International Journal of Grid and Distributed Computing
Vol. 9, No. 11 (2016), pp.221-234
http://dx.doi.org/10.14257/ijgdc.2016.9.11.19

Research on Multipath Virtual Network Mapping Based on Spatial and Temporal Correlation

Min Zhang [1], Xijie Tang [2], Ming Jiang[3], Qicheng Wang[4]

[1,2,3]*Institute of Software and Intelligent Technology, Hangzhou Dianzi University, Hangzhou 310018,China*

[4] *Hakim Information Technology Co., Ltd*

[1]*hz_andy@163.com,*[2] *141050065@hdu.edu.com,*[3]*13588161992@163.com*

Abstract

Network virtualization is an important technology that allows multiple heterogeneous networks running simultaneously share substrate network, it can effectively solve the problem of network ossification. Earlier research has made many heuristic algorithms, but it does not consider the time factor in the embedding process, existing algorithm often make decision according to the present available substrate network resource. This paper, we first explore the time and space of the two dimensional network model, form the perspective of the two-dimensional space-time, we proposed two-dimensional discrete weighted method and multi commodity flow virtual network embedding with time-varying model. We also proposed a virtual network algorithm based on space–time. The simulation result show that the proposed algorithm can effectively utilize the network resources and make substrate network resources maintain the load balance for a long term.

Keywords*: Network virtualization, time, virtual network embedding, discrete weighted*

1. Introduction

In the past few decades, the Internet architecture supports a variety of distributed applications and heterogeneous network and that’s quite successful. But now, with the increasing scale of the network, the popularity of the Internet has become the biggest obstacle to its development and innovation. Network virtualization as an important technology of the future network, has gained great attention in the past few years [1] [2]. It allows multiple heterogeneous virtual networks(VNs) to run on a shared physical network at the same time for network innovation and service upgrades, Network virtualization can effectively solve the network ossification problem[3][4], and it plays an important role in some emerging areas, such as Cloud Computing, Data Center Networks and Software Define Networks. In the network virtualization environment, the traditional Internet service providers (ISPs) are mainly divided into two roles: the infrastructure providers (InPs) that deploy and manage the physical network and the service providers (SPs) that build virtual networks. Service providers hire physical resources from infrastructure providers to build a virtual network (VN) and provide end to end service for the user.

A fundamental problem in network virtualization is the virtual network embedding (VNE) problem, and the essence of virtual network mapping actually is the issue of resource allocation, a new virtual network node and link mapping to the underlying network special physical nodes and links along the way. Due to the constraints of nodes CPU and links bandwidth resources as well as the diversity of the network topology, the problem of virtual network embedding is proved to be a NP-hard problem [1]. In recent

ISSN: 2005-4262 IJGDC

years, researchers have proposed a number of heuristic algorithms, these heuristic can be divided into two categories: a stage of virtual network embedding [6]-[9] and two stage of virtual embedding [10]-[13]. For two stage virtual network embedding, the first to use greedy algorithm mapping the virtual nodes. Then link embedding use the shortest path algorithm for unsplitting path. Most link embedding problem for path splitting can be reduced to the multi-commodity flow problem.

At present, these studies have made pretty good progress, however, these existing algorithm usually ignore the time attribute of the mapping process. In reality, the VN arrived in an uncertain time and have different lifetime spans. Further the VN dynamic arrival will occupy physical network resources, physical network resources will be released while leaving, which will lead to physical network resource is constantly change with the time , so the network resource status is not the same within different time. Yet a series of algorithms that many researchers had been proposed only considered one-dimensional spatial attributes, various assumptions and constraints are based on the one-dimensional space of the CPU, bandwidth, topology and other attributes. The virtual network embedding without from the time dimension, which is must lead to gradual deterioration of the utilization of physical resource over time.

Aiming at this problem, we study the problem of online virtual network mapping based on two dimensions of space and time, which makes the underlying network can maintain the load balance in the long term. This paper form the perspective of the two-dimensional space-time, we made a two-dimensional discrete weighted temporal attribute and spatial resource integration method. Based on the idea of path splitting and multi-path, we proposed the multi commodity flow virtual network mapping model with time varying. And we also propose the mapping algorithm based on two-dimensional space-time virtual network. Simulation results show that the proposed algorithm can improve the virtual request acceptance rate and maintain the load balance of the underlying network resources for a long time.

This paper is organized as follows: Section 2 describes related work, Section 3 gives network model and virtual network mapping problem description, Section 4 presents its own mapping model and algorithms, we evaluate algorithm performance in Part 5, Part 6 is the conclusion of this article.

2. Related Work

The VN embedding problem has both online-routing and off-line routing features: within the same VN request, all the VN requests are known in advance(similar to off-line routing)[10],[12], while no information is available for future VN requests (similar to on-line routing)[8][9].

Author in [14] consider the virtual network do not have to be static, it can over time and even include certain temporal flexibilities, and they propose a continuous-time mathematical programming approach to slove the temporal VN embedding problem.

