This is a preview version of the paper: Mohammad Ali Raayatpanah and Thomas Weise. Virtual Network Function Placement for Service Function Chaining with Minimum Energy Consumption. In *Proceedings of the 2018 IEEE International Conference on Computer and Communication Engineering Technology (CCET'18)*, August 18-20, 2018, Beijing, China, IEEE. Copyright is with IEEE.

Virtual Network Function Placement for Service Function Chaining with Minimum Energy Consumption

Mohammad Ali Raayatpanah
Institute of Applied Optimization
Faculty of Computer Science and Technology
Hefei University, Hefei, Anhui, China
Mathematical Sciences and Computer
Kharazmi University, Tehran, Iran
e-mail: raayatpanah@khu.ac.ir

Thomas Weise
Institute of Applied Optimization
Faculty of Computer Science and Technology
Hefei University
Hefei, Anhui, China
e-mail: tweise@hfuu.edu.cn
(corresponding author)

***Abstract*—Network Function Virtualization (NFV) is an emerging technique to improve the performance of enterprise networks. This technique moves packet processing from hardware middleboxes to virtual network functions (VNFs) running on servers. Traffic flows are usually required to pass through a specific sequence of VNFs to satisfy access control policies specified by an enterprise network administrator. In this paper, we consider the problem of routing of traffic flows and determining the number and location of VNFs to be deployed in the network for minimizing the server energy consumption. Additionally, we guarantee that the traffic flows pass all the necessary VNFs in a specified sequence and also meet end-to-end delays and bandwidth consumption constraints. We formulate the problem as an integer liner programing task and solve it with CPLEX. The simulation results show that we can save up to 33% of server energy consumption for typical ISP networks.**

Keywords-network function virtualization; integer linear programming; energy consumption; service function chains

I. Introduction

Network operators deploy middlebox services or network appliances in a network to realize various goals such as enhancing performance, traffic monitoring, traffic engineering, traffic policing, and network security enforcement [1]. Typical examples of middleboxes include Deep Packet Inspection (DPI), Network Address Translation (NAT), Intrusion Detection Systems (IDSs), Intrusion Prevention Systems (IPSs), wide area network (WAN) optimizers, firewalls, proxies, as well as logging, metering, charging, and advanced charging. Moreover, most often traffic flows in a network need to traverse a sequence of middleboxes in a particular order, which is referred to as Service Function Chaining (SFC) [2]. For example, a traffic flow to satisfy access control policies specified by a network administrator may need to pass through a firewall, then an IPS, and finally through a proxy.

While middleboxes offer new functionality to networks, services, and users, concerns have been raised regarding high purchase and operational costs as well as the resulting management complexity. Recently, Network Function Virtualization (NFV) emerged as a promising technology that can provide significant improvements for these concerns [3]. The key advantage of NFV is that packet processing moves from special purpose hardware middleboxes to general-purpose software VNFs. While NFV technology can reduce capital and operational expenditure and provide flexible ways for SFCs, there are several issues that need to be considered in this area. The flexibility in selecting locations to deploy VNFs in comparison with middleboxes leads to VNF placement problem, i.e. where to place VNFs in the network so that the total cost of deploying VNFs is minimized. Finding proper locations to deploy VNFs can simultaneously optimize deployment costs and traffic routing paths, which can significantly reduce the network operational expenditure. There are several research efforts in recent years about the NFV technology. Some of these efforts focus on determining optimal locations for VNFs without considering the routing of traffic flows [4, 5]. Other works address VNF management by considering both placement and traffic flows, but solve them separately [6, 7].

In this paper, we focus on determining the number and location of VNFs to be deployed in the network, finding optimal traffic flow paths according to an SFC, and meeting end-to-end delay and bandwidth consumption limits from user and provider perspectives, while minimizing the server energy consumption. As a key innovative feature, in this work, we investigate an integrated formulation to join the problems of VNF deployment and traffic flow routing by ensuring that the path of each flow starts from an entry switch, meets all the necessary VNFs in the right sequence, and ends at an exit switch. This formulation seeks to minimize server energy consumption and provides quality-of-service and quality-of-transmission guarantees, which include the realization of end-to-end demands, end-to-end delays, bandwidth consumption, physical link bandwidths, physical node capacities, and provides VNF services in the network.

