International Journal of Grid and Distributed Computing
Vol. 9, No. 7 (2016), pp.101-110
http://dx.doi.org/10.14257/ijgdc.2016.9.7.11

Energy-saving Methods in the Operation of Virtual Networks Based on Divided Time Slices

HU Ying[1] ZHUANG Lei[2] HU Yu-xiang[3] MA Ding[4]

[1]School of Information Engineering, Zhengzhou University, Zhengzhou 450001, China
[2]School of Information Engineering, Zhengzhou University, Zhengzhou 450001, China
[3]National Digital Switching System Engineering & Technology Research Center, The PLA Information Engineering University, Zhengzhou 450002,China
[4]School of Information Engineering, Zhengzhou University, Zhengzhou 450001, China
College of Information Science and Engineering, Henan University of Technology, Zhengzhou 450000, China

Abstract

Reducing energy consumption has also become an important way to reduce expenditures and increase revenue for ISP. For solving the problem of dynamically changing network-flow of virtual network in runtime, we first analyze relations and differences between virtual network and IP network on energy-saving issues. We secondly model the makeup of energy consumption in virtual network, and determine the object of energy-saving in virtual network at. Finally we propose the solutions to this energy-saving problem. The solutions can be described as follows: (1) based on the historical data of daily network-flow, we divide one day into two time slices, and one is called idle time slice, the other is called busy time slice; (2) we design the dynamic switch-off algorithm and run it at the beginning of idle time slice, and we use former algorithms to be started at the beginning of busy time slice; (3) in the dynamic switch-off algorithm we determine the method to select virtual networks and the method to remap; (4) we apply roulette wheel to select virtual networks, in the selection method we consider both current influences on energy-saving and influences in the future. Simulation results show that the dynamic switch-off algorithm can effectively select virtual networks, and the whole solution performs well in energy aware virtual network in runtime.

Keywords: *energy-saving; virtual network; in operation; time slice*

1. Introduction

With the rising electricity costs, Internet Service Providers (ISP) is aware of the importance of energy management. The Internet was designed for the peak load and has ensured the constant pattern of network operation, which leads to low resource utilization. Statistically, the average link utilization of large ISP's Internet backbone network is from 30% to 40% [1]. Low utilization causes the waste of energy, so it is important to reduce cost for ISP. Therefore, the energy-saving problem in network is becoming more and more popular in academia [2][3]. According to relevant research, energy consumption costs have accounted for about 12% to 20% of the total costs in data center, and it accounts for 40% to 50% of running costs [4-6].

Network virtualization is the key to solve the problem of network rigid, and it makes energy aware mapping possible. There are a lot of valuable research on energy aware virtual network mapping [7-11]. In this paper we consider energy-saving virtual network

ISSN: 2005-4262 IJGDC

operation. The number of network resource demands fluctuates widely in a day. If we no longer change the mapping results after the deployment of virtual networks, the operation of virtual networks will cause great waste of energy.

We propose energy-saving algorithm to get corresponding topology in each time slice. In this paper, we put forward a novel method, named energy-saving in virtual network operation based on divided time slice (ESVNO-DTS), to solve the problems above.

The rest of this paper is organized as follows. In section 2, we introduce the relations and differences between virtual network and IP network on energy-saving issues. The ESVNO-DTS algorithm is discussed in section 3. The simulation and analysis are presented in section 4. In section 5, we give the conclusions.

2. Comparison between Virtual Network and IP Network

Virtual network is a new type of network architecture. There are some relations between virtual network and Internet: the demands of network traffic in different network architecture are the same. Here, we substitute the demands of network traffic in IP network for the demands of network traffic in virtual network.

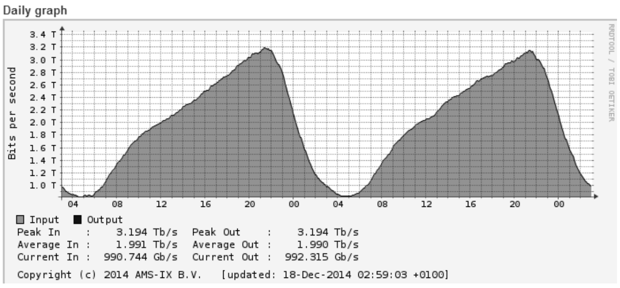

Figure 1. Daily Network-Flow Fluctuations of IP Network by Statistics of Amsterdam in the Netherlands

Figure 1 shows daily network-flow fluctuations of IP network by statistics of Amsterdam in the Netherlands [12]. Comparing the chart of different network-flow fluctuations in different days, we can see that the daily network-flow fluctuations are similar. By this variation rule, we can divide a day into multiple time slices and design the network topology within each time slice to save energy as much as possible. Because of the different network realization mechanism, there are fundamentally differences in the problem of energy-saving between IP network and virtual network.