The authors in [15] consider the time attribute of the virtual network embedding, and presented a probability model which is formulated to obtain the maximum probability that the available resources of substrate network can be used by succeeding VN requests. And based greedy algorithm embedding the VN.

The advantages of multipath can be shown in many areas, such as load balancing, path splitting mechanism can be integrated more bandwidth resource, the substrate network could accept more virtual request. Some studies introduce the idea of path splitting. In [16], author present an algorithm of link embedding, which is based on greedy k-shortest paths for path splitting. In [11], when VN request acceptance division, the link mapping problem could transform to multi commodity flow problem, the authors assume that a virtual link can be embedded to different substrate paths. Joint nodes and links are embedding simultaneously, proposed two mixed integer programming model in [8]. The goal of these algorithms to achieve near-optimal virtual network mapping scheme.

3. Network Model and Problem Description

As mentioned before, the traditional VN embedding without considering the embedding real-time, in other words, the time attribute of virtual network mapping constraints are not included in the existing work. In this section, we first describe the network model of substrate network and virtual networks request that capture the dual nature of time and resources.

3.1. Substrate Network Model

We model the topology of the substrate network as a weighted undirected graph $G_s = (N_s, E_s, A_s^n, A_s^e)$, where N_s is the set of substrate nodes and E_s is the set of substrate links between nodes of the set N_s. We use subscript to refer to substrate or virtual network, and use superscript to refer to nodes or links, unless otherwise specified. Substrate nodes and links are associated with their attributes, denoted by A_s^n and A_s^e, respectively. And $A_s^n = \{A_{s(0)}^n, A_{s(1)}^n, A_{s(2)}^n, \cdots\cdots, A_{s(n)}^n, \cdots\}$,

$A_s^e = \{A_{s(0)}^e, A_{s(1)}^e, A_{s(2)}^e, \cdots\cdots, A_{s(m)}^e, \cdots\}$. Where $A_{s(n)}^n$ refers to CPU resource capacity of substrate node at time n, and $A_{s(m)}^e$ refers to bandwidth resource capacity of substrate link at time m. We also denote P is the set of all loop-free paths in the substrate network. Figure. 1 shows a substrate network, where the numbers over the links represent available bandwidths and the numbers in rectangles represent available CPU resources.

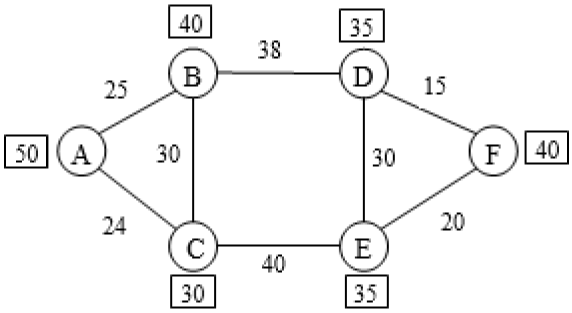


Figure 1. Substrate Network

3.2. Virtual Network Request

We model the topology of the virtual network as a weighted undirected graph $G_v = (N_v, E_v, A_v^n, A_v^e)$, where N_v is the set of virtual nodes and E_v is the set of virtual links between nodes of set N_v. We denote by A_v^n and A_v^e the set of node CPU and link bandwidth constraints, respectively. Because each VN request has a running start time and duration, so the VN request is represented by the triple $VN = \{G_v, t_a, t_b\}$,where t_a and t_d represent the start-time and lifetime of the VN request, respectively. When the VN request arrives, if the request is accepted, the substrate network allocates network resources on the substrate nodes and paths selected by that assignment for the VN Within the time constraints, and release the resources once the VN expires. When the substrate network insufficient resources in the lifetime, the substrate network rejects the VN request. Figure. 3(a) and (b) shows two VN requests with node and link constraints.

3.3 VN Embedding Problem Description

A virtual network embedding for a VN request is defined as a mapping M from G_v to a subset of G_s, $M: G_V(N_V, E_v) \rightarrow G_s^{'}(N_s^{'}, P^{'})$, where $N_s^{'}$ is the subset of N_s, $P^{'}$ is the subset of P. The VN embedding can be decomposed into two major components: node mapping and link mapping.

Node mapping refers to mapping virtual nodes to the substrate nodes. It is defined by a mapping $M_{\mathrm{N}}:(N_v, A_V^n) \rightarrow (N_s^{'}, R_s^n)$, where R_s^n is defined as the residual CPU capacity of the substrate node. $R_s^n(n_s) = A_s^n(n_s) - \sum_{\forall n_v \rightarrow n_s} A_V^n(n_v)$, where $n_v \rightarrow n_s$ denotes that the virtual node n_v is hosted on the substrate node n_s. The substrate node residual CPU capable supporting at least one virtual node request from time t_a to time $t_a + t_d$, *i.e.* $A_V^n(n_v) \leq R_{s(t)}^n(M_N(n_v)), t = t_a, \cdots, t_a + t_d$. Note: the virtual node of different virtual network request can be mapped to the same substrate node, but each virtual node from the same VN request must be assigned to a different substrate node.

