Proceedings of CECNet 2021
A.J. Tallón-Ballesteros (Ed.)

doi:10.3233/FAIA210447

Low-Cost Deployment Scheme of VNF and PNF in Optical Datacenter Networks

Jianghua WEI, Xin LI, Jingjie XIN, Ying TANG, Lu ZHANG and Shanguo HUANG[1]
School of Electronic Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China

Abstract. In this paper, we study the problem of network function deployment while user demands are served by both virtualized network functions (VNFs) and physical network functions (PNFs). A hybrid deployment scheme of VNFs and PNFs (HDVP) is proposed. A heuristic algorithm is developed for the HDVP scheme with the aim of minimizing the total cost of VNFs and PNFs. The results show the HDVP scheme achieves lower cost than the conventional only PNFs serving user demands scheme.

Keyword. Virtualized network functions, physical network functions, hybrid deployment

1. Introduction

Virtualized network functions (VNFs) play an important role because of their flexibility and scalability in today's optical datacenter networks (ODNs) [1]. Network function virtualization (NFV) technology decouples network functions (NFs) from proprietary hardware and realizes NFs implemented by software, namely VNFs [2]. VNFs run on standard commercial-off-the-shelf (COTS), while physical network functions (PNFs) run on proprietary hardware. Proprietary hardware has high processing capacity, high operation expense (CAPEX) and high capital expenditures (OPEX), while COTS has low processing capacity, low CAPEX and low OPEX [3, 4]. VNFs are installed into virtual machines (VMs) that provide virtual running environment, increasing additional time consumption, i.e. the booting time and shutdown time of VMs, and the installation time of VNFs. VNFs have low cost but poor time performance, while PNFs have high cost but good time performance [5]. In this paper, we think the cost of VNFs and PNFs depends on the utilization time and prices. Only using VNFs to serve user demands (VNF-only) cannot meet stringent delay requirements, and only using PNFs (PNF-only) is too expensive. The hybrid deployment of VNFs and PNFs is a promising scheme to achieve low cost while meeting stringent delay requirements. So far, many VNF deployment schemes have been proposed to meet delay requirements. A cost-efficient VNF placement and scheduling (VPS) scheme in public cloud networks was proposed which incorporates the impact of several realistic factors, e.g., VNF threading attributes, VM booting time, and VNF installation time [6]. Moreover, machine learning-based

[1] *Corresponding Author: Shanguo Huang, School of Electronic Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China; E-mail: shghuang@bupt.edu.cn*

approaches for predicting VPS decisions were proposed to shorten processing time of applying the deployment decision to networks [7, 8]. These VNF-only deployment schemes are inapplicable to demands with stringent delay requirements because of the poor time performance of VNF. In [9,10], authors proposed a placement scheme in a hybrid scenario, where VNFs and PNFs jointly serve user demands. Nevertheless, they deployed PNFs as priority, increasing cost.

To guarantee low cost when serving user demands, the NFs deployment is not confined to PNF, and any NFs can be mapped into VNFs or PNFs. It is possible to deploy VNFs and PNFs to serve one user demand. In this paper, we propose a novel hybrid deployment of VNF and PNF (HDVP) scheme which uses VNFs and PNFs to jointly serve one user demand to minimize the cost while meeting stringent delay requirements. Heuristic approach is developed for the HDVP scheme. Simulation results show that the HDVP scheme can reduce the cost in ODNs.

2. Hybrid Network Function Placement and Scheduling Scheme

A ODN topology is represented as $G(V, P, E)$, where V denotes the set of VNF-capable nodes, P denotes the set of PNF-capable nodes, and E denotes the set of fiber links. A user demand r is represented as r (a_r, s, b, F, d, t_r), where a_r denotes the arriving time, s and b represent the source node and destination node of demand r, respectively, d is the data size, t_r stands for the delay requirement. $F = <f_1, f_2... f_k>$ indicates NFs of demand r, which run in order, $f_k \in F$. A network function (NF) is represented as f_k $(c, p^v_{f_k}, p^p_{f_k})$, where c denotes the number of required CPU cores by the VNF, $p^v_{f_k}$ and $p^p_{f_k}$ are the processing capacities of the VNF and PNF respectively.