Firstly, each type of service is corresponding to one virtual network in virtual network, and each virtual network occupies a part of the physical network. If there is no arrive or depart event in the physical network during runtime, this part is relatively fixed. The scope of each virtual network flow is limited while the scope of IP network flow is the entire network.

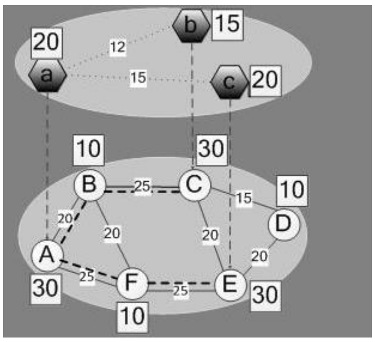

Figure 2. Virtual Network Embedding

Moreover, the resources that each virtual network occupies can be overlapping, as shown in figure 2. There can be zero, one or many virtual networks embedded in one physical node (or one physical link). The amount of network-flow on the physical node (or the physical link) is determined by multiple virtual networks which are embedded on the node (or the link). It has no direct relations with the overall network traffic.

3. ESVNO-DTS

The objective and constraints of energy-saving model are the same as [13].

3.1 The Whole Process of ESVNO-DTS

For reducing dynamic changes as much as possible, we divide one day into two time slices. One is the busy time slice, and the other is the idle time slice. Because the fluctuations of daily network-flow rate are not completely the same, so we can not foretell the division moment, but rather adaptively change the division moment according to the actual flow fluctuations. The methods for division are outlined as follows: (1) we set $\alpha(\alpha < 1)$ as a threshold for time slice division, which can be determined by the historical data (the determination method of α is described in section 4); (2) we determine if it is in the down phase by historical network-flow curve, and we determine if the value of current network-flow is less than α times of peak value of historical flow, if this is the case, we run the dynamic switch-off algorithm which will be discussed in next sections; (3) we determine if it is in the up phase by historical network-flow curve, and we determine if the value of current network-flow is more than α times of the peak value of historical flow, if this is the case, we reassign the original virtual network mapping results.

The process of ESVNO-DTS is described as follows:

Algorithm 1 ESVNO-DTS (energy-saving in virtual network operation based on divided time slice)

input： G^S ,G^V ,f

output：f

While 1 do

If in the night && in the historical flow decline phrase && the value of current network flow <= α times of peak value of data fitting flow chart(described in section 4) then

f’’=f;//save current mapping results

All virtual network requests topology unchanged， the amount of requested resource is set to α times of the original requests， then get $G^{V'}$;

f=dynamicswitch-off (f, G^S ,$G^{V'}$);

End if

If in the daytime && in the historical flow rising phrase && the value of current flow >= α times of the peak value of data fitting flow chart(described in section 4) then

f=f’’;//reassign the original mapping results

End if

End while

The key to the algorithm lies in the design method of dynamic switch-off algorithm. We will discuss this issue in the next sections.

3.2 Dynamic switch-off Algorithm

3.2.1 Whole Process of Dynamic Switch-off Algorithm: When the requested resources of current virtual networks decrease to α times of the original requested resources, the activated resources are relatively redundant to the current need of resources. The virtual networks should be selectively remapped to make the resources more concentrated. If the remapping unit is virtual link, changes will be limited, so we select the virtual network as the mapping unit.

The whole process of dynamic switch-off algorithm is shown as follows:

FUNCTION dynamicswitch-off(f, G^s ,G^v):

Do

Set roulette probability $P_1^{roulette}$, $P_2^{roulette}$,…, $P_{|request|}^{roulette}$ of mapped virtual networks ;

Create random number ran;

If $p_i^{roulette} \leq ran \leq p_{i+1}^{roulette}$ then select virtual network i ; end if

release virtual network i ;

Remap virtual network i using the former energy-aware virtual network embedding algorithm to get f' ;

until the remapping results have not been changed for successive two times

Return f';

It is important to note that we may repeatedly select the same virtual network, which may leads to early exit. In order to avoid this, we should record the IDs of selected virtual networks, and these virtual networks should be no longer involved in the selection process.