Link mapping refers to each virtual link is assigned to a substrate path between the corresponding substrate nodes. It is defined by a mapping $M_e:(E_v, A_V^{\mathrm{e}}) \rightarrow (P', R_s^e)$, where R_s^e is defined as the residual bandwidth capacity of the substrate link. $R_s^e(e_s) = A_s^e(e_s) - \sum_{\forall e_v \rightarrow e_s} A_V^e(e_v)$, the substrate path residual bandwidth capable supporting at least one virtual link request from time t_a to time $t_a + t_d$, *i.e.* $A_V^e(\mathrm{e}_v) \leq R_{s(\mathrm{t})}^e(M_e(e_v))$, $t = t_a + t_d$. The paper considers the substrate path splittable. That is to say, each link can be mapped to one or more substrate path.

We should focus on the time factor in the VN embedding, an example as below. We assume the substrate network capacity is 60, there are four VN request in table 1. If we don't consider the nature of the time, according to previous embedding strategy, the VN4 can't be embedded successfully, because there is not enough capacity, as shown in Fig.2.(a). However, in fact, after have embedded the previous three VN request, VN4 can be embedded to substrate network in start time as show in Figure.2.(b). Therefore, we need to consider the space and time properties of the demand of the virtual network.

Table 1. VN Request

Virtual network	Start	lifetime	capacity
VN1	0	60	15
VN2	10	40	10
VN3	20	20	20
VN4	45	50	30

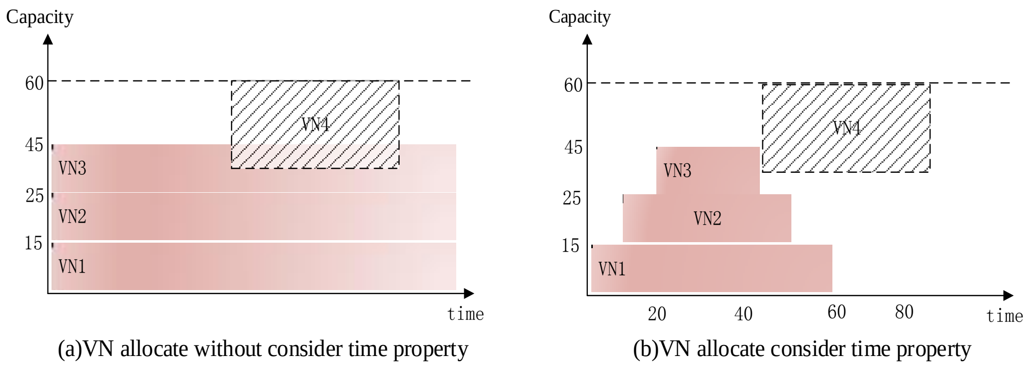


Figure 2. An Simple Example of VN Allocate

Figure. 3(c) shows the VN embedding solutions for the two VN requests. For example, in Figure.3, the VN request 1 has the node mapping $\{a \to A, b \to B\}$, it has been assigned the link mapping $\{(a,b) \to (A,B)\}$. And the VN request 2 has the node mapping $\{c \to D, d \to C, f \to F\}$, it has been assigned the link mapping $\{(c,d) \to (D,B,C),(c,f) \to (D,E,F),(D,F)\}$.

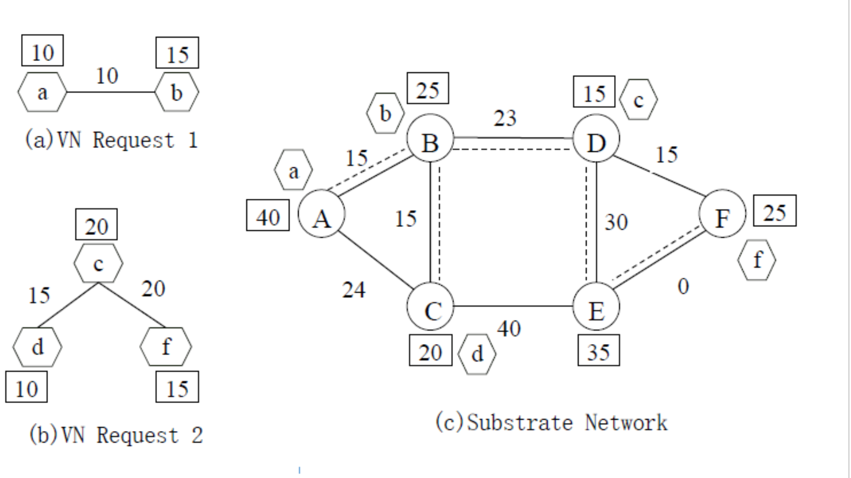


Figure3. An Example of VN Embedding

3.4. Objectives

Our main interest in this paper is to propose an efficient online VN embedding algorithm where VN requests arrive and depart over time. We want to increase revenue in the long term, and in addition to balancing load of the substrate network resources. Similar to the previous work in [8], [11], we introduce some define as follows.