The paper is organized as follows: A description of the mathematical model of the problem is presented in Section II. Section III shows numerical results to demonstrate the validity of the proposed method. Finally, the concluding remarks are given in Section IV.

II. SYSTEM DESCRIPTION

In this section, we describe a formal definition of the problem and provide a mathematical model of the problem.

A. System Model

In our scenario, we know the structure of a physical network, the server resource capacities, and the bandwidth capacity and propagation delay of physical links. The physical network can be modelled as a simple undirected graph $G = (V, L)$, where V denotes the set of vertices, and $L \subseteq V \times V$ denotes the set of links. Two types of vertices are considered in V: the physical servers denoted by N and the network switches denoted by S. Thus, $V = N \cup S$. A server has a set of resources that include CPU, storage, memory, and etc. The set of resources offered by servers be represented by R. Let c_i^r and n_i represent the resource capacity of sever i for each $r \in R$ and the maximum number VNFs acceptable to be deployed on sever i, respectively.

The link $(i, j) \in L$ represents a communication link between a server and a switch, or between a pair of switches. The bandwidth capacity and propagation delay of the physical link (i, j) are denoted by b_{ij} and d_{ij}, respectively. We can deploy the different types of VNFs, including firewall, IDS, IPS, DPI, NAT, proxy, WAN optimizers, etc, in the network. The set of all required VNFs in the network is denoted by Q. The resource requirement of category $r \in R$ to fulfill VNF q is denoted by γ_q^r. The processing delay of VNF q is represented by δ_q which shows the average delay incurred by a packet when passing through VNF q.

We assume that the network operator is receiving user's requests for different kinds of traffic known in advance. Each traffic request is only routed through a single path connecting the two end nodes of a request. We consider the network operator is serving a set K of traffic requests, which may contain just one or multiple traffic requests. The traffic request of type $k \in K$ can be specified by a 4-tuple $\Omega^k = \langle i^k, F^k, e^k, t^k, D^k \rangle$, where $i^k, e^k \in S$ denote the ingress and egress switches for traffic request k, respectively. Let t^k and D^k denote the bandwidth demand and the maximum delay tolerated based on service level agreement for traffic request k, respectively. Let $F^k = (f_1^k, f_2^k, \ldots, f_{\eta^k}^k)$ represents the SFC request for traffic k, i.e., the sequence of VNFs through which the traffic needs to pass. η^k denotes the number of VNFs in the SFC request for traffic k.

B. Mathematical Model

In this work, we wish to deploy VNFs with minimizing server energy consumption so that satisfying the following conditions:

(1) Each VNF should be successfully deployed at one location,

(2) The number of the deployed VNFs in one location should not exceed its threshold value,

(3) Each traffic is realized through a physical route,

(4) Each traffic passes through the proper VNF sequence,

(5) The cumulative bandwidth request on each physical link should not exceed its capacity,

(6) The cumulative resource requirement of each server should not exceed its resource capacity,

(7) Each end-to-end delay route of a traffic should not exceed maximum tolerable delay.

The binary variable x_{iq} is introduced to represent an VNF placement at a server, i.e.,

$$x_{iq} = \begin{cases} 1, If\ NFV\ q\ is\ deployed\ on\ server\ i, \\ 0, \qquad\qquad\qquad Otherwise. \end{cases}$$

Each VNF should be successfully deployed on one server, which can be expressed in the following constraint.

$$\textstyle\sum_{i \in N} x_{iq} = 1, \quad \forall q \in Q. \tag{1}$$

Energy consumption can be reduced by balancing the resource utilization for all active servers. Hence, we need to determine which servers are active. We define variable z_i for server i that is set to 1 if it's selected to host VNFs, 0 otherwise. In addition, the total number of the deployed VNFs on active server i should not exceed its threshold value n_i. The conditions can be described as follows:

$$\textstyle\sum_{q \in Q} x_{iq} \leq z_i n_i, \quad \forall i \in N. \tag{2}$$

We need to make sure that the deployed VNFs on each server do not violate resource capacity constraints for any resource. This constraint can be obtained as follows:

$$\textstyle\sum_{q \in Q} \gamma_q^r x_{iq} \leq c_i^{rs}, \quad \forall\, i \in N, r \in R. \tag{3}$$