Next, we will discuss the method of selecting the virtual networks.

3.2.1 The Selection Method: The roulette wheel selection probability of each virtual network can only be get from the angle of released device number. Virtual networks which can release more resources have a higher possibility of being selected. Such is a greedy algorithm. In order to save energy much more efficiently, we should amplify the selection possibility of the virtual networks which may release more resources in the next few steps. We design the method as follows: (1)sort each physical device by the number of virtual networks it bears in descending order; (2) each physical device is endowed with a weight which is determined by its serial number; (3)add the weights of physical devices on which the virtual network is embedded together. Next, we analyze the roulette wheel selection probability of each virtual network.

The $R_{link}^{vl_i}$ denotes physical link release effect of virtual link vl_i , the effect is computed by the weight of the physical link on the path:

$$R_{link}^{vl_i} = \sum_{j=1}^{length} o_{sl_j} \tag{1}$$

The sl_j denotes the jth physical link of the physical path sl on which virtual link vl_i is embedded; the $length$ denotes the length of the physical path sl; the O_{sl_j} denotes the weight of physical link sl_j.

Analogously, the $R^{vl_i}_{forwardingnode}$ denotes forwarding node release effect of virtual link vl_i, the effect is computed by the weight of the forwarding node on the physical path:

$$R^{vl_i}_{forwardingnode} = \sum_{j=1}^{length+1} O_{sn_j} \tag{2}$$

The sn_j denotes the jth physical forwarding node on the physical path sl on which virtual link vl_i is embedded; the O_{sn_j} denotes the weight of physical forwarding node sn_j.

The $R^{vn_i}_{hostnode}$ denotes release effect of virtual node vn_i which is computed by the weight number of the physical node on which virtual node vn_i is embedded:

$$R^{vn_i}_{hostnode} = O_{sn_{vn_i}} \tag{3}$$

The sn_{vn_i} denotes the physical node on which virtual node vn_i is embedded; The $O_{sn_{vn_i}}$ denotes the weight of physical node sn_{vn_i}.

Define the release effect of a virtual network:

$$R^{request} = \sum_{i=1}^{|vn|} R^{vn_i}_{hostnode} \times \omega_b + \sum_{i=1}^{|vl|} R^{vl_i}_{forwardingnode} \times \omega_n + \sum_{i=1}^{|vl|} R_{link} \times \omega_{lb} \tag{4}$$

The $|vn|$ denotes the total number of virtual nodes in the virtual network; the $|vl|$ denotes the total number of virtual links in the virtual network; the ω_b, ω_n and ω_{lb} denotes correlation coefficient.

We can obtain Roulette probability of the virtual network by the release effects of all the virtual networks:

$$P_i^{roulette} = \frac{R_i^{request}}{\sum_{j=1}^{|Request|} R_j^{request}} \tag{5}$$

The $|Request|$ denotes the total number of embedded virtual networks.

4. Evaluation

4.1 Experimental Setting

This experiment is performed on the computer which has 4G memory and 3.4GHz Pentium 4 CPU. The topologies of substrate network and virtual networks are generated by GT-ITM [13]. The substrate network is composed of 100 nodes and 570 links. The physical nodes connect to each other with a probability of 0.5. The weights (resources) on the nodes as well as links are uniformly distributed between 50 and 100. There are 400 virtual networks. Similarly, Nodes in the virtual networks vary from 2 to 10 connected to each other with a probability of 0.5. The weights (resources) on the virtual nodes are uniformly distributed between 0 and 10. The weights (resources) on the virtual links are uniformly distributed between 0 and 5. The arrival time follows Poisson distribution which reach 5 requests on average of 100 time units, and the lasting time of virtual network requests follows exponent distribution in which μ is set to 1000.

We initially select operation state after 30 virtual networks are mapped. We select it every 20 arrived virtual networks afterwards.

17:00-5:00 is identified as the night, and 5:00-17:00 is identified as the day time. One day is divided into two time slices as shown in figure 3.

