414 *The Open Automation and Control Systems Journal*, 2014, *6*, 414-425

A Survey on Virtual Network Embedding in Cloud Computing Centers

Xiaohui Wei[1,2], Shoufeng Hu[1], Hongliang Li[1,2,*], Fan Yang[1] and Yue Jin[1]

[1]*College of Computer Science and Technology, Jilin University,* [2]*Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, China*

Abstract: Bridging virtualized environments with physical environments, virtual network plays an important role in Cloud Computing infrastructures. How to allocate physical resources for virtual nodes/links to construct virtual network is known as Virtual Network Embedding (VNE) problem. It is a crucial issue that draws wide attention. This paper surveys existing VNE methods and algorithms and how they work under different application scenarios. First, a survey of different VNE objective metrics is presented; then several VNE algorithms are presented and categorized; finally, the performance of different VNE solutions are compared and discussed.

Keywords: Cloud computing, scheduling, virtualization, virtual infrastructure, virtual network.

1. INTRODUCTION

Nowadays, more and more Cloud computing centers are facilitated with virtualization technologies. Virtualization technology has involved from OS virtualization (Virtual Machines, VM) to virtualization of distributed system components, such as virtual cluster and virtual network. Network virtualization [1-3] plays an important role in bridging the virtual and physical infrastructures. Therefore, how to allocate virtual resources onto physical recourses to construct the virtual network became a crucial issue that eventually decides the resource utilization, throughput and energy consumption of a Cloud Computing environment.

Giving a Virtual Network Request, the allocation of a VN is not an easy task. How to match multiple VNs to a physical network [4-10], while satisfying the resource constraints is known as *Virtual Network Embedding* (VNE) problem [7, 8, 10-16], which is proved to be NP-complete [7]. Even if the locations of the virtual nodes are predetermined, embedding virtual links is still NP-hard [7].

Therefore, over the past decade, researchers have worked on approximate and heuristic algorithms [2, 4-13, 17-28, 31-41] to solve the VNE problem for different objectives, such as load balance, energy efficiency, and high system throughput, etc.

This paper examines and presents taxonomy of the existing VNE algorithms, then discusses the performance and complexity of these algorithms to raise pending issues.

The remainder is organized as follows: Section 2 reviews common VNE concepts and preliminaries of VNE problem; Section 3 introduces existing VNE solutions; Section 4summaries VNE algorithms; Section 5 analyses the performance and discusses some existing works; Section 6 raises pending issues and summarizes the paper to make conclusions.

2. CONCEPTS AND PRELIMINARIES

2.1. Substrate or Virtual Network

Substrate Network is usually denoted as a weighted undirected graph in existing literatures [5-8].

$$\begin{aligned}&\boldsymbol{G}_S=(\boldsymbol{N}_S,\boldsymbol{E}_S,\boldsymbol{C}_S,\boldsymbol{BW}_S)\\&\mathrm{c}(n)\in\boldsymbol{C}_S,\qquad\forall n\in\boldsymbol{N}_S\\&\mathrm{bw}(e)\in\boldsymbol{BW}_S,\quad\forall e\in\boldsymbol{E}_S\end{aligned}\tag{1}$$

$\boldsymbol{N}_S$ denotes the set of substrate nodes and $\boldsymbol{E}_S$ is the set of substrate links. The notations $\boldsymbol{C}_S$ and $\boldsymbol{BW}_S$ represent the constraint attributes of substrate nodes and links. The functions $\mathrm{c}(n)$ and $\mathrm{bw}(e)$ return the attributes of a certain node and link. In Fig. (**1**), $\boldsymbol{N}_S=\{\mathrm{A,B,C,D,E,F,G}\}$, $\boldsymbol{E}_S=\{$(A,B), (A, C), (B,D), (C,D), (C,E), (C,F), (D,G), (E,F), (F,G)$\}$. The number is the constraints of.

$\boldsymbol{P}_S$ is defined as the set of all substrate loop-free paths. $\boldsymbol{P}(n_1, n_k)$ denotes a set of all the reachable paths from n_1 to n_k In Fig. (**1**), $\boldsymbol{P}(\mathrm{A,B})=\{\ p(\mathrm{A,B})\ ,\ p(\mathrm{A,C,D,B})\ ,\ p(\mathrm{A,C,E,F,G,D,B})\ ,\ p(\mathrm{A,C,F,G,D,B})\ \}$. $\mathrm{bw}(\mathrm{A,C,F,G,D,B})=5$.

Similar to the substrate network, the virtual network can also be denoted as a weighted undirected graph:

$$\boldsymbol{G}_V=(\boldsymbol{N}_V,\boldsymbol{E}_V,\boldsymbol{C}_V,\boldsymbol{BW}_V)\tag{2}$$

All virtual nodes and links are respectively associated with their capacity constraints. In Fig. (**1**), there are 2

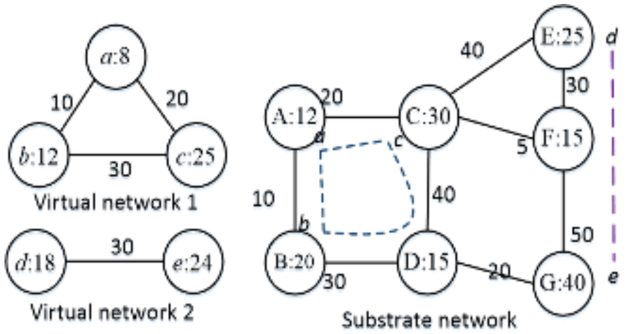


Fig. (1). Virtual Network Mapping.

VNs. VN 2 includes two virtual nodes(d , e), and one virtual link.

2.2. Virtual Network Embedding

The VNE problem can be described as a mapping.

$$\begin{aligned}&\mathbf{M}_{G_V}: \boldsymbol{G}_V=(\boldsymbol{N}_V,\boldsymbol{E}_V,\boldsymbol{C}_V,\boldsymbol{BW}_V)\rightarrow \boldsymbol{G}'_S=(\boldsymbol{N}'_S,\boldsymbol{P}'_S,\boldsymbol{C}'_S,\boldsymbol{BW}'_S)\\ &\qquad \boldsymbol{N}'_S\subseteq \boldsymbol{N}_S, \boldsymbol{P}'_S\subseteq \boldsymbol{P}_S\end{aligned} \tag{3}$$

The VNE problem can be divided into two parts: node mapping $\mathbf{M}_{N_V}$ and link mapping $\mathbf{M}_{E_V}$ M( ) returns a substrate node or path assigned to a virtual node or link.

Node mapping:

$$\begin{aligned}&\mathbf{M}_{N_V}: (\boldsymbol{N}_V,\boldsymbol{C}_V)\rightarrow(\boldsymbol{N}'_S,\boldsymbol{C}'_S), \forall n,n_\mathrm{i},n_\mathrm{j}\in \boldsymbol{N}_V\\ &\mathrm{M}(n)\in \boldsymbol{N}'_S, \mathrm{c}(n)\in \boldsymbol{C}'_S,\\ &\mathrm{M}(n_\mathrm{i})=\mathrm{M}(n_\mathrm{j}) \text{ and } \mathrm{c}(n_\mathrm{i})=\mathrm{c}(n_\mathrm{j}), \text{ iff } n_\mathrm{i}=n_\mathrm{j}.\end{aligned} \tag{4}$$

Link mapping:

$$\begin{aligned}&\mathbf{M}_{E_V}: (\boldsymbol{E}_V,\boldsymbol{BW}'_V)\rightarrow(\boldsymbol{P}'_S,\boldsymbol{BW}'_S), \forall e,e_\mathrm{i},e_\mathrm{j}\in \boldsymbol{E}_V\\ &\mathrm{M}(e)\in \boldsymbol{P}'_S, \mathrm{bw}(\mathrm{M}(e))\in \boldsymbol{BW}'_S,\\ &\mathrm{M}(e_\mathrm{i})=\mathrm{M}(e_\mathrm{j}) \text{ and } \mathrm{bw}(\mathrm{M}(e_\mathrm{i}))=\mathrm{bw}(\mathrm{M}(e_\mathrm{j})), \text{ iff } e_\mathrm{i}=e_\mathrm{j}.\end{aligned} \tag{5}$$

In Fig. (1)., node mapping includes $a\rightarrow \mathrm{A}$, $b\rightarrow \mathrm{B}$, $c\rightarrow \mathrm{C}$, $d\rightarrow \mathrm{E}$ and $e\rightarrow \mathrm{G}$. Link mapping includes $(a,b)\rightarrow(\mathrm{A,B})$, $(a,c)\rightarrow(\mathrm{A,C})$, $(b,c)\rightarrow(\mathrm{B,D,C})$ and $(d,e)\rightarrow(\mathrm{E,G})$.

