Virtual Network Function Placement with Function Decomposition for Virtual Network Slice

Defang Li, Peilin Hong, Wenzhe Wang, Jianing Pei
The Key Laboratory of Wireless-Optical Communications, Chinese Academy of Sciences, School of Information Science and Technology, University of Science and Technology of China, 230027
ldf911@mail.ustc.edu.cn, plhong@ustc.edu.cn, wzwang@mail.ustc.edu.cn, jianingp@mail.ustc.edu.cn

***Abstract*—Virtual network functions (VNFs) are placed across the network to complete the required network performance of virtual network slice (vNS), and the placement of VNFs is flexible and problematic. Besides, a VNF can be decomposed into multiple sub-functions, and the same type of sub-function can be re-used by multiple VNFs. Therefore, the resources of sub-functions in different vNS requests can be shared and the cost of the substrate network is reduced. So, function decomposition makes the VNF placement problem more flexible and challenging. In this paper, we study the VNF placement problem with function decomposition for vNS, and formulate it as an integer linear programming (ILP) aiming to minimize the total cost of the substrate network. Then, an approach named Placement algorithm based on Function Decomposition (P-FD) is proposed to get a near-optimal solution of this problem. Finally, simulation results show that P-FD can achieve less total cost compared with algorithms in existing.**

***Index Terms*—virtual network function placement, function decomposition, resource sharing, network slicing**

I. Introduction

Recently, the concept of virtual network slice (vNS) has been proposed to facilitate the building of a dedicated and customized logical network with isolated resources [1][2], in which the vNS usually consists of a series of virtual network functions (VNF). In this paper, we study the VNF placement problem considering function decomposition [3] for vNS. When considering function decomposition, a VNF can be decomposed into more fine-grained sub-functions. Furthermore, the sub-functions of the same type placed on the same virtual machine (VM) can be re-used to save the node resource consumptions. Therefore, how to combine these VNFs of various vNSs efficiently becomes a problem to be addressed. In addition, our attentions are also paid to the balance between node and link resource consumption.

Lots of researches have tackled the VNF placement problem, which is usually NP-Hard and therefore finding the optimal solution might not be affordable in large-scale scenarios. So plenty of heuristic approaches are proposed to make a trade off between optimality and algorithm complexity [4]. Li et al. [5] designed a mathematical formulation of the VNF placement problem and proposed a simulated annealing algorithm to solve it. Addis et al. [6] studied the VNF placement and routing optimization while considering the bit-rate variations at each VNF due to specific operations. Mechtri et al. [7] addressed the VNF placement and chaining relying on the eigendecomposition of matrix virtual requested graph and physical infrastructure graph. In [8], the authors studied the problem about how to optimally decompose and embed network services which compose of NFs.

However, there are few researches to address the VNF placement problem while taking advantage of the re-using of decomposed sub-functions. On one hand, most works aim at service decomposition rather than function decomposition, which cannot exploit re-using of the common sub-functions of various VNFs. One the other hand, for the work on function decomposition, a VNF is decomposed into several sub-functions but each sub-function is implemented on a separate function node of the substrate network [8]. In this circumstances, the interfaces and protocols to keep the interconnection and communication of the sub-functions are hard to be defined. In addition, the latency and bandwidth consumption between the sub-functions may affect quality of service.

In this paper, we study how to re-use the sub-functions of different vNS requests, in which the sub-functions of one VNF are located on the same function node. Furthermore, the interconnections between different VNFs are also considered.

The main contributions of this paper are described as follows:

1) We consider the VNF placement with the consideration of the connectivity of vNSs.
2) Function decomposition and sub-function sharing are considered in the VNF placement problem. And the problem is formulated as an ILP model with the aim of minimizing the total cost of the substrate network.
3) A cost efficient heuristic solution called Placement algorithm considering Function Decomposition (P-FD) is designed to solve the VNF placement problem, taking advantage of sub-functions re-using. Finally, we make a detailed simulation to evaluate the performance of P-FD, and the results show that P-FD can achieve lower cost compared with the benchmarks.

The rest of the paper is organized as follows. In Section II, the problem is formulated. Section III presents the proposed algorithm, P-FD. Performance evaluation is described in Section IV. Finally, Section V concludes our work.