VM additional delay t_{add} is calculated in Eq. (1), where t_b is the booting time, t_i is the VNF installation time and t_s is the shutdown time.

$$t_{add} = t_b + t_i + t_s \tag{1}$$

The total delay of a user demand r, i.e., t, is calculated in Eq. (2), where t_{tran} is the transmission delay, t_{prop} is the propagation delay, and t_{proc} is the processing delay.

$$t = t_{tran} + t_{prop} + t_{add} + t_{proc} \tag{2}$$

In the proposed HDVP scheme, each NF will be mapped into a VNF or a PNF according to the NF mapping principle, i.e., a user demand r with higher delay requirement will have a larger ratio coefficient of PNFs. The basic time represents total processing time while all NFs of a user demand are mapped to VNFs, and it is calculated using Eq. (3), where $p^v_{f_k}$ is the processing capacity of the VNF. The ratio coefficient λ is defined as Eq. (4) and it determines the number of PNFs and VNFs serving user demands.

$$t_{basic} = \sum_{f_k \in F} \frac{d}{p^v_{f_k}} \tag{3}$$

$$\lambda = \frac{t_{basic}}{t_r} \tag{4}$$

Processing capacity gap ΔC represents the difference of PNF processing capacity and VNF processing capacity, which is calculated by Eq. (5). The NF with big ΔC is more likely to be mapped into the PNF. The shortest path is selected to locate VNFs and PNFs to reduce the propagation delay. The VM holding VNFs is reused to reduce the transmission delay. P^{CPU} and P^{PNF} are the prices of the CPU cores and the price of the physical hardware, respectively.

$$\Delta C = p_{f_k}^{p} - p_{f_k}^{v} \tag{5}$$

For each user demand, the cost is calculated by Eq. (6), where t_{proc}^{v} and t_{proc}^{p} are the total processing time of VNFs and PNFs serving the user demand, respectively.

$$\varphi = P^{CPU} \cdot (t_{add} + t_{proc}^{v}) \cdot c + P^{PNF} \cdot t_{proc}^{p} \tag{6}$$

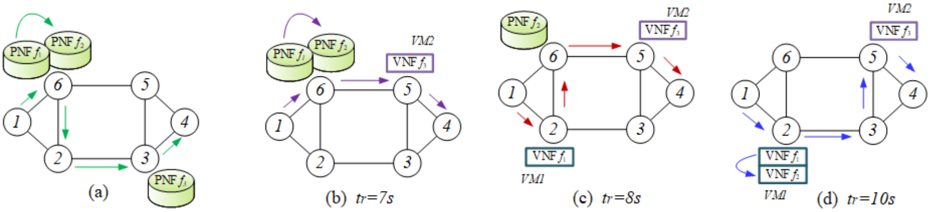

Figure 1. different deployment schemes: (a) PNF-only scheme, (b)-(d) HDVP schemes

Figure 1 illustrates the advantages of the proposed HDVP scheme (Fig. 1(b)-(d)) over PNF-only scheme (Fig. 1(a)). Supposing VNF-capable node *2* holds VM*1* allocated one CPU core, VNF-capable node *3* holds VM*2* allocated two CPU cores, PNF-capable node *3* holds PNFs f_3 and PNF f_1, and node *6* holds PNF f_2, respectively. Supposing the bandwidth between nodes is 10Gbps, and the bandwidth allocated to links of each VM and PNF is 5Gbps. The price P^{CPU} is 30 times as big as P^{PNF}. We consider one user demand r (*0s*, *1*, *4*, $<f_1, f_2, f_3>$, *2GB*, t_r), where f_1 (*1*, *1 Gbps*, *5 Gbps*), f_2 (*1*,*1 Gbps*,*10 Gbps*), and f_3 (*2*, *2 Gbps*, *5 Gbps*). As shown in Fig. 1(a), when r arrives in the network, it incurs the transmission delay 1G/10Gbps=0.1s. After PNF f_1 processes the data of r (processing delay is *1G/5Gbps=0.2s*), the data will be transmitted to PNF f_2 and the processing delay is *1G/2Gbps=0.5s*. The delay calculation for f_2 and f_3 are similar. The total time consumption is 1.8s, and the total cost is 24.