In accord with references [14][15], the values of $P_b^i(i \in N_s)$ are the same, so are P_m, P_n and P_{1b}. Finally we take the ratio of the values. Namely, the values of P_b, P_1, P_n and P_{1b} are respectively set to 1.5,1.5,6.8 and 1.

We simulate the virtual network mapping phrase and the virtual network runtime phrase. We simulate virtual network mapping phrase by SP algorithm [16]. In the virtual network runtime phrase, we compare the proposed algorithm with the EMSRN algorithm [17].

4.2 Determine the Value of α

We select the multiple data fitting flow graph as analysis data. The graph is shown in figure 3.

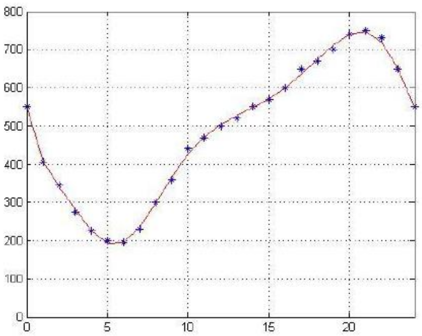

Figure 3. The Flow Graph Attained by Multi-Data Fitting

We get the values from the graph, and show them in Table 1.

Table 1. the Value of α and the Corresponding Values (According to Figure 3)

The value of α	*Dividing point of network-flow*	*Idle time slice*	*The length of idle time slice(minutes)*	*The length of busy time slice(minutes)*
40%	300	2:40-8:00	320	1120
45%	337.5	2:00-8:30	390	1050
50%	375	1:40-9:10	450	990
55%	412.5	0:50-9:40	530	910
60%	450	0:40-10:10	570	870
65%	487.5	0:25-11:40	675	765
70%	525	0:10-13:00	770	670

4.3 Different energy-saving effects

Firstly, we simulate the process ESVNO-DTS, and observe energy-saving effects under different values of α.

The table 1 presents little correlation between energy-saving effects and the value of α. So we use the network-flow graph retained by multi-data fitting to determine the best value of α.

We define the total amount of energy consumption in a day:

$$E_{oneday} = T_1 \times E_1 + T_2 \times E_2 \tag{6}$$

The T1 denotes the length of idle time slice, and the T2 denotes the length of busy time slice. The E1 denotes the value of energy consumption per unit in idle time slice. The E2 denotes the value of energy consumption per unit in busy time slice.

Total energy consumption of a day can be computed by formula (6). Figure 4 shows total energy consumption of one day under different number of mapped virtual networks and different value of the α.

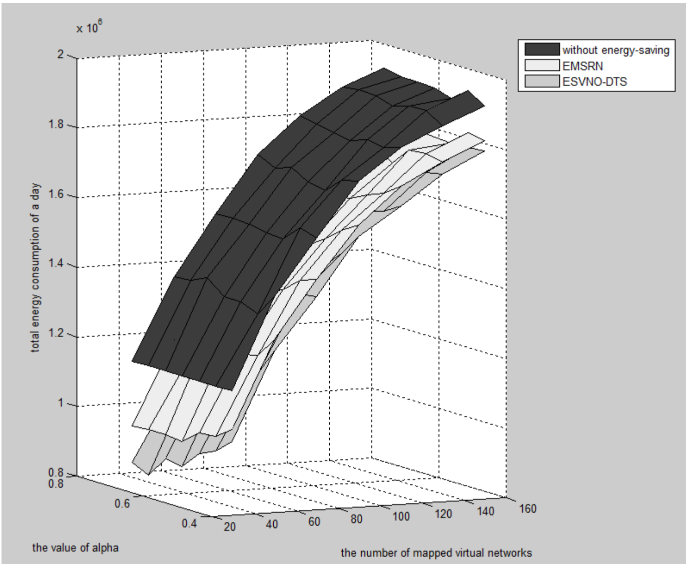

Figure 4. The Front View of Total Energy Consumption of One Day

In order to get the energy-saving effects for each value of α, we define the average amount of energy-saving:

$$E_{oneday}^{ave}(\alpha) = \frac{\sum_{i=1}^{7}\left(E_{sp}^{i}(\alpha) - E_{\mathrm{ESVNO\text{-}DTS}}^{i}(\alpha)\right)}{7} \tag{7}$$

The $E_{\mathrm{ESVNO\text{-}DTS}}^{i}(\alpha)$ denotes the total energy consumption of one day using ESVNO-DTS while the mapped virtual networks ID is i; the $E_{sp}^{i}(\alpha)$ denotes the total energy consumption of a day without energy-saving while the mapped virtual networks ID is i.