We define the parameter o_{ql}^k to indicate the order of VNF q in the SFC request for traffic k as follows:

$$o_{ql}^k = \begin{cases} 1, If\ q\ is\ the\ l^{th}\ NFV\ in\ the\ SFC\ request\ of\ k, \\ 0, \qquad\qquad\qquad\qquad Otherwise. \end{cases}$$

Next, we define the decision variable s_{il}^k to represent the mapping of a server node to each execution step of the SFC request for traffic k.

$$s_{il}^k = \begin{cases} 1, If\ server\ i\ is\ selected\ for\ step\ l\ of\ the\ SFC \\ \qquad request\ for\ traffic\ k, \\ 0, \qquad\qquad\qquad\qquad Otherwise. \end{cases}$$

This variable can be derived from variable x_{iq} and parameter o_{ql}^k as follows:

$$s_{il}^k = x_{iq} o_{ql}^k, \ \forall\, i \in N, q \in Q, k \in K, l \in O^k, \tag{4}$$

where O^k is a range from 1 to the length of the SFC for traffic flow k.

We present the flow conservation constraint that makes sure that the in-flow and out-flow of each vertex in the physical network is equal except at the ingress and egress switches. We therefore define the decision variable y_{lim}^k to represent a used physical link (i, m) in the network to reach to the l^{th} VNF in the SFC request for traffic k, i.e.,

$$y_{lim}^k = \begin{cases} 1, \qquad If\ physical\ link\ (i.m) is\ used\ to\ reach\ to \\ \qquad the\ l^{th}\ NFV\ in\ the\ SFC\ request\ for\ traffic\ k, \\ \qquad 0, \qquad\qquad Otherwise. \end{cases}$$

Then for each order in SFC k, a flow conservation constraint is defined as follows:

$$\textstyle\sum_{\{m|(i.m) \in L} y_{lim}^k - \sum_{\{m|(m.i) \in L} y_{lmi}^k = s_{il-1}^k - s_{il}^k. \ \forall\, i \in V, k \in K, l \in O'^k, \tag{5}$$

where O'^k is a range from 1 to the length of the SFC for traffic flow k plus one. Moreover, $s_{i^k 0}^k = s_{e^k \eta^k + 1}^k = 1$, because nodes i^k and e^k are the ingress and egress switches for the traffic request k, respectively. The right hand side of Constraint (5) is equal to zero except for the source (+1) and destination (-1) switches for each order in the traffic request k. We should make sure that selected routes in Constraints (5) from the ingress switch i^k to the egress switch e^k do not exceed the maximum tolerable delay for traffic k. The end-to-end delay of traffic k consists of two parts. The first part is a sum of the delay of the physical links in the connected path from the ingress node to the egress node, while the second part defines the delay incurred by packet processing on virtual network functions. Hence, the following inequality ensures that the end-to-end delay path requirement for each traffic demand will be met.

$$\sum_{l=1}^{\eta^k}\left(\sum_{(i.m)\in L} y_{lmi}^k\, d_{im} + \sum_{i\in N} s_{il}^k\, \delta_{f_l^k}\right) \le D^k,\ \forall k \in K, \quad (6)$$

The cumulative bandwidth request on each physical link $(i.j) \in L$ should not exceed its capacity, i.e.,

$$\sum_{k\in K}\sum_{l=1}^{\eta^k} y_{lim}^k t^k \le b_{ij},\quad \forall (i,j) \in L. \quad (7)$$

The energy consumption of servers relies on the comprehensive utilization of a CPU, memory, storage systems, the number of active network cards and so on. Among these factors, the CPU is the most important energy consumption factor. The resource utilization of a server usually is described as the CPU utilization of the server [8]. The power consumption of CPU utilization can be considered as a function of the workload. The power consumption of the server can then be modeled based on the CPU utilization [8]. In general, the energy consumed by a server can be expressed as

$$e = e^{Idle} + \alpha(e^{full} - e^{Idle}),$$

where e^{Idle} and e^{full} are the energy consumed by the server when it is idle and fully loaded, respectively. Parameter α is the normalized CPU speed and usually is equal to $\frac{r_c}{r_t}$, where r_c and r_t denote consumed resource and the total, respectively. Parameter e^{idle} is usually around $0.6 \times (e^{full})$, and is rarely lower than $0.5 \times (e^{full})$ in real scenarios [9].