In the proposed HDVP scheme as shown in Fig. 1(b), t_r = 7s. The basic time consumption t_{basic} = 2/1 + 1/2 + 2/2 = 5s. The gaps of f_1, f_2, and f_3 are $\Delta C_{f_1} = 5 - 1 = 4Gbps$, $\Delta C_{f_2} = 10 - 1 = 9Gbps$ and $\Delta C_{f_3} = 5 - 2 = 3Gbps$. So the NF order of PNF mapping precedence is f_2, f_1, and f_3. The ratio coefficient λ= 71.4%, according to the NF mapping principle I in Table 1, the number of PNF is 2. We deploy f_1, and f_2 into PNFs and deploy f_3 into a VNF. The delay calculations for f_1 and f_2 are similar to that of PNF-only scheme. For f_3, both PNF processing delay and VM additional delay are calculated. The time consumption is 4.7s, and total cost is 22.2. We change t_r

to 8s and 10s, and the deployment strategies are shown in Fig. 1(c) and (d), respectively. Table 2 records delay and cost between the HDVP scheme and the PNF-only scheme.

Figure 1(d) shows the results of the HDVP scheme are the same as that of VNF-only deployment scheme when $t_r \geq 8.2s$. If t_r <8.2s, VNF-only cannot meet the delay requirement. This example shows that the HDVP scheme can reduce more cost than that of the PNF-only deployment scheme while meeting stringent delay requirements.

Table 1. NF mapping principles

The proportion of PNF	**λ-I**	λ-II	**λ-III**
0	[0, 55%]	[0, 70%]	[0, 40%]
1/3	[55%, 70%]	[70%, 80%]	[40%, 60%]
2/3	[70%, 85%]	[80%, 90%]	[60%, 80%]
1	[85%, 100%]	[90%, 100%]	[80%, 100%]

Table 2. Result comparisons between two schemes

Scheme	**delay requirement t_r**	**PNF number**	**cost**	**delay**
PNF-only	5	3	24	1.8
HDVP	7	2	22.2	4.7
	8	1	12.3	7.2
	10	0	9.3	8.2

3. Heuristic Algorithm for HDVP

We propose a heuristic approach for the HDVP scheme to minimize the cost of VNFs and PNFs. The HDVP algorithm is detailed as follows. Q indicates the NF set of demand r, which is sorted in increasing order of ΔC, $q \in Q$, Q_v and Q_p indicate the sets of VNFs and PNFs for demand r respectively, β indicates the number of PNFs, $\beta = |Q_p|$, $p_{f_k}^{v}$ and $p_{f_k}^{p}$ indicate the VNF processing capacity and PNF processing capacity respectively, and k indicates the index of NF f_k in F, $1 \leq k \leq |F|$.

Specifically, we first initiate basic time t_{basic} and set Q, Q_v, and Q_p. Next, we determine the NFs of demand r. From line 3 to 6, VNF processing capacity and PNF processing capacity of each NF are obtained, basic processing time t_{basic} is updated, and processing capacity gap is calculated. In line 8, parameter λ is calculated. Then, according to the NF mapping principle, checking λ which level it belongs to, and the number of PNFs β is obtained. In lines 9-10, NFs are added into set Q in the descending order of ΔC. Set Q is divided into two parts. The former is set Q_p, whose size is equal to β and the latter is set Q_v. From line 12 to 19, each NF f_k of set Q_v is mapped into a VNF. If adjacent NFs in the set F are both in set Q_v and require the same number of CPU cores, they are located in a same VM. Else they are located in a same node. If k = 1, or NF f_{k-1} is in set Q_p, a new VM is used for f_k. PNFs are employed to serve set Q_p in line 21. Total data size processed at a certain moment by a PNF should be smaller than its maximum throughput to avoid congestion. In line 23-24, light-paths are calculated between the locations where two consecutive VNF instances are hosted and the total delay is calculated. In line 25-28, we check whether the delay requirement of the user demand is satisfied or not.