II. Problem Formulation

In this section, we formulate the VNF placement problem with consideration of function decomposition as an ILP and

TABLE I
NOTATIONS

Substrate Network	
(F, S, E^S)	The 3-tuple to indicate the substrate network, in which F and S represent the function nodes and switch nodes, respectively. And E^S indicates the set of links.
n_u^s	One node in the substrate network.
e_{uv}^s	Substrate link connecting node n_u^s and n_v^s.
$C_{n_u^s}$	Available node resource of node n_u^s.
$B_{e_{uv}^s}$	Available link resource of link e_{uv}^s.
vNS Request	
Γ	The vNS requests set.
Φ	The VM set.
η	The number of VNFs that a VM can host at most.
(N^γ, E^γ)	Node set and link set of vNS request γ.
$Q_{\gamma,i}^\nu$	The sub-functions set that corresponds to VNF n_i^ν of vNS γ.
n_i^ν	One node of vNS request.
e_{ij}^ν	Logical link connecting node n_i^ν and n_j^ν.
c_{γ,g,n_i^ν}	Requested node resource of sub-function g in VNF n_i^ν of vNS γ.
b_{γ,e_{ij}^ν}	Requested bandwidth resource of e_{ij}^ν in vNS γ..
Function Decomposition	
G, g	G is the set of all sub-functions and g is one of them.
$\lambda_{\gamma,g,n_i^\nu}$	Whether sub-function g is included in VNF n_i^ν.
Variables	
$x_{\gamma,n_i^\nu,\varphi}$	Whether VNF n_i^ν of vNS γ is placed on VM φ. If yes, $x_{\gamma,n_i^\nu,\varphi} = 1$, otherwise, 0.
l_{φ,n_u^s}	Whether VM φ is placed on substrate node n_u^s. If yes, $l_{\varphi,n_u^s} = 1$, otherwise, 0.
$y_{\gamma,e_{ij}^\nu,e_{uv}^s}$	Whether e_{ij}^ν of vNS γ goes through substrate link e_{uv}^s. If yes, $y_{\gamma,e_{ij}^\nu,e_{uv}^s} = 1$, otherwise, 0.

the main notations are listed in Table I.

$$\sum_{n_u^s \in N^S} \sum_{\varphi \in \Phi} x_{\gamma,n_i^\nu,\varphi} * l_{\varphi,n_u^s} = 1, \forall \gamma \in \Gamma,\ n_i^\nu \in N^\gamma \quad (1)$$

Eq. 1 indicates that for a VNF n_i^ν in N^γ, it can be placed on one and only one function node in substrate network.

Initially, the number of VM equals to the number of all VNFs, which is :

$$|\Phi| = \sum_{\gamma \in \Gamma} |N^\gamma| \quad (2)$$

In this paper, we use $|\odot|$ to indicate the number of elements in set $\odot$, for example, $|N^\gamma|$ indicates the number of VNFs in N^γ.

$$\sum_{e_{uv}^s \in E^S} y_{\gamma,e_{ij}^\nu,e_{uv}^s} \geq 0, \forall \gamma \in \Gamma,\ e_{ij}^\nu \in E^\gamma \quad (3)$$

Eq. 3 is the link constraint, and 0 indicates that two nodes in the same vNS request are placed on the same function node.

$$\lambda_{\gamma,g,n_i^\nu} = \begin{cases} 1 & g\ is\ in\ Q_{\gamma,i}^\nu, \\ 0 & g\ is\ not\ in\ Q_{\gamma,i}^\nu. \end{cases} \quad (4)$$

In Eq. 4, $\lambda_{\gamma,g,n_i^\nu}$ is used to represent whether sub-function g is included in VNF n_i^ν of vNS γ. Then the node resource of VM φ is as follows:

$$\begin{aligned} \mathrm{Cost}_{\mathrm{vm}}^\varphi = \sum_{g \in G} \max\{ & c_{\gamma,g,n_i^\nu} * x_{\gamma,n_i^\nu,\varphi} * \lambda_{\gamma,g,n_i^\nu} \\ & |\forall n_i^\gamma \in N^\gamma, \gamma \in \Gamma\ \} \end{aligned} \quad (5)$$

In Eq. 5, $\max\{c_{\gamma,g,n_i^\nu} * x_{\gamma,n_i^\nu,\varphi} * \lambda_{\gamma,g,n_i^\nu} |\forall n_i^\gamma \in N^\gamma, \gamma \in \Gamma\ \}$ indicates that the reserved resource just need to meet the demand of the largest request for the sub-functions of the same type on the same VM, owing to the re-usability.