2.3. Available Resources of Substrate Network

In order to accept the subsequent VNR, the resource availability of substrate network needs to be periodically checked. $\boldsymbol{G}_{res}$ denotes the available substrate resources for a substrate network. The available resources can be computed by the substrate resources minus the virtual resources.

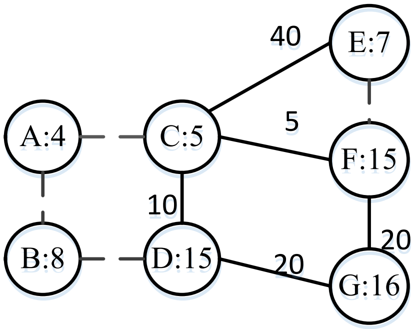


Fig. (2). Residual Network Resource of Fig. (**1**).

$$\boldsymbol{G}_{res}=(\boldsymbol{N}_{res},\boldsymbol{E}_{res},\boldsymbol{C}_{res},\boldsymbol{BW}_{res}), \tag{6}$$

After the mapping in Fig. (**1**), the surplus resources are depicted by Fig. (**2**). The dashed line means that the bandwidth is 0.

3. VNE SOLUTIONS

Since the VNE problem is proved to be NP-hard, over the past decade, most existing works are designed for certain scenario and objective. In this section, we introduce some typical solutions and evaluation metrics for VNE models.

3.1. Importance of Substrate or Virtual Node

Some heuristics [13, 28] have proposed a concept of importance of the substrate node. The concept can utilize physical network's topology to accelerate the mapping.

Sometimes, researchers use the residual capability $\mathrm{RC}(n)$ to evaluate the importance of a node. In other cases, it can be given by combining residual resource of its adjacent nodes or all nodes.

$$\mathrm{RC}(n)=\mathrm{c}_{res}(n)\sum_{e\in AdjE(n)}\mathrm{bw}_{res}(e) \tag{7}$$

$$\mathrm{Imp}(n)=\mathrm{RC}(n)+\sum_{\substack{n'\in AN(n)\\ n'\neq n}}\frac{\mathrm{RC}(n')}{\mathrm{dis}(n',n)+1} \tag{8}$$

$AdjE(n)$ denotes the adjacent links and nodes of n. $AN(n)$ represents all nodes or its adjacent nodes. $\mathrm{dis}(n',n)$ is defined as the distance between n' and n.

Similar to the above, the importance of a virtual node is defined by its capability and the link's importance is decided by its bandwidth.

3.2. Load Balance

Load balancing [32] aims to reduce the execution time of parallel jobs and improve system efficiency and QoS [24, 42, 43]. Many existing methods balance the workload.

(1) The *stress ratio* of the maximum and average number of virtual nodes/links embedded to substrate node/link around the substrate network [6] is used as a metric to evaluate the load balancing of a system.

$$\mathrm{Ratio}_S(\mathrm{t}) = \max_{n \in \boldsymbol{N}_S}\{\mathrm{Stress}(n,\mathrm{t})\} / \left(\sum_{n \in \boldsymbol{N}_S} \mathrm{Stress}(n,\mathrm{t}) / |\boldsymbol{N}_S| \right) \tag{9}$$

$$\mathrm{Ratio}_E(\mathrm{t}) = \max_{e \in \boldsymbol{E}_V}\{\mathrm{Stress}(e,\mathrm{t})\} / \left(\sum_{e \in \boldsymbol{E}_S} \mathrm{Stress}(e,\mathrm{t}) / |\boldsymbol{E}_S| \right) \tag{10}$$

It is obvious that $\mathrm{Ratio}_N(\mathrm{t}) \geq 1$ and $\mathrm{Ratio}_E(\mathrm{t}) \geq 1$. If equal to 1, the system is well-balanced. A smaller stress ratio stands for better load balance.

To improve the resource utilization and save energy, we can balance the workload in that servers/links (denoted by the notation $\boldsymbol{N}_H$ or $\boldsymbol{E}_H$) hosting virtual nodes/links. It just needs to make two small changes to equations (9) and (10), through replacing $\boldsymbol{N}_S$ and $\boldsymbol{E}_S$ with $\boldsymbol{N}_H$ and $\boldsymbol{E}_H$.

$$\boldsymbol{N}_H = \{n | n \in \boldsymbol{N}_S \wedge \mathrm{c}_{res}(n) < \mathrm{c}(n)\} \tag{11}$$

$$\boldsymbol{E}_H = \{e | e \in \boldsymbol{E}_S \wedge \mathrm{bw}_{res}(e) < \mathrm{bw}(e)\} \tag{12}$$

(2) Another way to balance load around the system is to aim at maximum link/node stress. Two definitions are proposed in [6] to balance computing and traffic loads, which are the neighborhood resource availability (NR) of a substrate node and the distance referring to the shortest-distance path algorithm. The maximum stress of a link and node is defined as $\mathrm{S}_{\mathrm{lmax}}(\mathrm{t}) = \max\{\mathrm{Stress}(e,\mathrm{t})\}$ and $\mathrm{S}_{\mathrm{nmax}}(\mathrm{t}) = \max\{\mathrm{Stress}(n,\mathrm{t})\}$. The distance of a substrate path p and NR is defined as:

$$\begin{aligned} \mathrm{NR}(v,\mathrm{t}) = & \left[\mathrm{S}_{\mathrm{nmax}}(\mathrm{t}) - \mathrm{Stress}(v,\mathrm{t})\right] \bullet \\ & \sum_{e \in AdjE(v)} \left[\mathrm{S}_{\mathrm{lmax}}(\mathrm{t}) - \mathrm{Stress}(e,\mathrm{t})\right] \end{aligned} \tag{13}$$

$$\mathrm{Distance}(p,\mathrm{t}) = \sum_{e \in e} \frac{1}{\mathrm{S}_{\mathrm{lmax}}(\mathrm{t}) + \delta_L - \mathrm{Stress}(e,\mathrm{t})} \tag{14}$$

A smaller distance value usually represents a lower traffic load. A higher NR represents lower load and connects with links of lower loads.

(3) A novel concept of "*skewness"* as an evaluation metric to quantify the unevenness in the utilization of multiple resources is proposed by [44]. n and r_i represent the number of all resources and the utilization of the i^{th} resource. Skewness of a server p is computed by

$$skewness(p) = \sqrt{\sum_{i=1}^{n} \left(\frac{r_i}{\bar{r}} - 1 \right)^2} \tag{15}$$

$\bar{r}$ is the average utilization of all resources in server p. It only needs to consider the bottleneck resources. By minimizing the value, it can balance workloads nicely and improve the overall utilization of server resources.

3.3. Virtual Node Migration

Migration [27, 44-47] is essential to the dynamic resource allocation and scheduling problems. Many previous VNE solutions are based on migration to allocate resources. VNE has to decide the migration time, migration source and destination of a virtual node.

Migration Time

In the paper [44], it defines a server as a *hot spot*. If any resource utilization is above a *hot threshold*, which indicates the server being overloaded, some VMs on it should be migrated away. The *temperature* of a hot spot p is defined as the variance of all resource utilization beyond the hot threshold. Only overloaded resources are considered. The temperature reflects its degree of overload. If a server is not a hot spot, its temperature is 0.

It also defines a server as a *cold spot* [44]. If all resource utilizations are below a *cold threshold*, which indicates the server being idle, it's a potential candidate to turn off to save energy.

Migration Source and Destination

For each server p, firstly, it decides whether it's the hot spots, and which VMs should be migrated. Secondly, it sorts hot spots according to temperature. Migrating such VM can reduce temperature and *skewness* the most. Finally, a destination server will be selected, that can reduce the *skewness* of the system the most [44].