Definition 1 VN request acceptance ratio:

$$a = \lim_{T\to\infty} \frac{\sum_{t=0}^{T} m}{\sum_{t=0}^{T} n} \tag{1}$$

Where $\sum_{t=0}^{T} m$ refers to the number of successful mapping from time $t=0$ to time T,

Where $\sum_{t=0}^{T} n$ refers to the total number of VN request from time $t=0$ to time T.

Definition 2 VN request revenue:

$$R(t)=\sum_{i=0}^{i=k}\alpha_i(\alpha\sum_{n\in n_v}cpu(n,t)+\beta\sum_{e\in E_v}bw(e,t)) \quad (2)$$

Where k refers to the number of VN request in time T, $\alpha_i \in \{0,1\}$, $\alpha_i = 1$ refer to the mapping of the $i'th$ VN is successfully, $\alpha_i = 0$ refers to fails. $R(t)$ refers to the revenue of VN request at time t. $cpu(n,t)$ and $bw(e,t)$ are the CPU and bandwidth requirements for the virtual node n and the virtual link e at time t, respectively.

Definition 3 VN request cost:

$$C(t)=\sum_{i=0}^{i=k}\alpha_i(\lambda\sum_{n\in n_v}cpu(n,t)+u\sum_{e\in E_v}\sum_{P}bw(e,t)) \quad (3)$$

Where P is the substrate path that virtual link has been mapped, k refers to the number of VN request in time T, λ and u refer to the unit cost of cpu and bandwidth.

The long-term R/C radio is defined as:

$$\Gamma=\lim_{T\to\infty}\frac{\sum_{t=0}^{T}R(t)}{\sum_{t=0}^{T}C(t)} \quad (4)$$

Definition 4 Network load standard deviation

$$D=(\frac{\sqrt{\sum_{e}(w(e)-\overline{w(e)})^2}}{|E|}+\frac{\sqrt{\sum_{n}(w(n)-\overline{w(n)})^2}}{|N|})\Big/2 \quad (5)$$

Where $w(n)=\frac{A_s^n-R_s^n(t)}{A_s^n}$ and $w(e)=\frac{A_s^e-R_s^e(t)}{A_s^e}$.

4. Multipath Virtual Network Mapping Based on Spatio-Temporal Correlation

In this section, we describe the time and space two-dimensional resource model. And then introduces the multi commodity flow model based on time-space two-dimensional load balance, and describes the embedding algorithm.

4.1. Establishing Two Dimensional Resource Measurement Model about Time-Space Correlation

Most researchers only consider the resources of the underlying network at the present time when a virtual request arrives, and does not consider the amount of resources in the following time. Because of virtual request is dynamic arrival and unpredictable, it's may resources fully in the current time, but resources relative scarcity in the subsequent time, virtual request dynamic building and delete will affect the balance of the entire physical network, it will lead to overload of some nodes and links, forming a bottleneck link. In view of this, we introduce the concept of time while considering the space resource factor. We proposed two-dimensional resource measurement model based on time-space correlation.

The time of virtual network effect load balancing of the substrate network is running at its start and end time. And in the time range, the loads on the link and node are constantly changing with different virtual networks start and end. Due to the SPs can not accept the virtual request that life cycle infinitely small, we introduce the concept of time slice that we build a virtual network needs minimum time (*i.e.* minute, hour, day), the life cycle of virtual network by a plurality of time slice composition. We can treat the resources of nodes and links in each time slice as a fixed, different time slice with different resources. From the current point of view, resources load of different times slice, there is not the same influence for at the present time. The closed from the current time, the greater the impact. This is because we are pre-allocated VN resources, if we need to transfer these resources, the closed the current time, the cost will be higher.

We don't think this is a bad link that a time slice resource load is too high within a period of time, we need a metric to handle resource of the different time slice. Therefore, as show in Figure, we introduce a monotonically decreasing function $w(t)$ that satisfies $\int_0^{\infty} w(t) = 1$. This paper takes the Gaussian distribution function $w(t) = \int_0^{\infty} \frac{2}{\sqrt{\pi}} e^{-t^2} dt$ that subject to the conditions required. The start time and the end time of each time slice, respectively, as the upper and lower bounds of the integral.