To ensure the performance of each VNF, the number of VNFs that a VM can support should be limited to a threshold:

$$\sum_{\gamma \in \Gamma} \sum_{n_i^\nu \in N^\gamma} x_{\gamma,n_i^\nu,\varphi} \leq \eta \quad (6)$$

Nextly the node capacity constraint is:

$$\sum_{\varphi \in \Phi} \mathrm{Cost}_{\mathrm{vm}}^\varphi * l_{\varphi,n_u^s} \leq C_{n_u^s}, \forall n_u^s \in N^S \quad (7)$$

The link capacity constraint is represented by Eq. 8:

$$\sum_{\gamma \in \Gamma} \sum_{e_{ij}^\nu \in E^\gamma} b_{\gamma,e_{ij}^\nu} * y_{\gamma,e_{ij}^\nu,e_{uv}^s} \leq B_{e_{uv}^s}, \forall e_{uv}^s \in E^S \quad (8)$$

$$\begin{aligned} & \sum_{n_v^s \in F \cup S} y_{\gamma,e_{ij}^\gamma,e_{uv}^s} - \sum_{n_v^s \in F \cup S} y_{\gamma,e_{ij}^\gamma,e_{vu}^s} = \\ & (x_{\gamma,n_i^\nu,\varphi} - x_{\gamma,n_j^\nu,\varphi}) * l_{\varphi,n_u^s}, \forall n_u^s \in F \cup S \end{aligned} \quad (9)$$

The flow related constraint is ensured by Eq. 9. The flow of the request is consecutive and cannot be split.

Finally, the optimization target is:

Objective function:

$$\begin{aligned} & \min\ (\mathbb{N} + \mathbb{L}). \\ & = \min\ (\sum_{n_u^s \in N^S} \mathrm{Cost}_{n_u^s} + \sum_{e_{uv}^s \in E^S} \mathrm{Cost}_{e_{uv}^s}) \end{aligned} \quad (10)$$

$$\mathrm{Cost}_{n_u^s} = \sum_{\varphi \in \Phi} \mathrm{Cost}_{\mathrm{vm}}^\varphi * l_{\varphi,n_u^s} * \rho \quad (11)$$

$$\mathrm{Cost}_{e_{uv}^s} = \sum_{\gamma \in \Gamma} \sum_{e_{ij}^\gamma \in E^\gamma} b_{e_{ij}^\gamma} * y_{e_{ij}^\gamma,e_{uv}^s} * \varrho \quad (12)$$

The objective function aims to minimize the total cost of the substrate network. ρ and ϱ represent the node cost and link cost per unit, respectively. $\mathbb{N}$ indicates the node resource cost and $\mathbb{L}$ indicates the link resource cost.

III. PROPOSED SOLUTION

In this section, we present P-FD to solve the problem, aiming to minimize the total cost of the substrate network.

A. P-FD

In large-scale network, finding the optimal solution for the ILP model of our problem is unaffordable. So we design a heuristic solution to solve it, named Placement algorithm considering Function Decomposition, i.e. P-FD, which tries to balance the node resource consumption and bandwidth resource consumption efficiently to get the near optimal solution. P-FD firstly calculates the embedding order of the vNS requests, then the vNS requests are embedded in the network in sequence.

At first, a matrix M is built to record the saved resources of different combination of vNS requests. $M_{(p,q)}$ in the matrix represents the metric of saved resource between vNS request γ_p and vNS request γ_q. The metric is calculated by summing up the saved resources between VNFs in γ_p and VNFs in γ_q.

Then the vNS pair that has larger value in matrix M are embedded preferentially, and the vNS request that demands more resource is embedded in the network firstly for the same pair of vNSs. The embedding order can be obtained by applying Prim search algorithm [9] to matrix M. According to this embedding order, the vNS request combination that can save more resources are put together as much as possible.

Next, for the VNFs in one vNS, they are sorted in descending order based on their requested resources, in which the requested resources include node resource and bandwidth resource linked to the VNF. For the function nodes, their residual resources are calculated in similar way, and are sorted in ascending order based their residual resources.

Then the VNF that demands the most resource is placed on the function node that has the maximum available resource. For each of the rest unplaced VNFs, the candidate function node that has minimum available resource but can host the VNF is chosen as the location, which aims to utilize the substrate network resource adequately.

After the placement of all vNSs, the VNFs should be mapped to one VM in each function node. In the process, the VNFs that have most common sub-functions are put together together, while the number of VNFs on one VM should not exceed the threshold. The node resource consumption cost is fixed, but the link resource consumption cost is related to the physical paths that connect the VNFs. To save link resources, we use Dijkstra algorithm to get the shortest path among two selected function nodes.