On the contrary, cold spot is also handled [44]. The challenge is to avoid sacrificing performance and load oscillation, while reducing the number of active servers due to low load.

3.4. Energy Consumption

The paper [18] purposes a power consumption model. All active substrate nodes consist of working nodes and intermediate nodes which are responsible for forwarding packets. Power consumptions for accommodating a new VN request include three parts.

Node Power Consumption

The node power consumption for accommodating a new VN request, denoted by P_N, is proportional to N_W, which denotes the number of working nodes needed to be powered *on* from *off* state.

$$P_N = N_W P_b + P_l \sum_{u \in \boldsymbol{N}_V} \mathrm{c}(u) \tag{16}$$

P_b is the server's baseline power, and P_l represents the proportion factor for virtual node u.

Link Power Consumption

The link consumption denoted by P_L, is proportional to both N_W and N_I which is the number of the intermediate nodes needed to be powered from *off* state to *on*.

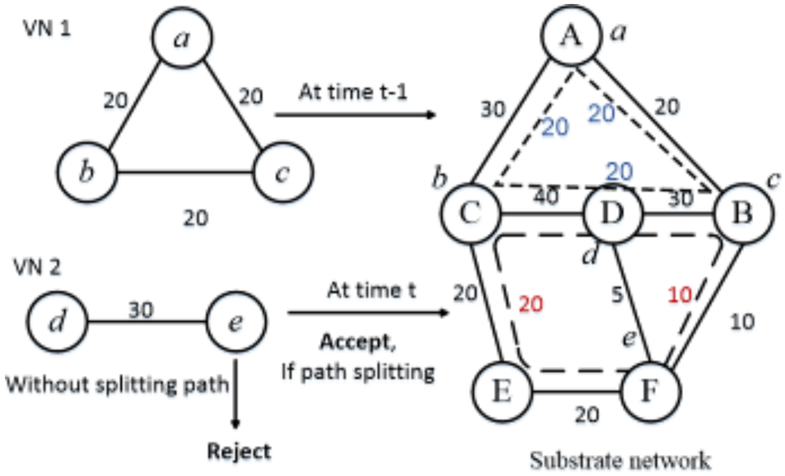


Fig. (3). Illustration of path splitting.

$$P_L = N_W P_n + N_i(P_b + P_n) \tag{17}$$

where P_n denotes the power consumption of dedicated offload engine [18].

Overall Power Consumption

The overall power consumption for accommodating a VN request, denoted by P, is proportional to both N_W and N_i. And the overall energy consumption is denoted as E.

$$P = P_N + P_L = P_l \sum_{u \in N_V} \mathrm{c}(u) + (N_w + N_i)(P_b + P_n) \tag{18}$$

$$E = P \cdot T_d = (P_l \sum_{u \in N_V} \mathrm{c}(u) + (N_w + N_i)(P_b + P_n)) \cdot T_d \tag{19}$$

3.5. Load Fluctuation

To deal with the peak workload and traffic fluctuation, all of the prior proposals reserve the maximum of fixed resources, resulting in resources waste and lower the resource utilization. However, some papers [22, 24] propose Opportunistic Resource Sharing (ORS).

ORS reflects the time-varying resource requirement and load fluctuation, which models load fluctuation as the combination of a *basic* and a *variable sub-load* which occurs with a probability. For the basic sub-load, it has no choice but to allocate the equal required slots. For the variable sub-load, multiple virtual links can opportunistically share one slot.

3.6. Path Splitting and Migration

To harness some small pieces of bandwidth, the paper [7] allows the substrate network to split a virtual link over multiple substrate paths, which is called *path splitting*(PS) [5]. PS has some benefits: load balance, reliability, etc.

Fig. (**3**) is one example. Firstly, map the VN 1 to the substrate network at time t-1. Secondly, embed VN 2 at time t. If PS is not allowed, VN 2 would be rejected. However, if allowed, VN 2 would successfully be accepted. PS generates 2 mappings: (d, e) → (D, C, E, F) with 20 units of bandwidth and (d, e) → (D, B, F) with 10 units.

In order to reduce the bandwidth fragments, *path migration* [7] can periodically re-optimize the allocation. In Fig. (**3**), after VN 1 departs and those assigned resources are released, migrate the two paths (D, C, E, F) and (D, B, F) to the same path (D, B) to reduce some path fragments. In practice, migrating path may produce overhead. Tradeoff between benefits and overheads is essential.

3.7. Evaluation Models

3.7.1. Revenue and Cost

A common objective is maximizing the revenue [7], while minimizing the cost. The cost of successfully accepting a VN is defined as the sum of substrate resources assigned to the VN, and the revenue can be gotten by the virtual resources.

$$\text{Revenue}(\boldsymbol{G}_V, \mathrm{t}) = \alpha \sum_{n \in \boldsymbol{N}_V} \mathrm{c}(n) + \beta \sum_{e \in \boldsymbol{E}_V} \mathrm{bw}(e)\ . \tag{20}$$

$$\text{Cost}(\boldsymbol{G}_V, \mathrm{t}) = \alpha \sum_{n \in \boldsymbol{N}_V} \mathrm{c}(n) + \beta \sum_{e^v \in \boldsymbol{E}_V} \sum_{e^s \in \boldsymbol{E}_S} \mathrm{bw}(e^v, e^s)\ . \tag{21}$$

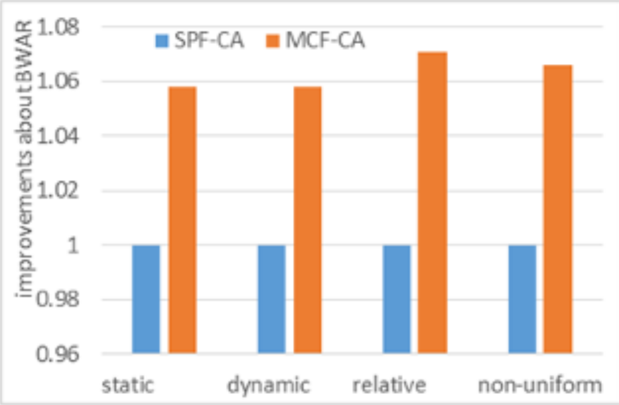
$\mathrm{bw}(e^v, e^s)$ is the bandwidth assigned to e^v from e^s. α and β are tunable weights.

The ratio of revenue and cost is also a common evaluation metrics.

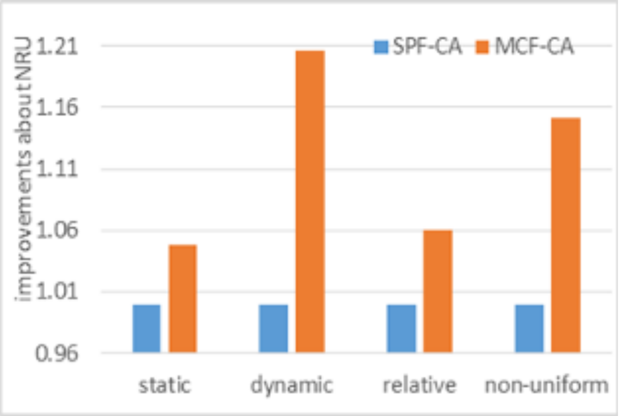
3.7.2 Resource Utilization

Higher substrate resource utilization has always been an objective. Some works focus on accepting more VNRs, while others focus on re-arranging the virtual network.

Resource utilization includes three types: *node utilization* [24], *link utilization* and *entire network resource utilization*. $\boldsymbol{N}_S^{\text{mapped}}$ and $\boldsymbol{E}_S^{\text{mapped}}$ include those nodes and links hosting virtual nodes and links.