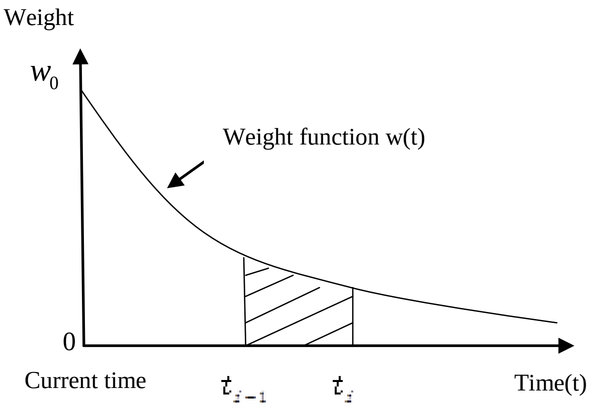


Figure 4. Weight Function Schematic

As mentioned above, we define physical node and link metrics as follows：

$$H(n_s) = \sum_{i=1}^{k} (R(n_s, t_i) \times \int_{t_i}^{t_i+T} w(t)dt) \tag{6}$$

$$H(l) = \sum_{i=1}^{n} (R(l, t_i) \times \int_{t_i}^{t_i+T} w(t)dt) \tag{7}$$

Where $R(n_s, t)$ represents the remaining resource of n_s at the time t.

We will use a simple example to illustrate, as shown in the Figure.5, we define two underlying links. Now there is a virtual request, its bandwidth requirement is 2, start time is t_0, $lifetime = 100 * slice$. Which link should we choose? Previous research strategy is based on the current time network resource capacity to choose link 1, obviously this will lead to link with a high load state in future time. If we use our model, we assure that the probability density of the first few time slice is 0.3, 0.2, 0.15, 0.1, 0.05, respectively. We could calculate $H(link1) = 3.9$, $H(link2) = 4.2$ by equation(7). According to the result of our model, we choose link 2. It is could ensure that substrate network maintain load balancing in long term.

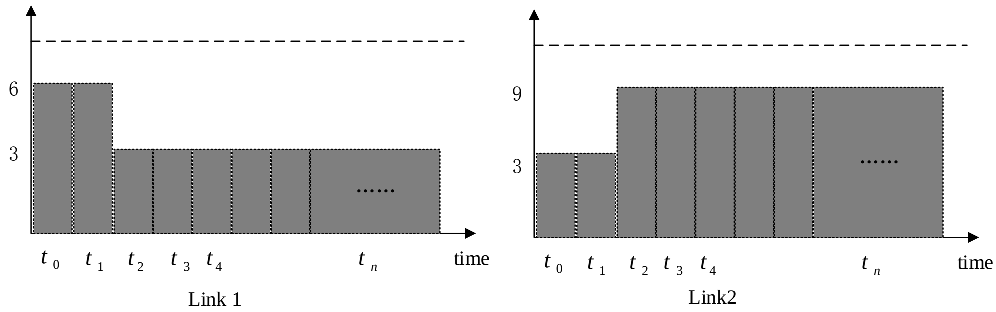


Figure 5. Two Link for Substrate Network

4.2. Multi Commodity Flow Model Based on Time and Spatial Correlation

The two endpoints of the virtual link is mapped to physical node pair (s_k, t_k) when mapping virtual network, virtual link mapping based multi-commodity flow to look for one or several physical paths between the physical node pair, which makes the bandwidth of these paths equal to the bandwidth of virtual link request.

The goal of the multi commodity flow algorithm is to minimize the cost of the transport these commodity, and the cost is determined by the transport flow and physical link cost coefficient product. Considering the time factor, each link available capacity is not the same at different times. The residual bandwidth of the physical link is changed with time, according to the model of upper section, a two-dimensional metric about resource and time is obtained, and the reciprocal of the metric is used as the cost coefficient of the physical link. Since residual bandwidth changes over time, cost coefficient is change over time, so that you can ensure that physical link the more residual bandwidth and the less cost coefficient, can has the priority to be selected, which can ensure the load balance of the physical network.

The linear programming of multi commodity flow model based on the time (T-MCF) as follows:

Objective function:

$$Min \sum_{(u,v)\in l} c(u,v) * \sum_{i=1}^{k} f_i(u,v) \tag{8}$$

Constraints:

$$\sum_i (f_i(u,v) + f_i(v,u)) \le R(l_{uv}, t_j), \forall l_{uv} \in l_s \quad \forall n, v \in N_s \tag{9}$$

$$\sum_{w\in N} f_i(s_i, w) - \sum_{w\in N} f_i(w, s_i) = d_i \tag{10}$$

$$\sum_{w\in N_s} f(t_i, w) - \sum_{w\in N_s} f(w, t_i) = -d_i \tag{11}$$

$$\sum_{w\in N_s} f_i(u, w) - \sum_{w\in N_s} f_i(w, u) = 0, u \in N_s / \{s_i, d_i\} \tag{12}$$

$$f_i(u,v) \ge 0, \forall u, v \in N_s \tag{13}$$

Where $c(u,v) = \alpha * \frac{1}{H(l)+\delta}$, $f_i(u,v)$ represents the link (u,v) allocated bandwidth for $i'th$ commodity. $R(l_{uv}, t_j)$ indicates the residual bandwidth of link (u,v) in the time

t_j . $\{s_i, t_i, d_i\}$ represents source node, destination node, and demand of commodity, respectively.