In summary, P-FD firstly gets the embedding order of the vNS requests to re-use the resources of different vNS request combinations as much as possible, so more node resources are saved and the total costs are reduced. Then, when placing the VNFs of each vNS request, we strike a balance between resource saving and load balancing. Therefore, P-FD can save the cost of substrate network effectively.

B. Complexity

The complexity of our solution is at the level of $\mathbf{O}(|\Gamma|^2(|F|+|S|)log(|F|+|S|))$.

IV. Performance Evaluation

In this section, we describe our simulation environment firstly, then compare the performance of P-FD with other algorithms. Through the results, P-FD has been proved to be effective to solve the VNF placement problem with function decomposition.

A. Simulation Settings

We use BRITE [10] to generate the substrate network with Waxman model [11]. Simulation configurations are shown in Table II.

TABLE II
SIMULATION SETTINGS

Substrate Network	
Number of nodes	60
Number of function nodes	27
Available CPU of each node	1000 MIPS
Available bandwidth of each link	10 Gbps
CPU cost per unit (ρ)	1/MIPS
Bandwidth cost per unit (ϱ)	0.1/Mbps
VNS Request	
Number of requests	100
Number of nodes in a vNS	uniform distribution of (5,11)
Number of sub-functions in a vNS	uniform distribution of (1,3)
Type of VNFs	10
Type of sub-functions	10
Requested CPU of each sub-function	uniform distribution of (1,20) MIPS
Requested bandwidth of each logical link	uniform distribution of (1,500) Mbps
η	3

In the evaluation, P-FD is compared with existing algorithms: Greedy algorithm and Eig-Dec algorithm [7] under the same simulation environment. In Greedy algorithm, the VNFs are sorted in descending order firstly based on their requested resources, and then they are placed on the function node in sequence. In the process, the function node that has the least available resource but can host the VNF is chosen with highest priority. The EigDec algorithm firstly establishes an adjacency matrix for a vNS and the substrate network. Then, VNF instances are selected based on Umeyamas eigendecomposition approach.

We perform 10 groups of experiments for each result.

B. Evaluation Results

1) Comparison of different kinds of costs: Fig. 1 shows the CPU cost and bandwidth cost of three algorithms. From the figure, we can see that P-FD always gets the lowest cost compared with the other two algorithms. With Greedy, the sub-functions' re-using between the vNS requests is not considered. In each vNS request, although the total cost is minimized in the process of placing each VNF, the result of the overall situation may not be well. As for Eig-Dec, the widest-shortest path routing algorithm results in the longest path, which leads to the biggest resource consumption and the worst performance in the simulation. P-FD arranges vNS

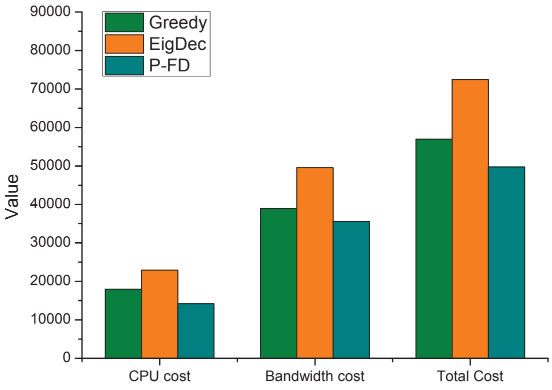

Fig. 1. Performance comparisons versus different kinds of costs

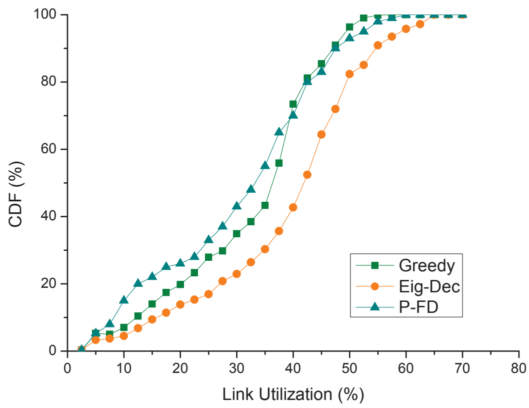

Fig. 2. Performance comparisons versus link utilization

requests in descending order considering the sub-functions re-using, and achieves good balance between CPU resource utilization and bandwidth resource utilization.

