, M. Jiang, and Q. Yang, "Mapping multicast service-oriented virtual networks with delay and delay variation constraints", in GLOBECOM 2010, 2010 IEEE Global Telecommunications Conference, 2010, pp. 1–5.

[7] S. ZHANG and X. QIU, "A novel virtual network mapping algorithm for cost minimizing", Cyber Journals: Journal of Selected Areas in Telecommunications (JSAT), 2011.

[8] H. Yu, C. Qiao, V. Anand, X. Liu, H. Di, and G. Sun, "Survivable Virtual Infrastructure Mapping in a Federated Computing and Networking System under Single Regional failures", in GLOBECOM 2010, 2010 IEEE Global Telecommunications Conference, 2010, pp. 1–6.

[9] G. Sun, H. Yu, L. Li, V. Anand, H. Di, and X. Gao, "Efficient algorithms for survivable virtual network embedding", in Asia Communications and Photonics Conference and Exhibition, 2010.

[10] I. Fajjari, N. Aitsaadi, G. Pujolle, and H. Zimmermann, "VNE-AC: Virtual network embedding algorithm based on ant colony metaheuristic", in Communications (ICC), 2011 IEEE International Conference on, 2011, pp. 1–6.

[11] Y. Zhu and M. Ammar, "Algorithms for assigning substrate network resources to virtual network components", in Proc. IEEE INFOCOM, 2006, vol. 2.

[12] 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.

[13] J. F. Botero, X. Hesselbach, A. Fischer, and H. De Meer, "Optimal mapping of virtual networks with hidden hops", Telecommunication Systems, pp. 1–10, 2011.

[14] JIANG Ming, WANG Bao-jin, WU Chun-ming,KONG Xiang-qing, MIN Xiao and ZHANG Min. Research on Network Virtualization and Virtual Network Mapping Algorithm[J]. Acta Electronica Sinica, 2011, 39(6): 1315-1320.

[15] Qi Ning, Wang Bao-jin, Wang Bin-qiang and Zhang Dong. Research on Balanced Construction Algorithm of Virtual Network[J]. Jounal of Electronics & Information Technology, 2011, 33(6): 1301-1306.

[16] A. Fischer, J. F. Botero Vega, M. Duelli, D. Schlosser, X. Hesselbach Serra, and H. De Meer, "ALEVIN-A framework to develop, compare, and analyze virtual network embedding algorithms", 2011.