Algorithm: Heuristic algorithm for HDVP

Input: demand *r* (a_r, s, b, F, d, t_r) and NF mapping principle.
output: the cost of VNFs and PNFs φ , and the PNF numbers β .

```
Initialize basic time $t_{basic} = 0$, $Q = \phi$ , $Q_v = \phi$ and $Q_p = \phi$ .
Find the set $F$ of demand $r$, i.e. $F = (f_1, f_2, ..., f_{|F|})$
for each NF $f_k \in F$, do
    Get the VNF processing capacity $p^v_{f_k}$ and PNF processing capacity $p^p_{f_k}$ respectively.
    $t_{basic} += d / p^v_{f_k}$
    $\Delta C_{f_k} = p^p_{f_k} - p^v_{f_k}$
end for
Calculate the parameter $\lambda = t_{basic} / t_r$, check the level of $\lambda$ in NF mapping principle and calculate $\beta$ (i.e., the number of PNFs).
Sort f in $\Delta C$ descending order, and put the results into the set $Q = (q_1, q_2, ..., q_{|F|})$ .
$Q_p = (q_1, ..., q_\beta)$ , $\beta = |Q_p|$; $Q_v = Q - Q_p$ .
for each NF $f$ of demand $r$ do
    if NF $f_k \in Q_v$ then
        if the index of NF $k > 1$ and $f_{k-1} \in Q_v$ then
            if the required CPU cores are the same then
                place $f_k$ to the VM holding $f_{k-1}$ .
            else
                place $f_k$ to the same node of $f_{k-1}$ .
        else
            place $f_k$ to the VM with $c$ CPU cores
    else
        place the NF to a PNF, whose throughput is enough to process all data size at the time.
end for
Calculate the light-path between the datacenters hosting $f_k$ .
Calculate the delay $t_{total} = t_{tran} + t_{prop} + t_{add} + t^v_{proc} + t^p_{proc}$ and check whether the latency requirement is satisfied.
if $t_{total} \le t_r$ then
    calculate the cost $\varphi = P^{cpu} \cdot (t_{add} + t^v_{proc}) \cdot c + P^{PNF} \cdot t^p_{proc}$ .
else
    mark it blocked.
```

4. Results and Analysis

We evaluate the HDVP scheme in the US Backbone network topology presented in Fig. 2. According to NF mapping principles (I), (II) and (III) in Table 1, the corresponding experimental results of the HDVP scheme are represented as HDVP (I), (II) and (III). The price ratio of VNFs and PNFs is set to 1/10. Demands with different user nodes are randomly generated. We adjust delay requirement scales *(ddl_s)* to change demands. In this experiment, the blocking rate of VNF-only scheme is greater than 25%, so VNF-only scheme is not considered. PNF-only scheme is deployed as comparison. We compare the cost with different NF mapping principles and different delay requirement scales. Then we focus on the PNF usage.

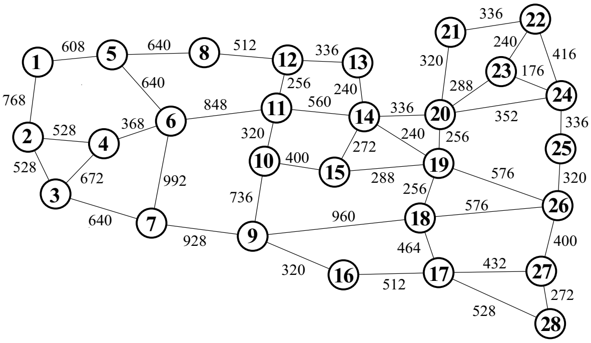

Figure 2. Network topology

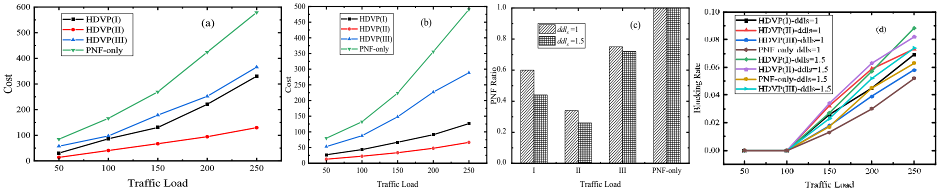

Figure 3. Numerical results (a) cost and PNF ratio (ddl_s =1) (b) cost and PNF ratio (ddl_s =1.5) (c) PNF usage (d) blocking rate

Figure 3(a) and (b) show the cost increases with traffic load for each NF mapping principle at $ddl_s = 1$ and at $ddl_s = 1.5$ respectively. Figure 3(c) shows the PNF usage ratio of different schemes at $ddl_s = 1$ and at $ddl_s = 1.5$ respectively. Figure 3(d) show the blocking rate increases with traffic load at $ddl_s = 1$ and at $ddl_s = 1.5$ respectively. Each HDVP scheme achieves slightly higher blocking rate compared with the PNF-only scheme at the same ddl_s.