(**a**) bandwidth acceptance ratio



(**b**) network resource utilization

Fig. (4). Improvements of MCF-CA relative to SPF-CA.

$$\text{NodeUtilization} = \sum_{v\in N_S^{\text{mapped}}} \mathrm{c}(v) \Big/ \sum_{n\in N_S} \mathrm{c}(n) \qquad (22)$$

$$\text{LinkUtilization} = \sum_{e'\in E_S^{\text{mapped}}} \mathrm{bw}(e') \Big/ \sum_{e\in E_S} \mathrm{bw}(e) \qquad (23)$$

$$\text{NetUtilization} = \left(\sum_{v\in N_S^{\text{mapped}}} \mathrm{c}(v) + \sum_{e'\in E_S^{\text{mapped}}} \mathrm{bw}(e') \right) \Big/ \left(\sum_{n\in N_S} \mathrm{c}(n) + \sum_{e\in E_S} \mathrm{bw}(e) \right) \qquad (24)$$

3.7.3. Acceptance Ratio of VNs

The ratio of VNs [35] is one of the most popular metrics to evaluate a VNE algorithm. It is defined as a ratio of VNs successfully accepted to all VNs. The variable $x_i \in \{0,1\}$. If the i$^{\text{th}}$ VNR is accepted successfully, $x_i = 1$; otherwise, $x_i = 0$.

$$\text{AptRatio} = \left[\sum_{i\in \boldsymbol{VNR}^{accept}} x_i \right] / \left| \boldsymbol{VNR} \right| \bullet 100\% \qquad (25)$$

There are also some uncommon metrics, such as, the number of active physical nodes [16], delay [41], total allocated bandwidth [37], throughput [41, 48-62], etc.

4. SIMPLY SUMMARIES OF VNE MODELS

This section simply summaries and compares some existing models. Table **1** shows the comparison of the objective, category, evaluation metric, etc.. Table **2** lists the variants of VNE algorithms. (**Note** that ">" represents the former outperforms or equal the latter about the relevant metrics).

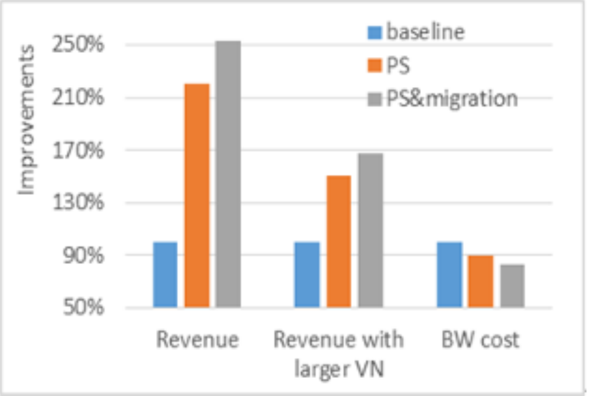


Fig. (5). Improvements of PS&M relative to baseline

5. ANALYSES AND COMPARISONS

This section presents some performance comparisons of VNE models from three different perspectives: virtual link mapping, virtual node mapping and entire VN mapping. The data used in this paper is based on the data and statements from the original literatures [4, 7, 8, 13, 17, 18, 21, 22, 24, 28]. Some data are enlarged and reduced in proportion, so it is easier to make comparisons.

5.1. Virtual Link Mapping

This section makes comparisons among several virtual link mapping algorithms, which include MCF, path splitting and migration, shortest-path (SP), BFS and ORS, etc.

(1) *MCF outperforms SP*. Fig. (**4**). shows the comparisons of bandwidth acceptance ratio (BWAR) and network resource utilization (NRU) between MCF and SP. It proves that MCF-CA outperforms SPF-CA about both BWAR and NRU [4].

(2) In Fig. (**5**), PS can increase the revenue and save the bandwidth cost in contrast to baseline using SP. migration further strengthens the improvements. Because PS can improve the VN acceptance ratio and migration can reduce bandwidth fragments [7].

(3) ORS *is better than R-ViNE, while R-ViNE is better than Greedy using Greedy node mapping and path splitting*, proved by Fig. (**6**). Because ORS can map multiple virtual elements into one substrate element [24].

(4) *BFS algorithm has better performance than MCF and SP*. In Fig. (**7**), Based-BFS produces the highest long-term R/C ratio, since based-BFS avoids mapping virtual links onto a long substrate path [13].

(5) In Fig. (**8**), NCM is better than G-SP, but worse than G-MCF about link utilization [24].

(6) *ORSTA outperforms TA, while ORS located between TA and Greedy*. In Fig. (**9**), we observe similar results that ORSTA outperforms better than TA, ORS, and Greedy [22].

Table 1. Comparisons among algorithms.

Algorithm	Purpose and Contributions	Technology	Performance Metric	Performance Comparison	Complexity	Refer
MCF	Maximizing the number of VNs	MCF	BWAK, NRU	MCF > SP-CA > LCP-CA		[4]
VNA-I: VNA-II	Increasing efficiency, On-demand, reconfiguration	1-stage	Max node stress; For Max link stress: Adaptive > subVN > Basic > Least−load Link−opt > Adaptive > Node−opt			[6]
G-SP	This is a 2-stage algorithm, which is used as comparison.				$O(\lvert E_S\rvert+\lvert N_S\rvert\ log\lvert N_S\rvert+k)$	[7]
PS&M-VNE	Support Splitting and migration of Path (PS)	2-stage, G-SP, MCF	Average Revenue Bandwidth Cost	PS&M > PS > BL		[7]
vnmFlib	Reduce the mapping time	1-stage, path splitting	R/C−Ratio	vnmFlib > 2-stage	The worst: $\Theta(\lvert N_S\rvert!\lvert N_V\rvert)$	[8]
FELL	Balance load and consider response time	Simulated annealing, PS, 1-stage	Node Utilization	FELL > R-ViNE		[11]
HA-I HA-II	allocate bandwidth on traffic fluctuations	Bin packing, 2-stage	Number of time slots: HA-I > HA-II > no OBS			[12]
RW-MM	Increase revenues and acceptance ratio	2-stage, MCF, SP	Acceptance ratio, revenue and R/C Ratio. RW-based > BL-based/CB-based		polynomial-time	[13]
RW-BFS		-stage, BFS, MCF			exponential time	
MBPA	mprove efficiency of VC and QoS	Mixed bin packing 2-stage	slowdown degree, remaining resource	MBPA ≅ Greedy > GR		[17]
EA-VNE	Reduce energy consumption	ILP, 2-stage	Average energy, revenue, running time	EA-VNE > D-ViNE-SP	in polynomial-time	[18]
HGA	Maximize utilization reduce energy cost.	Hybrid Genetic Algorithm	PM, Variance and Migration ratio	Itself with some parameters		[20]
ViNEYard D-ViNE	Leverage better coordination between node and link mapping.	MIP, LP, MCF, Coordinated	For acceptance ratio and revenue node/link utilization: ViNE-LB > ViNE-SP > G-MCF >G-SP		in polynomial time	[21]
R-ViNE						
ORSTA	Consider the workload fluctuation	bin packing, Markov chain	Acceptance Ratio Node/Link Utilization Ratio	ORSTA > TA > ORS > Greedy		[22]
ORS	Consider time-varying resource requirements	Greedy, ILP, bin packing, first-fit,	acceptance ratio utilization ratio	ORS > R-ViNE > Greedy-PS	$O(\lvert N_S\rvert^4+F\lvert N_S\rvert^2)$	[24]
NCM	Support QoS	MIP, 2-stage, ViNEYard,	Revenue, acceptance ratio: G-MCF > NCM > G-SP. Cost: G-MCF > NCM and G-SP		in polynomial time	[24]
DVMA	Consider demands' fluctuations and dependable allocation	MIP, bin packing		itself	time partitioning $O(N\lvert T\rvert^2\lvert V\rvert)$	[27]
TOP-VCM	parallelism of applications	1-stage, GID	Processing Time, R/C, Revenue	TOP-VCM_CB/sumTR > A_vnmFlib	$\Theta(\lvert N_S\rvert!\lvert N_V\rvert)$	[28]

Table 2. Illustrations of notions.