2) Comparison of link utilization: Fig. 2 shows Cumulative Distribution Function (CDF) of link utilization of three algorithms. We can see that Eig-Dec consumes the most bandwidth resource, which is corresponding to the maximum bandwidth cost in Fig. 1. The reason is that Eig-Dec uses the widest-shortest path routing algorithm to derive the pathes between the VNFs, which usually results in the longest path. P-FD shows the best performance, because shortest path algorithm tends to result in the least bandwidth resource consumption.

3) Comparison with optimal solution: In addition, We use Gurobi [12] to get the optimal solution in a smaller network. There are 20 nodes of substrate network and 3 vNS requests in this scenario. Fig. 3 presents the total cost of different approaches. We can see that P-FD can achieve near-optimal performance.

V. Conclusion

In this paper, we study the VNF placement problem for vNS with function decomposition. And the total cost of substrate network can be saved by sub-functions re-using when function decomposition is supported. Then the problem is modeled

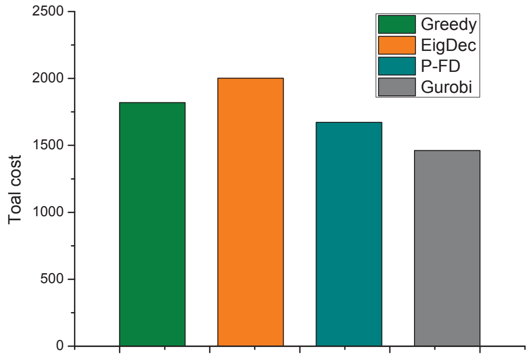

Fig. 3. Performance comparisons with optimal solution

as an ILP, and an efficient heuristic solution called P-FD is designed to solve the problem. Simulation results show that P-FD outperforms the existing algorithms, and consumes less resources when settling down the same set of vNS requests.

Acknowledgment

This work was supported by the National Natural Science Foundation of China under Grant No. 61671420 and No. 61672106.

References

[1] N. Alliance, "NGMN 5G white paper," *Next Generation Mobile Networks Ltd, Frankfurt am Main*, 2015.

[2] M. Richart, J. Baliosian, J. Serrat, and J.-L. Gorricho, "Resource slicing in virtual wireless networks: A survey," *IEEE Transactions on Network and Service Management*, vol. 13, no. 3, pp. 462–476, 2016.

[3] M. R. Sama, X. An, Q. Wei, and S. Beker, "Reshaping the mobile core network via function decomposition and network slicing for the 5g era," in *Wireless Communications and Networking Conference Workshops (WCNCW), 2016 IEEE.* IEEE, 2016, pp. 90–96.

[4] F. Bari, S. R. Chowdhury, R. Ahmed, R. Boutaba, and O. C. M. B. Duarte, "Orchestrating virtualized network functions," *IEEE Transactions on Network and Service Management*, vol. 13, no. 4, pp. 725–739, 2016.

[5] X. Li and C. Qian, "The virtual network function placement problem," in *IEEE Computer Communications Workshops (INFOCOM WKSHPS), Conference on*, 2015, pp. 69–70.

[6] B. Addis, D. Belabed, M. Bouet, and S. Secci, "Virtual network functions placement and routing optimization," in *IEEE Cloud Networking (CloudNet), 4th International Conference on*, 2015, pp. 171–177.

[7] M. Mechtri, C. Ghribi, and D. Zeghlache, "A scalable algorithm for the placement of service function chains," *IEEE Transactions on Network and Service Management*, vol. 13, no. 3, pp. 533–546, 2016.

[8] S. Sahhaf, W. Tavernier, M. Rost, S. Schmid, D. Colle, M. Pickavet, and P. Demeester, "Network service chaining with optimized network function embedding supporting service decompositions," *Computer Networks*, vol. 93, pp. 492–505, 2015.

[9] J. C. Gower and G. J. Ross, "Minimum spanning trees and single linkage cluster analysis," *Applied statistics*, pp. 54–64, 1969.

[10] A. Medina, A. Lakhina, I. Matta, and J. Byers, "Brite: An approach to universal topology generation," in *IEEE Modeling, Analysis and Simulation of Computer and Telecommunication Systems, Proceedings. Ninth International Symposium on*, 2001, pp. 346–353.

[11] B. M. Waxman, "Routing of multipoint connections," *IEEE journal on selected areas in communications*, vol. 6, no. 9, pp. 1617–1622, 1988.

[12] *Gurobi optimizer reference manual, version 7.0*, Gurobi, [Online]. Available: http://www.gurobi.com/documentation/7.0/refman.pdf.