At the same traffic load, the cost of each HDVP scheme is smaller than that of PNF-only scheme and the cost of the HDVP scheme (II) is the smallest. For different delay requirements, the cost of each scheme is smaller at $ddl_s = 1.5$ than the cost at $ddl_s = 1$, and the PNF ratio of each HDVP scheme is smaller at $ddl_s = 1.5$ than the PNF ratio at $ddl_s = 1$. For the same delay requirement, the HDVP scheme (II) outperforms the HDVP scheme (I), the HDVP scheme (III) and the PNF-only scheme. The value of ddl_s is bigger in Fig. 3(b) than that in Fig. 3(a), i.e., the delay requirement of the demand is lower in Fig. 3(b) than that in Fig. 3(a). In the same scheme with the same load, the bigger ddl_s, the smaller PNF proportion, and the smaller cost. Compared with the PNF-only scheme at $ddl_s = 1$, the HDVP scheme (I) (II) and (III) reduce PNF ratio above 40%, 66%, and 25%, and reduce cost above 43.0%, 75.1%, and 32.4%, respectively. Compared with the HDVP scheme (I) and (III) at $ddl_s = 1$, the HDVP scheme (II) reduces PNF ratio above 43.3%, and reduces cost above 47.4%.

5. Conclusion

We propose a hybrid deployment of VNFs and PNFs to minimize cost consumption while meeting stringent delay requirements. In this study, we focus on the problem of dynamic deployment of VNFs and PNFs in ODNs where the number of PNFs and VNFs serving a user demand is determined by the ratio of its basic time to its delay requirement. Numerical results indicate that the proposed HDVP scheme can reduce the cost of VNFs and PNFs significantly compared with the PNF-only scheme.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China (Nos. 61701039 and 61821001) and the project was supported by Fund of State Key Laboratory of IPOC (BUPT) (No. IPOC2021ZT15), P. R. China.

References

[1] W. Xia, et al., “A Survey on Data Center Networking (DCN): Infrastructure and Operations,” *IEEE Commun. Surv. Tutorials*, vol. 19, no. 1, pp. 640-656, 2017.

[2] B. Yi, et al., “A comprehensive survey of Network Function Virtualization”, *Computer Netw.*, vol. 133, pp. 212-262, 2018.

[3] H. Ko, et al., “Optimal middlebox function placement in virtualized evolved packet core systems”, in *17th APNOMS 2015*, Busan, Korea(South), pp. 511–514.

[4] D. Bhamare, et al., “Optimal virtual network function placement in multi-cloud service function chaining architecture”, *Computer Commun*, vol. 102, pp. 1–16, 2017.

[5] ETSI, “Network Functions Virtualisation-Introductory White Paper ,” (Issue 1),2012

[6] T. Gao, et al., “Cost-Efficient VNF Placement and Scheduling in Public Cloud Networks,” *TCOM*, vol. 68, no. 8, 4946-4959, Aug. 2020.

[7] S. Lange, et al., “Machine Learning-based Prediction of VNF Deployment Decisions in Dynamic Networks,” in *20th APNOMS 2019*, Matsue, Japan, pp. 1-6.

[8] S. Park, et al., “Machine Learning-based Optimal VNF Deployment,” in *21st APNOMS 2020* , Daegu, Korea (South) , pp. 67-72.

[9] J. Zhang, X. Lu and D. K. Panda, “Performance Characterization of Hypervisor-and Container-Based Virtualization for HPC on SR-IOV Enabled InfiniBand Clusters,” 2016 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW), Chicago, IL, 2016, pp. 1777-1784.

[10] J. Cao, et al., “VNF Placement in Hybrid NFV Environment: Modeling and Genetic Algorithms,” in IEEE 22nd ICPADS 2016 , Wuhan, China, pp. 769-777.