Notation	Description
MCF	MCF is a Multi-Commodity Flow [29, 51, 52] based approach to VN resource allocation algorithm [4]
SPF-CA/ LCP-CA	Capacity allocation with Shortest/ Least-cost path algorithm [4]
Least-Load	Select the least loaded substrate nodes as the virtual nodes and uses the shortest distance path algorithm. [6]
VNA-I/VNA-I	VNA-I/VNA-II is a VN Assignment without/with reconfiguration. [6]
Basic/ subVN	The basic VN assignment algorithm or Basic with subdividing the VN topology [6].
Adaptive	The subVN based scheme with the adaptive optimization strategy. [6]
G-SP	An optimal embedding computationally intractable algorithm. [7]
BL/ BL-SP	Baseline VNE Algorithm including Greedy node and link mapping. [7]
PS&M	PSM-VNE is a novel VNE method support for Path Splitting and Migration. [7]
VnmFlib	VnmFlib is a VNE backtracking algorithm based on sub-graph isomorphism detection (GID) [8]
FELL	FELL is a Flexible virtual network embedding algorithm with guaranteed Load Balancing. [11]
HA-I / HA-II	Two versions of opportunistic bandwidth sharing(OBS) for VNE. [12]
RW-based	Include RW-MM-SP using shortest path, RW-MM-MCF using MCF and RW-BFS. [13]
CB-based	Using equation (7) to compute node rank, include CB-MM-SP, CB-MM-MCF and CB-BFS. [13]
BL-based	Include BL-SP, which is BL, and BL-MCF, which is BL using MCF to embed virtual links. [13]
RW-MM / RW-BFS	Two VNE algorithms through the Markov Random Walk with Max Match and Breadth-First Search. [13]
GR / ILP	Gradually Relaxed Algorithm [17] OR Integer linear programming [18]
MBPA	Mixed Bin Packing algorithm to implement a novel elastic resources allocation strategy. [17]
DVMA	Dependable virtual machine allocation considering fault-tolerance and maintenance. [27]
HGA	Hybrid Genetic Algorithm combined with knack problem and multiple fitnesses. [20]
ViNEYard	ViNEYard uses mixed integer programming, which includes D-ViNE, R-ViNE and their extensions. [21]
ViNE-LB/ ViNE-SP	D/R-ViNE with load balancing/ shortest path link mapping. [21]
G-MCF	Greedy node mapping with MCF link mapping. [7]
TA	A partial framework, including topology-awareness, and greedy node/link mapping. [22]
Greedy	The traditional greedy embedding algorithm, including greedy node/link mapping. [22]
ORSTA	A VNE framework based on Opportunistic Resource Sharing and Topology-Aware node ranking. [22]
ORS	An Opportunistic Resource Sharing-based mapping framework. [24]
ARMS	An automated resource management system uses virtualization technology to allocate resources. [44]
A_vnmFlib	VnmFlib algorithm using Equation (7) & full mapping. [28]
TOP-VCM	A Topology-aware Partial Virtual Cluster Mapping algorithm based on graph isomorphism detection. [28]
based_CB/sumTR	TOP-VCM algorithm using equation (7) & full mapping / equation (8) & partial mapping. [28]

Generally speaking, a higher acceptance ratio, a higher utilization with a smaller cost can improve the revenue and R/C ratio. Concluding the above 6 points gets one conclusion about the virtual link mapping: ORSTA > TA > ORS > R-ViNE > PS&M > PS > MCF> NCM > SP.

5.2. Virtual Node Mapping

This section makes comparisons of node utilization among some algorithms.

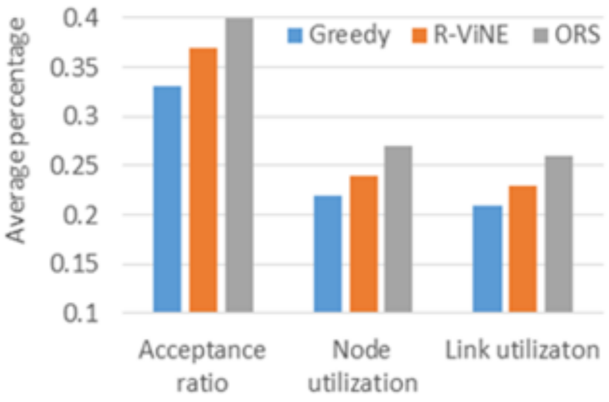
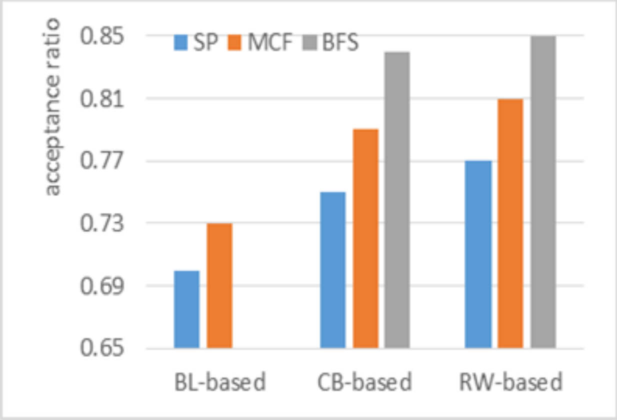
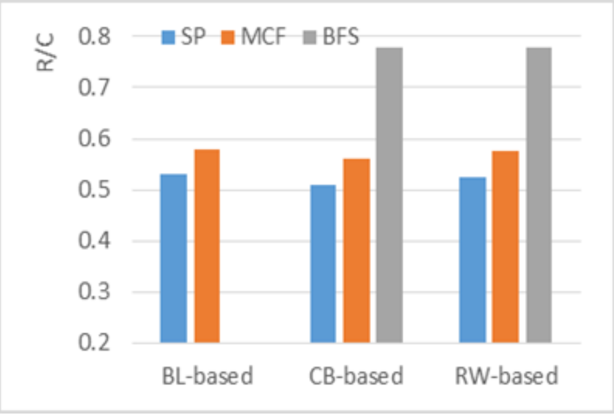


Fig. (6). Comparisons among Greedy, R-ViNE and ORS.



(a) average acceptance ratio



(b) average R/C

Fig. (7). Comparisons among SP, MCF and BFS.

(1) In Fig. (**5**), *ORS has better node utilization than R-ViNE and Greedy, and Greedy is worse than R-ViNE* [24], because ORS algorithm allows the multiple virtual nodes to share one substrate node.

(2) *NCM's performance is positioned in between G-SP and G-MCF* [24]. In Fig. (**8**), the CPU utilization and memory utilization have similar shapes that NCM is better than G-SP, but worse than G-MCF.

(3) *ORSTA outperforms TA, while ORS located between TA and Greedy*. In Fig. (**9**), we observe similar results where ORSTA outperforms better than TA, ORS, and Greedy [22].

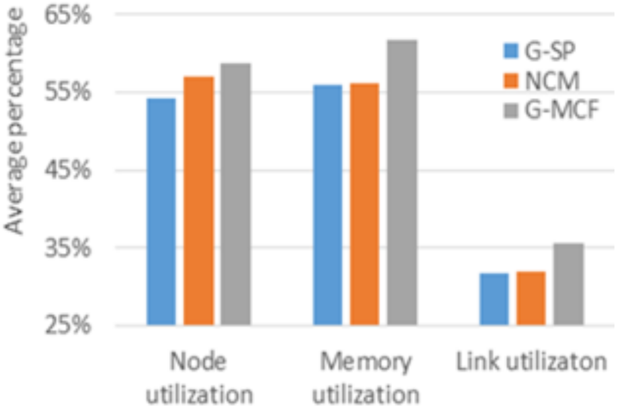


Fig. (8). Comparisons among G-SP, NCM and G-MCF.

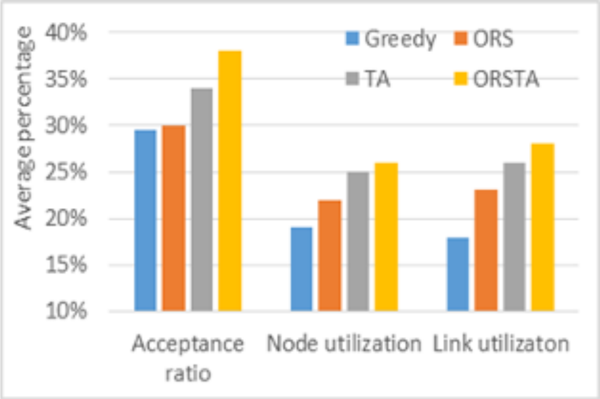


Fig. (9). Comparisons among Greedy, ORS, TA, ORSTA.

(4) *RW-based outperforms CB-based and BL-based (using Greedy), and Greedy produces the worst performance*. In Fig. (**10**), RW-based gains the best performance, and CB-based has the second. Because the resource ranking is not only determined by the node itself, but also influenced by its neighbors with NodeRank [13].

