A Markov Approximation Based Approach for VNF Scheduling

Chuan Pham, Minh N.H Nguyen and Choong Seon Hong*

Department of Computer Science and Engineering, Kyung Hee University

Abstract

The competition from the top of service providers, such as Google, Microsoft, has placed telecom service providers to high pressure to provide cost-effective in deploy the system. Virtualization technologies becomes the most notable features, especially, Network Function Virtualization (NFV), Virtualizing all network functions (called virtual network functions, VNFs) as software to run on resource pool of cloud/datacenters, NFV can hide the complexity of lower deployment and orchestration, however, it arises an efficient resource allocation mechanism requirement. In this paper, we propose an algorithm for scheduling allocate processing/network resources to execute service chains. Our proposed algorithm can find the close-optimal solution of the NP-hard VNF scheduling, further, it improves 28.98% the scheduling time, compared to current methods.

1. INTRODUCTION

Network Function Virtualization (NFV) [1], [2] has been considered as the heart of service providers with profound impacts on technology planning, network engineering, operations and procurement. And despite the enormous upheaval that NFV brings the service providers to their business underlying virtually every network functions, it is still be in its infancy state. Specifically, the approaches for resource allocation appear as the main issue in NFV.

There are currently different trends pushing for resource allocation in NFV, however, in this paper, we focus on network function scheduling, one of critical problems is to allocate processing/network resources for executing service chains. Using virtualization, network functions can be deployed as virtual machines on physical nodes (called virtual network function, VNF). In this context, multiple VNFs share the resource nodes, and also to transmit data, they need to share physical links. An effective sharing schedule can improve significant the performance of the network system. There exist many method that are used to share processing/network resources, such as first come first serve, round-robin [3]. However, due to the diverse of VNF instances and the service chain requirements, it is difficult to make a schedule to execute VNFs in the system optimally. Currently, round-robin approach is valuated as a standard method, which is simple and not expensive in implementation, despite without guarantee in a good performance. Finding an optimal scheduling to execute each virtual network functions in terms of minimizing the total execution time without degrading the service performance.

As shown in [3] given the physical network resource (e.g., CPU, memory, network bandwidth), and a placement scheme that allocates VNFs on physical nodes and embeds virtual links on physical links, we need to find the order of execution that can complete all given service chains in the shortest time. The Fig.1 shows a feasible schedule in case of executing all VNFs within 9 timeslots.

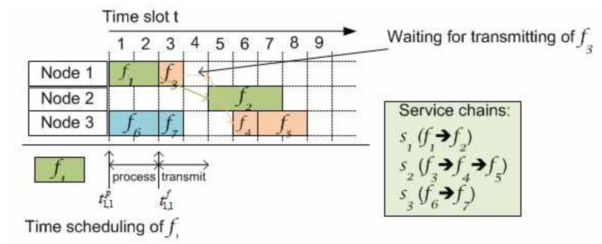

Fig. 1 An example of VNF scheduling.

The VNF scheduling is a combinatorial optimization problem, which cannot be solved in polynomial time [3]. This challenge prevents to find an optimal solution. Further, the network operator also does not have much time (i.e., 10ms to make a schedule) to run complicated algorithm to solve. Therefore, existing heuristic algorithms are applied to address this problem, even though the optimality gap is still high.

In this paper, we advocate the Markov approximation framework [5] that can achieve a close-optimal solution in VNF scheduling. Markov approximation approach appears in recent years as a promising solution for the set of combinatorial network optimization problem since it owns i) a simple implementation without requiring complicated and powerful solver, ii) a flexible framework that can be applied to multiple approaches, iii) enabling both centralized and distributed manners.

The rest of the paper is organized as follows. The VNF scheduling is modeled and formulated in Section 2. The proposed algorithm is presented in Section 3. The simulation results are reported in Section 4 and the conclusion of the paper is given in Section 5.

2. SYSTEM MODEL

In this model, we consider a set S of network service chains, and for each network service chain $i \in S$ exists a set of F_i of VNFs. The VNFs of a service chain i have to be run in a specific order. We denote f_{ij} $(i \in S, j \in F_i)$ to represent the j^th VNF of the service chain i. Given the computation resource of nodes and VNFs, we define N is the set of physical nodes that are hosting all VNFs. We use x_{ijn}^{p} to indicate VNF j of service chain i is placed on node n (x_{ijn}^{p} $= 1$ otherwise x_{ijn}^{p} $= 0$). Similarly, we denote x_{ijk}^{f} $= 1$ to indicate the virtual link j of service chain i placed on physical link k, otherwise this parameter is set 0, where physical link k belongs to the set K of physical links.