We define the average degree of energy-saving:

$$\theta_{oneday}^{ave}(\alpha) = \frac{\sum_{i=1}^{7}\left(E_{sp}^{i}(\alpha) - E_{ESVNO-DTS}^{i}(\alpha)\right) / E_{sp}^{i}(\alpha)}{7} \tag{8}$$

We get table 2 according to formula (7) and formula (8).

Table 2. The Average Amount and Average Degree of Energy-Saving in One Day under Different Values of α

The value of α	α=0.4	α=0.45	α=0.5	α=0.55	α=0.6	α=0.65	α=0.7
Average amount of energy-saving	152459.43	183500.57	203509.29	229471.07	238146	281344.82	299926
Average degree of energy-saving	9.545%	11.542%	12.930%	14.666%	15.115%	17.872%	18.878%

The experiment results show us that the best value of α is 0.7. So, it is set to 0.7 in the subsequent experiments.

4.4 Energy Consumption when α = 0.7

Physical network is composed of three kind of devices which include physical host nodes, physical forwarding nodes and physical links.The number of activated devices and the utilization of host nodes directly determine the energy saving effects. Figures 5, 6, 7, 8 show the energy saving situation in idle time slice when the value of α is 0.7.

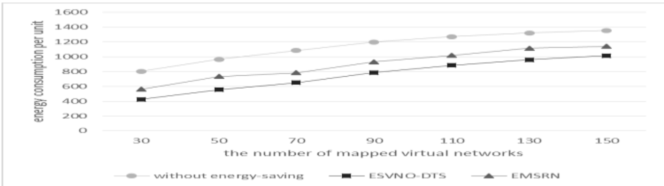

Figure 5. Energy Consumption per Unit

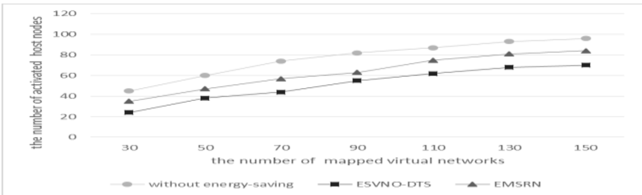

Figure 6. The Number of Activated Physical Host Nodes

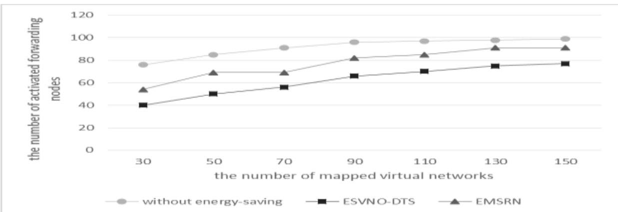

Figure 7. The Number of Activated Physical forwarding Nodes

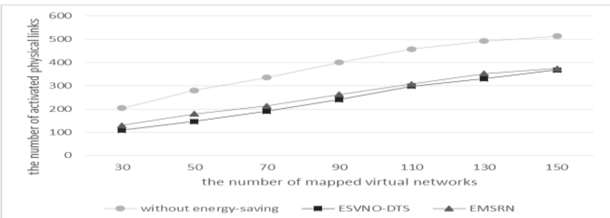

Figure 8. The Number of Activated Physical Links

It can be observed that ESVNO-DTS algorithm performs a significant advantage on the activated physical host nodes and activated physical forwarding nodes, and it also has a clear advantage on the energy consumption per unit.

5. Conclusions

We presented a model for the energy-aware virtual network operation problem, and we designed energy-saving method by divided time slice. Experiments show that the proposed algorithm has good performance on energy-saving problem in the operation of virtual networks.

Acknowledgements

The work was supported in part by the following funding agencies of China: National Natural Science Foundation under Grant 61379079, and the National Basic Research Program (973 Program) under Grant 2012CB315901.