(5) *The node utilization of sum_TR is more than CB, while CB being more than A_vnmFlib*. In both random and light-first VC request scenarios [28], Fig. (**11**). shows that TOP-VCM-CB using equation (8) is positioned between A_vnmFlib and TOP-VCM-sumTR using equation (10), despite the three are similar in the heavy–first VC request scenarios with allowing deviation.

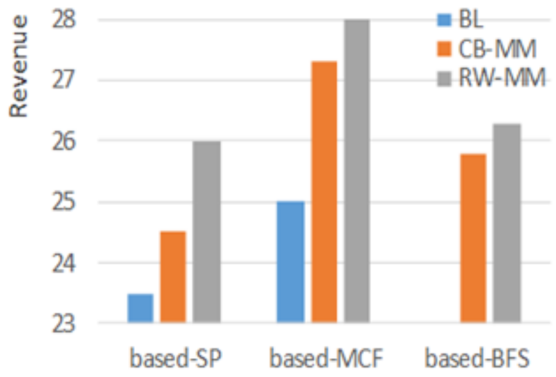
(6) Fig. (**12**) shows: *ViNE-LB > ViNE > ViNE-SP > G-MCF > G-SP* [21].

Generally, combining the above 6 points on node utilization gets the conclusions: ORSTA > TA > ORS > R-ViNE > Greedy-path splitting, RW-based/sum_TR > CB-based > A_vnmFlib, and ViNE-LB > ViNE > ViNE-SP > G-MCF > NCM > G-SP.

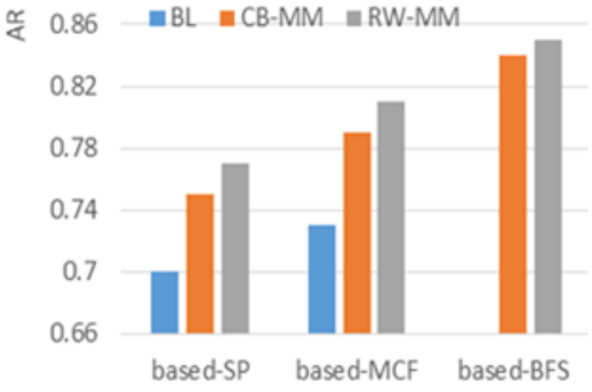
5.3. VN Mapping

(1) ORS is better than R-ViNE in acceptance ratio, while R-ViNE is better than Greedy using Greedy node mapping and path splitting [24], proved by Fig. (**7**).

(2) *NCM's resource utilization is positioned in between G-SP and G-MCF*. In Fig. (**8**), the CPU, memory and link



(a) average revenue



(b) average acceptance ratio

Fig. (10). Comparisons among SP, MCF and BFS

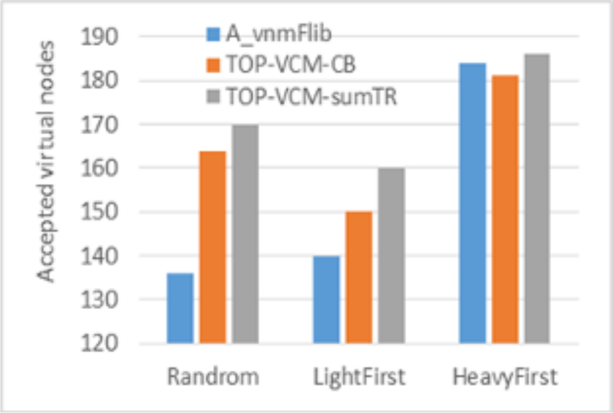


Fig. (11). Comparisons of accepted virtual nodes.

utilization have similar shape that NCM is better than G-SP, but worse than G-MCF [24].

(3) *ORSTA outperforms TA, while ORS located between TA and Greedy in the aspect of acceptance ratio*. In Fig. (**9**), opportunistic resource sharing and topology-aware node ranking indeed improve the deployment of virtual networks and further enable the substrate network to accept more VN requests [22].

(4) *VnmFlib outperforms 2-stage algorithms in both revenue and R/C, and consume the less time than 2-stage* [8], as demonstrated by Fig. (**13**).

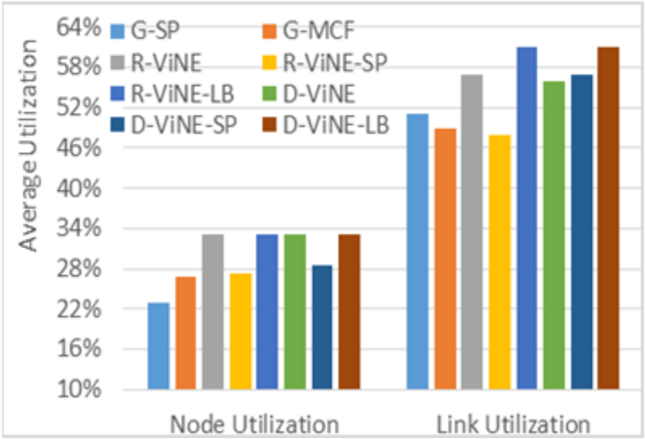
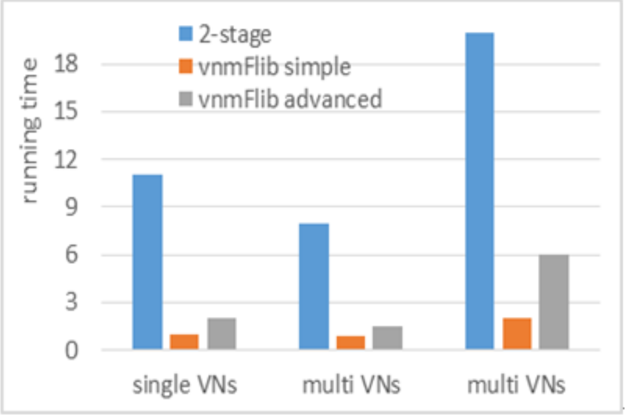
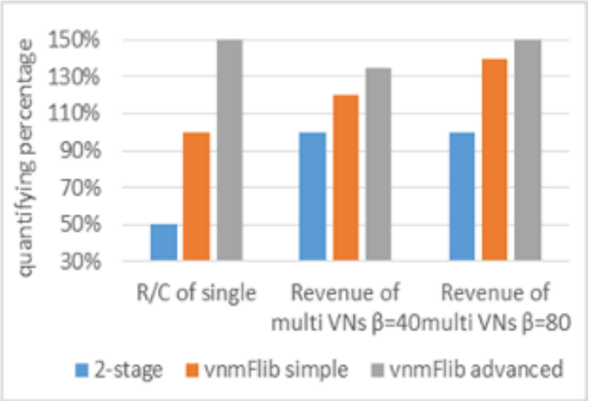


Fig. (12). Comparisons of resource utilization.



(a) runtime

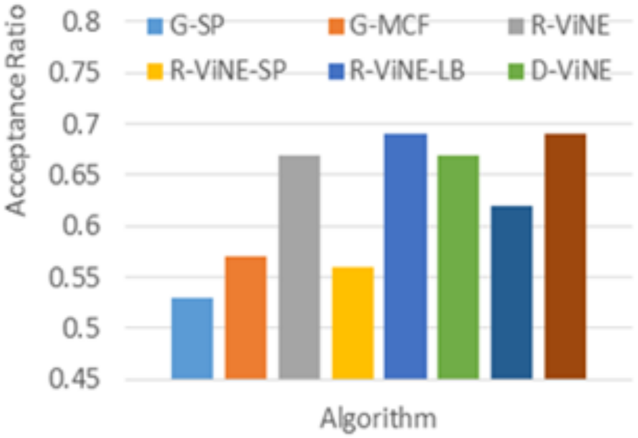


(b) revenue and R/C

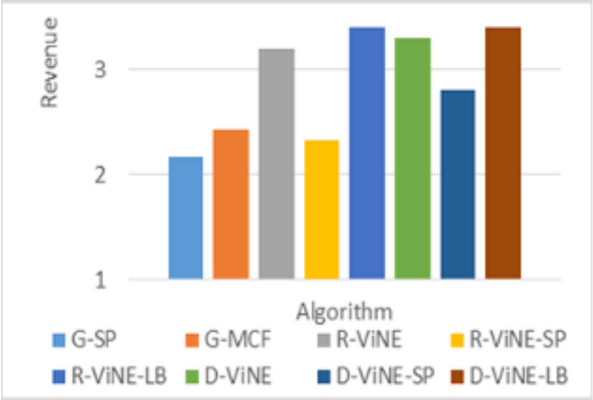
Fig. (13). Comparisons between 2-stage and vnmFlib.