Summing up $f_i(u,v)$ 和 $f_i(v,u)$ in the constraint (9) ensure that the summing of flow on both directions of the undirected edge (u,v) remains within its available bandwidth for each time slice during VN lifetime.

Constraint set (10), (11), and (12) refer to the flow conservation conditions, which denote that the net flow to a node is zero, except for source node s_i and the destination node d_i. Constraint(13) denote each flow non-negative constraint.

4.3. Virtual Network Embedding Algorithm with the Time

In this section, similar to two-stage VNE algorithm [12], we use T-MCF model and propose VN embedding algorithm based on time with substrate network that supports path splitting.

For node mapping, as the algorithm show, we introduce node two-dimensional load intensity metrics $w_n = \sum_{i=1}^{k}(\frac{C(n_s) - R(n_s, t_i)}{C(n_s)} \times \int_{t_i}^{t_i+T} w(t)dt)$ (14). We sort the nodes non-increasing sequence of resource and then mapping the sorted nodes to physical node. For each virtual node, we find node set from physical node that meet the virtual node resource constraints and has not been mapped, and calculate the two-dimensional load intensity of these physical node, so as to ensure node resource maintain load balancing. If all virtual nodes are mapped successful, continue to the next, otherwise node embed failed and exit algorithm. For the link embedding at step 10, we create the T-MCF model and find the minimum cost of the virtual embedding, if we can get the minimum cost, the link mapping is successful, then we update the substrate network resource.

Algorithm VNE algorithm with time

Input: G_s : current substrate network;

$VN_i = \{G_v^i(N_v, E_v), t_a, t_b\}$: the i^{th} arriving VN request;

1: Sort the VN nodes of VN_i according to cpu resource, Store the ordered VN nodes into set N_{sort} ;

2: **for** each unmapped $n_v \in N_{sort}$ **do**

3: Find all n_s with satisfies n_v resource constrain, store into set S ;

4: Compute W_{n_s} according to equation(14);

5: Mapping n_v to the substrate node with the least W_{n_s} and unmapped;

6: **if** the node n_v mapped failed **then**

7: **return** NODEMAPPED_FAILED

8: **end if**

9: **end for**

10: Solve T-MCF to map virtual edges

11: **if** T-MCF succeeded **then**

12: Update all state of substrate nerwork

13:**else**

14： **return** EDGEMAPPED_FAILED

15：**end if**

5. Performance Evaluation

In this section, we first describe our simulation environment, and then introduce some compare algorithm with existing. Lastly, we present our evaluation result.

5.1 Evaluation Environments

We have implemented discrete event simulator based on c++ to evaluate performance our embedding algorithm. We use GT-ITM tool[17] to generate substrate and virtual network. Similar to previous work [8], substrate network randomly generated 100 nodes, about 500 links, the scale similar to a medium-sized ISP. Each pair of substrate nodes is randomly connected with probability 0.5. The CPU and bandwidth of the substrate network node and link are real numbers uniformly distributed between 50 and 100. We assume that VN requests arrive and start running time obey Poisson distribution with $\lambda = 15$ time units, and each VN duration obey exponential distribution with an average of $u = 1000$ time units. In each VN request, the number of virtual nodes randomly obtain and it obeys the uniform distribution from 4 to 10, and the probability of random connection between each pair of nodes is 0.5, the cpu and bandwidth requirements of the VN node and link are real number uniformly distributed between 0 and 50. We have used the open source mixed integer programming library glpk[18] to solve T-MCF.

5.2 Comparison Method

In this paper, in order to better illustrate the advantages of our proposed algorithm, compared with the previous algorithms in the study [10] [11]. As shown in the table 2, a greedy node embedding and link mapping by shortest path algorithm G-SP [10]; Greedy node mapping with embeds the link by MCF model [11]. In order to evaluate our experiments, we used several performance indicators, including virtual request acceptance radio, revenue cost radio R/C and network equilibrium.

Table 2. Compared Algorithms

Notation	Algorithm description
G-SP	Greedy node mapping with shortest path based link mapping
MCF	Greedy node mapping with splittable link embedding using MCF
T-VNE	Node mapping based on node stress with embeds the link by T-MCF model

5.3 Evaluation Results

We use several performance metrics for evaluation purposes in the experiment. It including the VN request acceptance radio and revenue cost radio. We also measure the network load standard deviation. In all these cases, the performance metrics are plotted against time to show how each of these algorithms actually performs in the long run. The key observations are summarized in the following.

Figure.6 and Figure.7 show that T-VNE lead to better acceptance radio than existing algorithms. From Figure.6, we could see with the VN request number gradually increased, the acceptance radio decreased and gradually to reach the steady. In Figure.7, with VN request arrived rate increased, the acceptance radio would decreased. No matter what, T-VNE have better acceptance. This is because we not only consider the network resources at current time, but also consider the impact of future time network resources for current time.

Figure.8 depict that T-VNE leads to higher revenue cost radio. Since the T-VNE algorithm consider substrate network resource at each time in the node and link embedding, it can take advantage of more network resources, lead to accept more VN request to obtain greater revenue under the same conditions of network resources.