If server $i \in N$ is active, the total energy consumed by server i can be expressed as

$$e_i = e_i^{Idle}\, z_i + \sum_{q\in Q} \alpha_q (e_i^{full} - e_i^{Idle}) x_i,$$

where α_q is the normalized CPU load of VNF q.

By minimizing server energy consumption, we can prevent the deployment of too many VNFs on servers and find the optimal number and placement of VNFs. Then the VNF placement problem with the goal of minimizing the total server energy consumptions is formally formulated as follows:

$$\min \sum_{i\in N} e_i^{Idle} z_i + \sum_{i\in N}\sum_{q\in Q}\sum_{r\in R} \frac{\gamma_q^r}{c_i^r}(e_i^{full} - e_i^{Idle})\, x_{iq}$$

$s.t.$

$$\sum_{i\in N} x_{iq} = 1, \qquad \forall q \in Q,$$

$$\sum_{q\in Q} x_{iq} \le n_i z_i, \qquad \forall i \in N,$$

$$\sum_{q\in Q} \gamma_q^r x_{iq} \le c_i^r, \qquad \forall i \in N, r \in R,$$

$$s_{il}^k = o_{ql}^k x_{iq}, \forall i \in N, q \in Q, k \in K, l = 1, 2, \ldots, \eta^k,$$

$$\sum_{\{m|(i,m)\in L} y_{lim}^k - \sum_{\{m|(m,i)\in L} y_{lmi}^k = s_{il-1}^k - s_{il}^k,$$
$$\forall\, i \in V, k \in K, l = 1, 2, \ldots, \eta^k, \eta^k + 1,$$

$$\sum_{l=1}^{\eta^k}\left(\sum_{(i.,m)\in L} y_{lmi}^k\, d_{im} + \sum_{i\in N} s_{il}^k\, \delta_{f_l^k}\right) \le D^k, \forall\, k \in K,$$

$$\sum_{k\in K}\sum_{l=1}^{\eta^k} y_{lim}^k t^k \le b_{ij}, \quad \forall (i,j) \in L,$$

$$z_i, x_{iq}, s_{il}^k, y_{lij}^k \in \{0,1\}, \quad \forall i \in N, q \in Q, k \in K, l = 1, 2, \ldots, \eta^k, (i,j) \in L.$$

III. Simulation Results

In this section, the computational results of simulations on well-known benchmark problems are presented in order to evaluate the performance of the proposed model.

A. Simulation Settings

Topology Dataset: We assess the proposed model on well-known topologies with different sizes from the SNDLib [10] library, which is standard in our field. We choose the three scenarios Abilene, Atlanta, and Janos. The characterizations of the selected topologies are shown in Table I.

TABLE I. The Characteristics of Topologies

	Abilene	*Atlanta*	*Janos*
# of nodes	11	15	161
# of links	14	22	332
# of demands	132	210	650

Network Traffic Model: For each demand, we assume there are different applications and that each application has a different rate of traffic. We consider multiple demand matrices per network and these matrices are based on detailed measurements of traffic in real IP networks [10].

Server, Middlebox, and VNF Data: We obtained parameters for servers, hardware middleboxes, and VNFs from work [2]. Table II shows the parameters used for servers and middleboxes (physical network functions). Table III shows the processing capacities and CPU required of VNs considered in our evaluation.

TABLE II. Parameters used for servers, VNFs and middleboxes

Server Data [10]		
Physical CPU Cores	***Idle Energy***	***Peak Energy***
16	80.5W	2735W

Server Data [10]		
Physical CPU Cores	***Idle Energy***	***Peak Energy***
Middlebox Data		
Idle Energy	***Peak Energy***	***Processing Capacity***
1100W	1700W	40Gbps

TABLE III. PROCESSING CAPACITIES AND CPU REQUIRED OF VIRTUAL NETWORK FUNCTIONS [11]

Network Function	*CPU Required*	*Processing Capacity*
Firewall	4	900 Mbps
Proxy	4	900 Mbps
Nat	2	900 Mbps
IDS	4	600 Mbps

B. *Evaluation of the Proposed Model*

We evaluate the performance of the proposed model through the following scenarios in the experiments.