References

[1] W. Fisher, M. Suchara, J. Rexford. Greening backbone networks: Reducing energy consumption by shutting off cables in bundled links [C]. In Proceedings of the 1st ACM SIGCOMM Workshop on Green Networking. **(2010)**August 29-34; New Delhi, India.

[2] C. Lin, Y. Tian, M. Yao. Green network and green evaluation: Mechanism, modeling and evaluation [J]. Chinese Journal of Computers ,34,4**(2011)**: 593-612.

[3] K.J. Ye, Z.H. Wu, X.H. Jiang, Q.M. He. Power management of virtualized cloud computing platform [J]. Chinese Journal of Computers, 35,6**(2012)**: 1262-1285.

[4] B. Chun, G. Iannaccone, G. Iannaccone, R. Katz, G. Lee, L. Niccolini. An energy case for hybrid datacenters [J]. Operating Systems Review(ACM), 44,1**(2010)**: 76-80.

[5] Gartner. http: / /www. gartner. com/it/page \ ? jsp?id=1442113.

[6] Neil Rasmussen. Determining Total Cost of Ownership for Data Center and Network Room Infrastructure: White Paper 6. http: //www. apcmedia. com/salestools/CMRP-5T9PQG/CMRP - 5 T 9P QG_R4_EN. pdf.

[7] J. F. Botero, X. Hesselbach, M. Duelli, D Schlosser, A. Fisher, H. De Meer. Energy efficient virtual network embedding[J]. Communications Letters, 16,5**(2012)**: 756-759.

[8] B. Wang, X.L. Chang, J.Q. Liu, J.K. Muppala. Reducing power consumption in embedding virtual infrastructures[C]. In proceeding of Globecom Workshops, **(2012)** December 714-718; Anaheim, CA, United states.

[9] J. F. Botero, X. Hesselbach. Greener networking in a network virtualization environment[J]. Computer Networks, 57,9**(2013)**: 2021-2039.

[10] S. Su, Z.B. Zhang, X. Cheng, Y.W. Wang, Y. Luo, J. Wang. Energy-aware virtual network embedding through consolidation[C]. In Proceedings of Computer Communications Workshops, **(2012)** March 127-132; Orlando, FL, United states.

[11] S. Su, Z.B. Zhang, A. X. Liu, X. Cheng, Y.W. Wang, X.C. Zhao. Energy-aware virtual network embedding[J]. IEEE/ACM Transactions on Networking, 22,5**(2014)**: 1607-1620.

[12] https://ams-ix.net/technical/statistics

[13] X.H. Chen, C.Z. Li, L.Y. Chen, Z.B. Zeng. Energy efficient virtual network embedding based on actively hibernating substrate nodes and links [J]. Journal of Software,25,7**(2014)**:1416-1431.

[14] L. Chiaraviglio, M. Mellia, F. Neri. Minimizing ISP network energy cost: formulation and solutions[J]. IEEE/ACM Transactions on Networking, 20,2**(2012)**: 463-476.

[15] L. A. Barroso, J. Clidaras, U. Hölzle. The datacenter as a computer: An introduction to the design of warehouse-scale machines[J]. Synthesis lectures on computer architecture, 8,3**(2013)**: 1-154.

[16] M. Yu, Y. Yi, J. Rexford. Rethinking virtual network embedding: substrate support for path splitting and migration[J]. ACM SIGCOMM Computer Communication Review, 38,2**(2008)**: 17-29.

[17] S.Q. Liao, C. M. Wu, B. Wang, M. Jiang. Research on energy-saving method based on reconfigurable network architecture [J]. Journal on Communications, 33,9**(2012)**:77-84.

Authors

Ying Hu. Ying Hu is a Ph.D. Candidate from Zhengzhou University in China. Her major Computer Science. His research interests include network virtualization.

Lei Zhuang. Lei Zhuang received the Ph.D. Degree in Computer Science from The PLA Information Engineering University, China. She is currently a Professor at Zhengzhou University. Her research interests include next generation of Internet.

Yuxiang Hu. Yuxiang Hu received his Ph.D. Degree in Computer Science from The PLA Information Engineering University, China. He is currently a Lecturer at The PLA Information Engineering University. His research interests include next generation of Internet.

Ding Ma. Ding Ma is a Ph.D. Candidate from Zhengzhou University. His major is Computer Science. His research interests include the next generation of Internet and network virtualization.