Figure.9 depict that T-VNE have lower network load standard deviation than other algorithm. The lower load standard deviation shows that the network resources are more balanced and stable, it will get higher acceptance rate and revenue in the future.

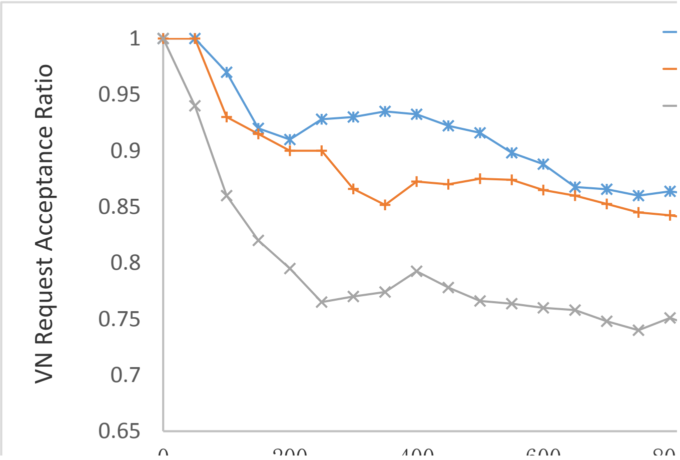


Figure 6. VN Request Acceptance Radio

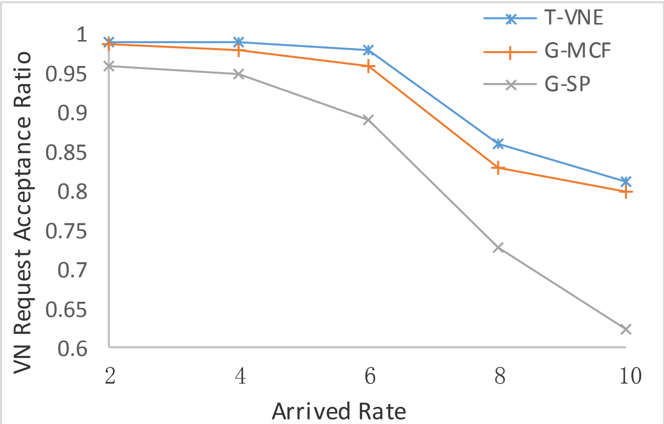


Figure 7. VN Request Acceptance Ratio

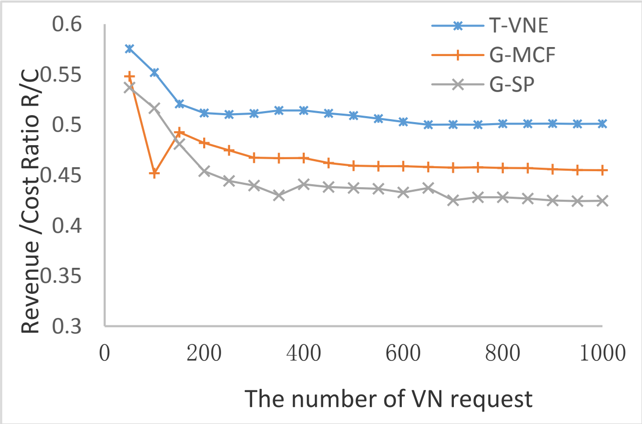


Figure 8. The Revenue/Cost Ratio

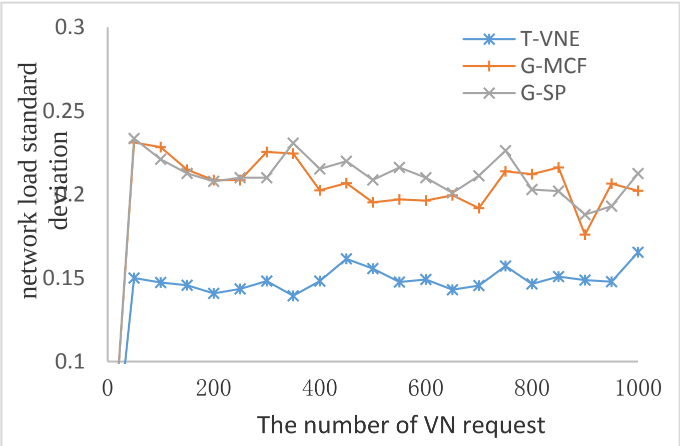


Figure 9. Network Load Standard Deviation

6. Conclusion

This paper firstly studies the present situation of virtual network embedding, then we points out the problem of ignoring the time factor of virtual network embedding. To address this problem, we consider two-dimensional space-time property and we proposed a virtual embedding algorithm for link splitting based on time. Finally, we design experiments to verify the correlation algorithm, simulation result show that the proposed algorithm have better performance.

Acknowledgments

This work is supported by 863 Program(2015AA011901、2015AA015602) and Zhejiang Province Science and technology project (No. 2014C01051) and Zhejiang Province Key scientific and technological innovation team (2013TD20).