We next formulate the VNF scheduling underlying processing constraints, network flow constraints and the objective function. Due to guarantee the order execution of a service chain i, the starting process time t_{ij}^{p} of the preceding VNF $(j+1)$ cannot be earlier than VNF j [3]. We denote p_{ij} as the processing time of j. Constraints to ensure the order of processing can be formulated as follows:

$$x_{ijn}^{p}\ (t_{ij}^{p} + p_{ij}) \le x_{ij+1n}^{p} t_{ij+1}^{p}, \quad \text{The order of processing} \quad (1)$$

$$x_{ijn}^{p}\ (t_{ij}^{p} + p_{ij}) \le x_{ijn}^{p}\ \bar{p}_{ij}, \quad \text{Hard deadline to process } j \quad (2)$$

Constraint (1) is to guarantee the order process of j and $j+1$ in service chain i, (2) is to prevent the hard-deadline execution $\bar{p}_{ij}$ when making a scheduling for j. Next, we formulate the constraints for processing network flows. Similarly, we define t_{ij}^{f} as the starting time to transmit data of VNF j and s_{ij} as the duration of sending data of VNF j. We denote as follows:

$$x_{ijk}^{f}\ (t_{ij}^{p} + p_{ij}) \le x_{ijk}^{f}\ t_{ij+1}^{f}, \quad \text{Packet needs to be processed before forwarding to the next VNF} \quad (3)$$

$$x_{ijk}^{f}\ (t_{ij}^{f} + s_{ij}) \le x_{ijk}^{f}\ t_{ij+1}^{p}, \quad \text{Packet needs to be sent completely before processing at the next VNF} \quad (4)$$

$$x_{ijk}^{f}\ (t_{ij}^{f} + s_{ij}) \le \bar{s}_{ij}, \quad \text{Hard deadline to send packet of VNF } j \quad (5)$$

Constraint (3) is to ensure that the traffic of the VNF j is only forwarded after being processed completely. (4) is to ensure that the next VNF starts to process only when the preceding VNF is forwarded completely. (8) is to ensure the deadline of transmit data of VNF j.

Considering the last VNF in the schedule that has the latest completion time, we then define the objective function for the VNF schedule is as follows:

$$Min\ \{\max_{i\in S, j\in F_i, n\in N} x_{ijn}^{p}\ (t_{ij}^{p} + p_{ij})\}, \quad \text{Minimize the execution time VNFs}$$

Consequently, the VNF scheduling problem can be formulated briefly as follows:

$$\textbf{VNFS}: Min\ \{\max_{i\in S, j\in F_i} (t_{ij}^{p} + p_{ij})\},$$

$$s.t. (1) - (5).$$

VNFS is the combinatorial optimization problem, which cannot be solve in polynomial time. Among several solution for this set of optimization problem, we advocate the Markov approximation framework to solve VNFS.

3. MARKOV ARPPOXIMATION BASED ALGORITHM

3.1. Log-sum approximation

Let $f \in F$ be a feasible solution to VNFS, where F is the set of all feasible solutions. We denote C_f as the objective function value of VNF corresponding to the solution f and p_f as the percentage of time that f should be in use. We adapt the approximation version of P_{SCP} using *log-sum-exp* approximation as follows:

$$\boldsymbol{P_{VNFS-\delta}}: \min_{p_f} \sum_{f\in F} p_f C_f + \frac{1}{\delta} \sum_{f\in F} p_f \log p_f, \quad (3)$$

$$s.t. \sum_{f\in F} p_f = 1,$$

where δ is a positive constant that controls the accuracy of the approximation. $\boldsymbol{P_{VNFS-\delta}}$ is a convex problem that can be solved to derive p_f^{*} [5].

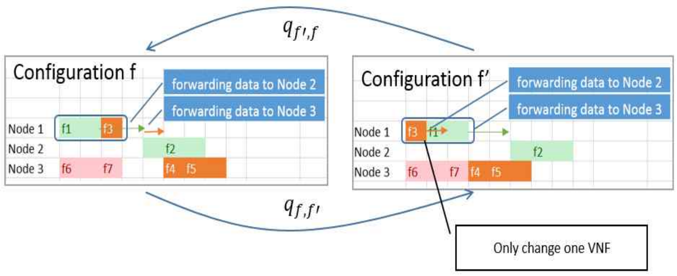

Fig 2: Transition rate between two states

3.2. Algorithm Design

Following the theoretical insights from [5], we design the transition rate between two states f and f' to ensure that in the Markov chain: (i) any two configurations (states) are reachable from each other (irreducible); and (ii) the balance condition is satisfied. For two states f and f with direct transitions, we design the transition rate between two states as

$$q_{f,f'} = \exp(\alpha) \frac{\exp\left(\delta\left(C_f - C_{f'}\right)\right)}{1+\exp\left(\delta\left(C_{f'} - C_f\right)\right)} \quad (4)$$

$$q_{f',f} = \exp(\alpha) \frac{\exp\left(\delta\left(C_f - C_{f'}\right)\right)}{1+\exp\left(\delta\left(C_f - C_{f'}\right)\right)} \quad (5)$$

where α is a positive constant.