(5) ViNE-LB has the best performance and ViNE has the second best performance, while G-MCF located between G-SP and ViNE-SP, and G-SP being the worst. In the paper [21], coordinated mapping of node and link leads to higher acceptance ratio Fig. (**14a**) and larger revenue Fig. (**14b**), and load balancing further increases the acceptance ratio and the revenue.

(6) TOP-VCM [28] produces better total revenue and R/C, while saving the average processing time. As demon-



(a) average acceptance ratio



(b) average revenue

Fig. (14). Comparison of (4, 8) arrival rate

strated Fig. (**15**), TOP-VCM_CB and TOP-VCM_sumTR achieve higher total revenue than A_vnmFlib in both Random and LightFirst VC request scenarios and improve R/C ratio. TOP-VCM-sumTR produces better results than TOP-VCM-CB. Because the latter only considers the own residual resource using equation (8), while the former uses both itself and adjust links and nodes using equation (9).

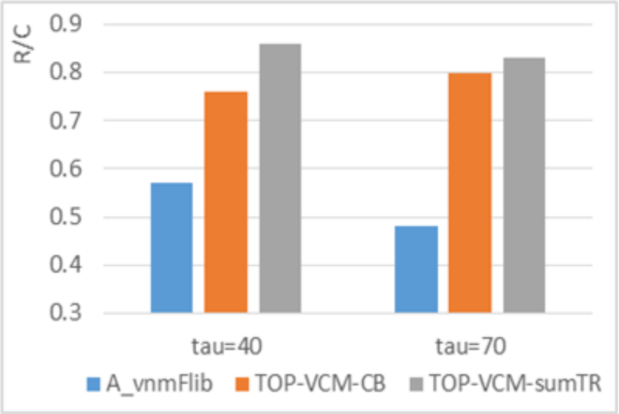
Generally, the network resource utilization, VN acceptance ratio, revenue and R/C have a positive correlation. Therefore, we can find some conclusions: ORSTA > TA > ORS > R-ViNE > Greedy-path splitting > G-MCF > NCM > G-SP, RW-based and TOP-VCM > A_vnmFlib > MBP > G-SP > GR, and ViNE-LB > ViNE > ViNE-SP > G-MCF > NCM > G-SP.

6. CONCLUSION AND PENDING ISSUES

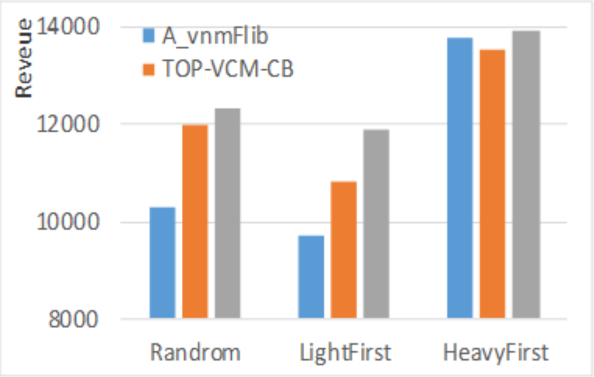
6.1. Pending Issues

Although the existing models have already dealt with many thorny problems, there are also some issues needing to be solved.

Firstly, with the development of wireless network [49, 50], wireless virtualization may become another topic, so it needs many totally different solutions to guarantee the load balance, QoS, etc.



(a) R/C about single VC test



(b) revenue about multi VC test

Fig. (15). Comparisons between TOP-VCM and A_vnmFlib

Secondly, VNE will contribute to connecting virtual computing clusters. Interconnecting distributed cloud sites is a promising way to provision on-demand large-scale virtualized networked systems. Therefore, virtual topology embedding among multi-domain wide-area networks will appear in the future works.

Finally, TOP-VCM [28] models the parallel applications as the CPU and links demands, but it doesn't consider the type of the parallel jobs: computation and communication intensive. Schedule the computation intensive jobs to a local network with higher computing capacity, while allocate the communication intensive jobs to such local network with higher communication capacity.

6.2. Conclusion

Most people agree that Cloud computing as a revolutionary technology greatly promotes the development of the information society. Network virtualization is also an especially important technology to fight back the ossification of the current internet. VN is a fundamental and necessary basis in cloud computing and virtualization environment, which further develops along with the Cloud Computing and virtualization. The virtual network embedding/mapping as one of the most important steps to setup a virtual network and core technologies will get long-term interest. It makes great contributions to provide a high quality cloud value-added service to end users.

In this paper, we have surveyed the past and the state of the art VNE research. We also performed many detailed analyses and comparisons amongst the existing VNE models, and finally promote our proposals and trends. It's evident that VNE provides a promising virtual resource allocation and scheduling methodology about substrate resource utilization, load balancing, green computing, overload avoidance, traffic fluctuation and QoS in Cloud Computing environment.

CONFLICT OF INTEREST

The author confirms that this article content has no conflict of interest.

ACKNOWLEDGEMENTS

This work is supported by the National Natural Science Foundation of China (NSFC) (Grant No.61170004), IBM Shared University Research (SUR) project (3D514BF-41421), Specialized Research Fund for the Doctoral Program of Higher Education (20130061110052), Key Science and Technology Research Project of Science and Technology Department of Jilin Province (20140204013GX), Special Fund for Scientific Research in the Public Interest (SinoProbe09-01), Basic Scientific Research Foundation of Jilin University (450060491439) and the research supported by China Geological Survey project (12120113006300).

REFERENCES

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

[2] M. R. Rahman and R. Boutaba, "SVNE: Survivable Virtual Network Embedding Algorithms for Network Virtualization" *IEEE Transactions on Network and Service Management,* vol. 10, no.2, pp. 105-118, 2013.

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

[4] W. Szeto, Y. Iraqi and R. Boutaba, "A multi-commodity flow based approach to virtual network resource allocation." Global Telecommunications Conference, 2003. GLOBECOM'03. IEEE. Vol. 6. IEEE, 2003.

[5] J. Lu and J. Turner, "*Efficient mapping of virtual networks onto a shared substrate*", Washington University in St. Louis, Tech. Rep, 2006.

[6] Y. Zhu, and M. H. Ammar, "Algorithms for Assigning Substrate Network Resources to Virtual Network Components", *INFOCOM*, pp. 1-12, 2006.

[7] Yu, Minlan, "Rethinking virtual network embedding: substrate support for path splitting and migration", ACM SIGCOMM *Computer Communication Review*, vol. 38, no. 2, pp. 17-29, 2008.

[8] J. Lischka and H. Karl, "A virtual network mapping algorithm based on subgraph isomorphism detection", Proceedings of the 1st ACM workshop on Virtualized infrastructure systems and architectures, ACM, pp. 81-88, 2009.

[9] Y. Li, W. Li and C. Jiang, "A survey of virtual machine system: Current technology and future trends", Electronic Commerce and Security (ISECS), 2010 Third International Symposium on IEEE, pp. 332-336, 2010.

[10] N. F. Butt, M. Chowdhury and R. Boutaba, "Topology-awareness and reoptimization mechanism for virtual network embedding", Springer: Berlin Heidelberg, 2010.

[11] S. Zhang, "FELL: A flexible virtual network embedding algorithm with guaranteed load balancing", Communications (ICC), International Conference on IEEE, pp. 1-5, 2011.

[12] S. Zhang, "Opportunistic bandwidth sharing for virtual network mapping", Global Telecommunications Conference (*GLOBECOM* 2011), IEEE, pp. 1-5, 2011.

[13] X, Cheng and S. Su, "Virtual network embedding through topology-aware node ranking", ACM *SIGCOMM Computer Communication Review*, vol. 41, no. 2, pp. 38-47, 2011.