BASE denotes the baseline scenario for comparison which minimizes the total bandwidth used, without violation of the link capacity and the chain constraints. In this scenario, both the control and the data plane are controlled through routers. Also, functions are executed using particular middleboxs deployed in fixed locations.

Hardware denotes the hardware scenario which places specific hardware at given positions in the network and decouples data and control plane. We here follow the same approach as work [11], where middleboxes are placed in networks at access points.

NFV denotes the NFV scenario, which places VNF in the network by the model presented in Section II and decouples data and control plane.

For evaluating the traffic over time, we consider the traffic patterns for the topologies over a period of 24 hours.

Figure 1 shows our results for the different networks. As we can see, the *BASE* scenario consumes the maximum energy for all networks. Depending on network topologies and parameters, the energy consumptions in this scenario are on average 42% higher than those of the *NFV* scenario. Such result is expected, as the objective in this scenario is the minimization of the total bandwidth usage, while the energy consumption is not taken into consideration at all.

The *Hardware* scenario can reduce the required energy, but not nearly as much as the *NFV* scenario resulting from our approach. The savings in the *Hardware* scenario are between 23% and 28% in comparison to the *BASE* scenario. This is due to the fact that the *Hardware* scenario respects the constraints given by the service chains and the capacities. But, the energy consumptions of the *Hardware* scenario are on average 33% higher than those obtained using the *NFV* scenario. This is due to the fact that middleboxes in the *Hardware* scenario consumes considerably higher energy than *NFV* and also because the *Hardware* scenario fixes the positions of middleboxes in the network. This limits the flexibility for the choices of paths. The *NFV* scenario provides more reduction in the energy consumption. Because in this scenario, we place VNFs in optimal locations and have a higher chance to find efficient paths for demands.

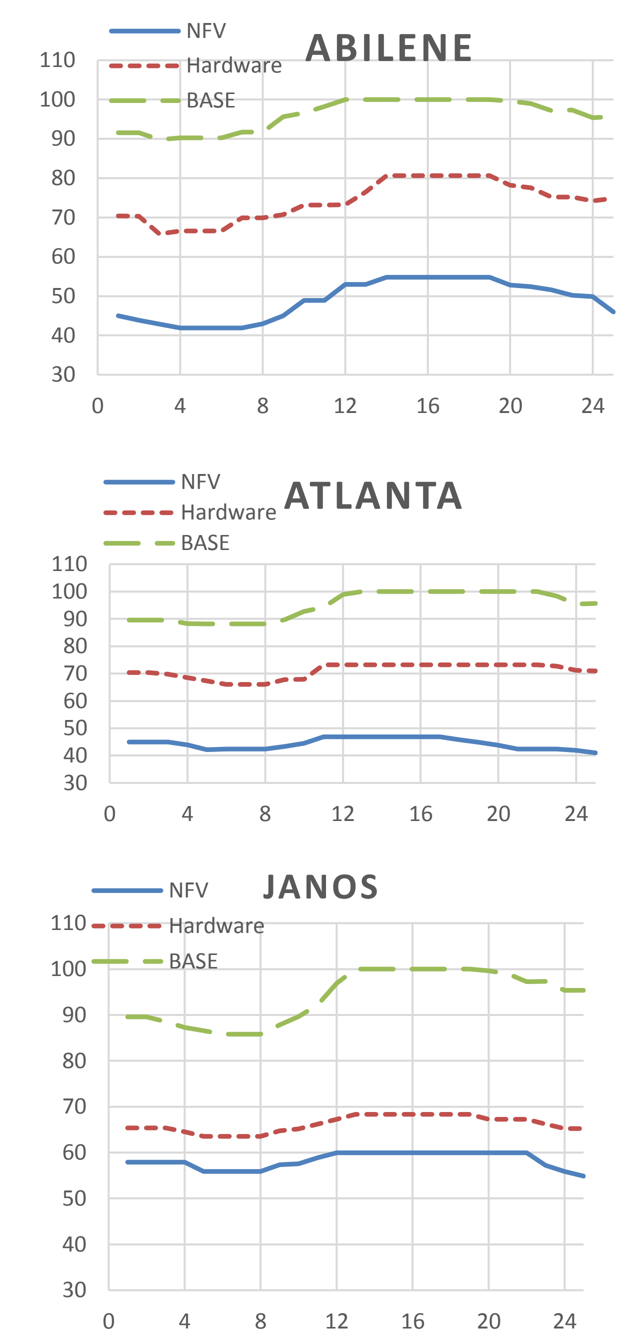

Figure 1. Energy used for three scenarios for Abilene, Atlanta, and Janos networks.