We propose a distributed algorithm to embed service chains. The algorithm is executed separately for each node. At one time, only one node can change its configuration, others have to be silent at that time. We schedule for all nodes and the next node has to wait the free message to execute its procedure.

Algorithm 1 Distributed algorithm for VNF scheduling

Initialization: Set parameter α, δ;
for $n \in N$ **do**
    Randomly make an order to execute in node n;
end for
Compute the total timeslots for all VNFs;
repeat
    Randomly change the order of one VNF;
    Compute the total timeslots for scheduling all VNFs;
    Using (4) and (5) to probably select the better schedule;
until Meet the convergence criteria;

4. SIMULATION AND NUMERICAL RESULTS

In this part, we represent the simulation result of our algorithm, based on Markov approximation with 20 nodes and 100 VNFs of 45 service chains.

We compare the performance of our approach with the round-robin method. Fig. 4 shows the convergence of our algorithm around 2000 iterations.

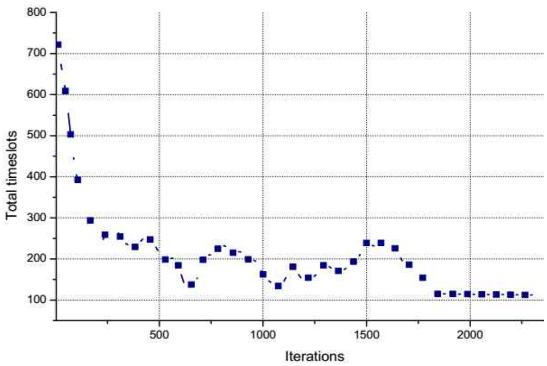

Fig. 4 The convergence of VNFS.

Furthermore, we compare VNFS to round-robin. The simulation result shown in Fig. 5 illustrate the improvement of our proposed approach. In average, VNFS can reduce by 38.89% the total scheduling time.

5. CONCLUSION

In this paper, we consider the VNF scheduling problem, which cannot solve in polynomial time. We formulate the problem and propose the Markov approximation based algorithm, VNFS to find a close-optimal solution. In particular, to find the solution, we design an algorithm based on the designed Markov chain, which can achieve the optimal solution at the stationary distribution. We conduct our algorithm based on the simulation results and compare to round-robin approach. The result shows that our approach can improve significantly in total scheduling time. The analysis and simulation show that our proposed system is adaptable and can be applied in NFV architectures.

ACKNOWLEGMENT

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIP) (NRF-2017R1A2A2A05000995). *Dr. CS.Hong is the corresponding author.

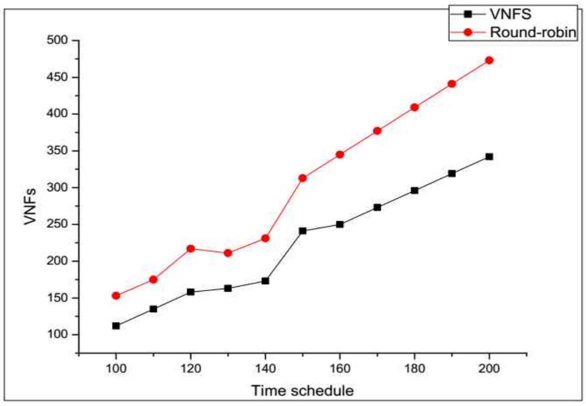

Fig. 5 Comparison between VNFS and round-robin.

REFERENCES

[1] NFV. [Online]. Available: https://portal.etsi.org/nfv/nfv white paper.pdf

[2] C. Pham, N. H. Tran, S. Ren, W. Saad, and C. S. Hong, “Traffic-aware and energy-efficient vnf placement for service chaining: Joint sampling and matching approach,” IEEE Transactions on Services Computing, vol. PP, no. 99, pp. 1–1, 2017.

[3] J. F. Riera, E. Escalona, J. Batall, E. Grasa, and J. A. Garca-Espn, “Virtual network function scheduling: Concept and challenges,” (SaCoNeT), 2014 International Conference on, June 2014, pp. 1–5.

[4] L. Qu, C. Assi, and K. Shaban, “Delay-aware scheduling and resource optimization with network function virtualization,” IEEE Transactions on Communications, vol. PP, no. 99, pp. 1–1, 2016.

[5] Chen, Minghua, et al. "Markov approximation for combinatorial network optimization." IEEE Transactions on Information Theory 59.10 (2013): 6301-6327.