[14] A. Fischer, "Virtual network embedding: A survey", *Communications Surveys & Tutorials*, IEEE, vol. 15, no. 4, pp. 1888-1906. 2013.

[15] S. Herker, A. Khan and X. An, "Survey on Survivable Virtual Network Embedding Problem and Solutions" ICNS 2013, The Ninth International Conference on Networking and Services, pp. 99-104, 2013.

[16] C. Xiaolin, "Green cloud virtual network provisioning based ant colony optimization" Proceeding of the fifteenth annual conference companion on Genetic and evolutionary computation conference companion. ACM, pp. 1553-1560. 2013.

[17] F. Liu and X. Dong, "A novel elastic resource allocation strategy of virtual cluster" Parallel Architectures, Algorithms and Programming (PAAP), Fourth International Symposium on IEEE, pp. 168-173, 2011.

[18] J. Thomas and K. Mahadik, "Flexible resource allocation for reliable virtual cluster computing systems" Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis, ACM, p. 48, 2011.

[19] J. Xiao and Z. Wang, "A Priority Based Scheduling Strategy for Virtual Machine Allocations in Cloud Computing Environment", CSC, 2012 International Conference on IEEE, pp. 50-55, 2012.

[20] S. Chen, J. Wu and Z. Lu, "A Cloud Computing Resource Scheduling Policy Based on Genetic Algorithm with Multiple Fitness", CIT, 12th International Conference on IEEE, pp. 177-184. 2012.

[21] M. Chowdhury, M. R. Rahman and R. Boutaba, "ViNEYard: virtual network embedding algorithms with coordinated node and link mapping", *IEEE/ACM Transactions on Networking (TON)*, vol. 20, no. 1, pp. 206-219, 2012.

[22] Z. Sheng, "An opportunistic resource sharing and topology-aware mapping framework for virtual networks", INFOCOM, Proceedings IEEE, pp. 2408-2416. 2012.

[23] A. Mansoor and T. V. Lakshman, "Network aware resource allocation in distributed clouds", INFOCOM, 2012 Proceedings IEEE. IEEE, pp. 963-971, 2012.

[24] Z. Sheng "Virtual network embedding with opportunistic resource sharing", *IEEE Transactions on Parallel and Distributed Systems* vol. 25, no. 3, 816-827, 2014.

[25] P. Chrysa, "On the optimal allocation of virtual resources in cloud computing networks." *IEEE Transactions on Computers*, vol. 62, no. 6, pp. 1060-1071, 2013.

[26] A. Leivadeas and C. Papagianni, "Efficient Resource Mapping Framework over Networked Clouds *via* Iterated Local Search-Based Request Partitioning", *IEEE Transactions on Parallel and Distributed Systems*, pp. 1077-1086, 2011.

[27] H. Yanagisawa, T. Osogami, "Dependable virtual machine allocation". INFOCOM, 2013 Proceedings IEEE, pp. 629-637, 2013.

[28] X. Wei and H. Li, "Topology-aware Partial Virtual Cluster Mapping Algorithm on Shared Distributed Infrastructures", *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 10, pp. 2721-2730, 2013.

[29] J. Castro and N. Nabona "An implementation of linear and nonlinear multi-commodity network flows", *European Journal of Operational Research*, vol. 92, no. 1, pp. 37-53, 1996.

[30] D. G. Andersen, "*Theoretical approaches to node assignment*". Dec. 2002, unpublished Manuscript.

[31] G. Wang and Z. Zhao, "A virtual network embedding algorithm based on mapping tree", ISCIT, 2013 13th International Symposium on. IEEE, pp. 243-247, 2013.

[32] J. Xue and J. You, "Nodes clustering and dynamic service balance awareness based virtual network embedding," TENCON 2013 - 2013 IEEE Region 10 Conference (31194), vol. 1, no. 4, Oct. 2013.

[33] C. Wang and S. Shanbhag, "Virtual network mapping with traffic matrices", IEEE International Conference on ICC, pp. 2717-2722, 2012.

[34] I. Houidi, W. Louati and D. Zeghlache, "A distributed virtual network mapping algorithm". Communications, ICC'08. IEEE International Conference on IEEE, pp. 5634-564, 2008.

[35] Z. Wang and Y. Han, "Virtual network embedding by exploiting topological information", GLOBECOM, IEEE, pp. 2603-2608, Dec, 2012.

[36] F. Ilhem, "Adaptive-VNE: A flexible resource allocation for virtual network embedding algorithm" Global Communications Conference (GLOBECOM), IEEE, 2012.

[37] L. R. Bays, "Security-aware optimal resource allocation for virtual network embedding" Proceedings of the 8th International Conference on Network and Service Management. International Federation for Information Processing, pp. 378-384, 2012.

[38] A. Karmouch, "Cost-Efficient Mapping for Fault-Tolerant Virtual Networks," *IEEE Transactions on Computers*, 11 Feb. 2014.

[39] S. Zhang, X. Qiu and L. Meng, "Virtual network mapping algorithm for large-scale network environment", Communications and Networking in China (CHINACOM), 2011 6th International ICST Conference on, pp. 765-770, 17-19 Aug. 2011.

[40] D. Dietrich and A. Rizk, "AutoEmbed: automated multi-provider virtual network embedding" Proceedings of the ACM SIGCOMM 2013 conference on SIGCOMM, *ACM*, pp. 465-466, 2013.

[41] F. Yang and Z. Wang, "Vlb-vne: A regionalized valiant load-balancing algorithm in virtual network mapping" IEEE International Conference on WCNIS, pp. 432-436, 2010.

[42] C. Werle, P. Papadimitriou, I. Houidi, W. Louati, D. Zeghlache, R. Bless, and L. Mathy, "Building virtual networks across multiple domains", *ACM SIGCOMM Computer Communication Review*, ACM, vol. 41, no. 4, pp. 412-413, 2011.

[43] L. Xu and Z. Zeng, "Multi-objective Optimization Based Virtual Resource Allocation Strategy for Cloud Computing", IEEE/ACIS 11th International Conference on ICIS, pp. 56-61, 2012.

[44] Z. Xiao and W. Song, "Dynamic Resource Allocation Using Virtual Machines for Cloud Computing Environment", *IEEE Transactions on Parallel and Distributed Systems*, pp. 1107-1117, June 2013.

[45] N. Bobroff and A. Kochut, "Dynamic placement of virtual machines for managing sla violations", Integrated Network Management, 2007. IM'07. 10th IFIP/IEEE International Symposium on IEEE, pp. 119-128, 2007.

[46] D. N. B. Ta and T. Nguyen, "A virtualization-based approach for zone migration in distributed virtual environments", Proceedings of the 4th International ICST Conference on Simulation Tools and Techniques. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), pp. 249-256, 2011.

[47] C. Jo, E. Gustafsson, J. Son, and B. Egger, "Efficient live migration of virtual machines using shared storage", Proceedings of the 9th ACM SIGPLAN/SIGOPS international conference on Virtual execution environments. ACM, vol. 48, no. 7, pp. 41-50, 2013.

[48] S. Cabuk, C. I. Dalton, H. Ramasamy, and M. Schunter, "Towards automated provisioning of secure virtualized networks", Proceedings of the 14th ACM conference on Computer and communications security. ACM, pp. 235-245, 2007.

[49] Y. Donggyu and Y. Yi, "Virtual network embedding in wireless multihop networks", Proceedings of the 6th International Conference on Future Internet Technologies ACM, pp. 30-33, 2011.

[50] R. Mangharam and M. Pajic. Demo abstract: "Embedded Virtual Machines for wireless industrial automation //Proceedings of the 2009 International Conference on Information Processing in Sensor Networks. *IEEE Computer Society*, pp. 413-414, 2009.

[51] S. Even, A. Itai and A. Shamir. "On the complexity of time table and multi-commodity flow problems", IEEE 16th Annual Symposium on Foundations of Computer Science, pp. 184-193. 1975

[52] A. Gupta, J. Kleinberg, A. Kumar, R. Rastogi, and B. Yener, "Provisioning a virtual private network: a network design problem for multicommodity flow", Proceedings of the thirty-third annual ACM symposium on Theory of computing, ACM, pp. 389-398, 2001.

Received: September 22, 2014 Revised: November 03, 2014 Accepted: November 06, 2014