IV. CONCLUSIONS

In this work, we considered the problem of finding the optimal number and location of VNFs with minimum server energy consumptions in which traffic flow steered through a sequence of network functions in a particular order. In this problem, we determined the paths for the demands while meeting end-to-end delay and bandwidth requirements from user and provider perspectives. We formulated this task as an integer linear programming problem.

The tested scenarios on the networks showed that we can save about *one third* of the energy consumption by using network function virtualization. This is an early result, but it is quite significant: Given the fact that computer networks and the internet account for a large fraction of the overall energy consumption in most industrialized countries, such a potential for savings can have major economical impact. VNFs provide a flexible way to deploy and operate network services rather than middleboxes that leads to further the energy savings for different networks. Our idea of optimized automatic VNF placement can help utilizing this potential.

ACKNOWLEDGMENT

Dr. Weise acknowledges support by the National Natural Science Foundation of China under Grants 61673359, 61150110488, and 71520107002.

REFERENCES

[1] J. Sherry, S. Hasan, C. Scott, A. Krishnamurthy, S. Ratnasamy, and V. Sekar, "Making middleboxes someone else's problem: network processing as a cloud service," ACM SIGCOMM Computer Communication Review, 42(4), pp. 13-24, 2012.

[2] Md. F. Bari, S. R. Chowdhury, R. Ahmed, R. Boutaba, and O. C. Duartey, "Orchestrating virtualized network functions," IEEE Transactions on Network and Service Management, DOI 10.1109/TNSM.2016.2569020, 2016.

[3] R. Mijumbi, J. Serrat, J. Gorricho, N. Bouten, F. Turck, and R. Boutaba, "Network function virtualization: State-of-the-art and research challenges," IEEE Communications Surveys & Tutorials, 18(1), pp.236-262, 2015.

[4] S. Lange, S. Gebert, T. Zinner, P. Tran-Gia, D. Hock, M. Jarschel, and M. Hoffmann, "Heuristic approaches to the controller placement problem in large scale sdn networks," IEEE Transactions on Network and Service Management, 12(1), pp. 4-17, 2015.

[5] A. Basta, W. Kellerer, M. Hoffmann, H. Jochen Morper, and K. HoffmannB, "Applying nfv and sdn to lte mobile core gateways, the functions placement problem,". In Proceedings of the 4th workshop on All things cellular: operations, applications, & challenges, pp. 33-38, 2014.

[6] Y. Zhang, N. Beheshti, L. Beliveau, G. Lefebvre, R. Manghirmalani, R. Mishra, R. Patneyt, M. Shirazipour, R. Subrahmaniam, C. Truchan, "Steering: A software-defined networking for inline service chaining," 21st IEEE International Conference on Network Protocols (ICNP), pp. 1-10, 2013.

[7] A. Gember, A. Krishnamurthy, S. St John, R. Grandl, X. Gao, A. Anand, T. Benson, A. Akella, and V. Sekar, "Stratos: A network-aware orchestration layer for middleboxes in the cloud," Technical report, Technical Report, 2013.

[8] Q. Huang, F. Gao, R. Wang, and Z. Qi, "Power consumption of virtual machine live migration in clouds," In IEEE, Third International Conference on Communications and Mobile Computing (CMC), pp. 122-125, 2011.

[9] M. Cardosa, A. Singh, H. Pucha, and A. Chandra, "Exploiting spatio-temporal tradeoffs for energy-aware mapreduce in the cloud," IEEE transactions on computers, 61(12), pp.1737-1751, 2012.

[10] S. Orlowski, M. Piro, A. Tomaszewski, and R. Wessly, "SNDlib 1.0Survivable Network Design Library," Networks, 55(3), pp. 276-286, 2010.

[11] J.Liu, Y. Li, Y. Zhang, L. Su, D. Jin, "Improve Service Chaining Performance with Optimized Middlebox Placement," IEEE Transactions on Services Computing, DOI 10.1109/TSC.2015.2502252, 2015.