References

[1] N. Feamster, L. Gao and J. Rexford, 'How to lease the internet in your spare time', Acm Sigcomm Computer Communication Review, vol. 37, no. 1 (2007), pp. 61-4.

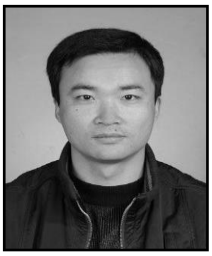
[2] C. Guo, G. Lu, H. J. Wang, S. Yang, C. Kong, P. Sun, W. Wu and Y. Zhang, 'SecondNet: a data center network virtualization architecture with bandwidth guarantees', Proceedings of the 2010 ACM Conference on Emerging Networking Experiments and Technology, CoNEXT 2010, Philadelphia, PA, USA, November 30 - December 03, 2010, (2010), pp. 620-2.

[3] N. M. M. K. Chowdhury and R. Boutaba, 'A survey of network virtualization ', Computer Networks, vol. 54, no. 5 (2010), pp. 862-76.
[4] A. Fischer, J. F. Botero, M. Till Beck and H. De Meer, 'Virtual Network Embedding: A Survey', IEEE Communications Surveys & Tutorials, vol. 15, no. 4 (2013), pp. 1888-906.
[5] A. Wang, M. Iyer, R. Dutta, G. N. Rouskas and I. Baldine, 'Network Virtualization: Technologies, Perspectives, and Frontiers', Journal of Lightwave Technology, vol. 31, no. 4 (2013), pp. 523-37.
[6] I. Houidi, W. Louati and D. Zeghlache, 'A Distributed Virtual Network Mapping Algorithm', Communications, 2008. ICC '08. IEEE International Conference on, (2008), pp. 5634-40.
[7] 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, (2010), pp. 81-8.
[8] N. M. M. K. Chowdhury, M. R. Rahman and R. Boutaba, 'Virtual Network Embedding with Coordinated Node and Link Mapping', Proceedings - IEEE INFOCOM, vol. 20, no. 1 (2009), pp. 783-91.
[9] M. Chowdhury, M. R. Rahman and R. Boutaba, 'ViNEYard: Virtual Network Embedding Algorithms With Coordinated Node and Link Mapping', IEEE/ACM Transactions on Networking, vol. 20, no. 1 (2012), pp. 206-19.
[10] Y. Zhu and M. Ammar, 'Algorithms for Assigning Substrate Network Resources to Virtual Network Components', Proceedings - IEEE INFOCOM, (2006), pp. 1-12.
[11] 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 (2015), pp. 17-29.
[12] J. Lu and J. Turner, 'Efficient Mapping of Virtual Networks onto a Shared Substrate', Washington University in St Louis, (2006).
[13] R. Ricci, C. Alfeld and J. Lepreau, 'A Solver for the Network Testbed Mapping Problem', Acm Sigcomm Computer Communication Review, vol. 33, no. 2 (2003), p. 2003.
[14] M. Rost, S. Schmid and A. Feldmann, 'It's About Time: On Optimal Virtual Network Embeddings under Temporal Flexibilities', Proceedings of the 2014 IEEE 28th International Parallel and Distributed Processing Symposium, (2014), pp. 17-26.
[15] M. Jiang, Z. Zhao, M. Zhang, J. Tang, W. U. Chunming and X. Min, 'A Virtual Network Mapping Algorithm Based on Time', Chinese Journal of Electronics, no. 1 (2014), pp. 31-6.
[16] Z. Zhang, X. Cheng, S. Su, Y. Wang, K. Shuang and Y. Luo, 'A unified enhanced particle swarm optimization-based virtual network embedding algorithm', International Journal of Communication Systems, vol. 26, no. 8 (2013), pp. 1054-73.
[17] E. W. Zegura, K. L. Calvert and S. Bhattacharjee, 'How to model an internetwork', Fifteenth Joint Conference of the IEEE Computer & Communications Societies Conference on the Conference on Computer Communications-volume, (2010), p. 594.
[18] " GNU Linear Programming Kit," http://www.gnu.org/software/glpk/.

Authors

Min Zhang, received the Ph.D.degree in computer science from Zhejiang University in 2012, and currently works at College of Computer Science at Hangzhou Dianzi University. His research areas include network virtualization, software defined network and Internet QoS provisioning.

Xijie Tang, currently is a postgraduate of College of Computer Science at Hangzhou Dianzi University. His research areas include network virtualization, software defined network.



Ming JIANG, received the Ph.D.degree in computer science from Zhejiang University in 2004, and currently is a professor of College of Computer Science at Hangzhou Dianzi University. His research areas include network virtualization, software defined network and Internet QoS provisioning.

Qicheng Wang, received bachelor's degree from Zhejiang University in 2002, and currently is the president of Hakim Information Technology Co., Ltd, responsible for the cloud data center technology research and development.

