

Virtual Resources Allocation Scheme in ICT Converged Networks

Hyuncheol Kim^(✉)

Department of Computer Science, Namseoul University, Cheonan, Korea
hckim@nsu.ac.kr

Abstract. NFV enhancing the infrastructure agility, thus network operators and service providers are able to program their own network functions (e.g., gateways, routers, load balancers) on vendor-independent hardware substrate. One of the main challenges for the deployment of NFV is the efficient resource (e.g. virtual network function (VNF)) allocation of demanded network services in NFV-based network infrastructures. However, the effective mapping and scheduling of VNFs are essential to successfully provide NFV services. In this paper, we proposed revised online (dynamic) virtual network function allocation scheme to cope with successive network service (NS) requests. Unlike previous research on resource allocation, we assumed that each virtual node processes one or more functions at a time using multiprocessing technologies as in the real environment.

Keywords: Resource allocation · Virtual network · Network function virtualization

1 Introduction

With the advent of cloud and virtualization technologies and the integration of various computer communication technologies, today's computing environments can provide virtualized high-quality services. The network traffic has also continuously increased with remarkable growth. With such a huge trend, due to the flexibility and significant economic potential of these technologies, software defined networking (SDN) and network functions virtualization (NFV) are emerging as the most critical key enablers [1, 2]. As shown in Fig. 1, NFV enhancing the infrastructure agility, thus network operators and service providers are able to program their own network functions (e.g., gateways, routers, load balancers) on vendor-independent hardware substrate. They facilitating the design, delivery and operation of network services in a dynamic and scalable manner. NFV allows for the decoupling of physical network equipment from the services or functions that run on them, such that a given service can be decomposed into a set of virtual network functions (VNFs), which could then be implemented in software that can run on one or more industry standard computing nodes [3–5].

Funding for this paper was provided by Namseoul university.

© Springer Nature Singapore Pte Ltd. 2017

K. Kim and N. Joukov (eds.), *Information Science and Applications 2017*,

Lecture Notes in Electrical Engineering 424, DOI 10.1007/978-981-10-4154-9_87

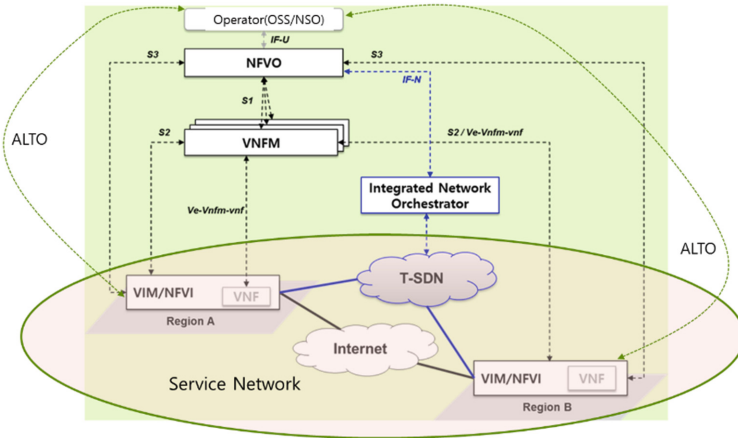


Fig. 1. Typical NFV network components and architecture

One of the main challenges for the deployment of NFV is the efficient resource (e.g. virtual network function (VNF)) allocation of demanded network services in NFV-based network infrastructures. However, the effective mapping and scheduling of VNFs are essential to successfully provide NFV services. In this paper, we proposed revised online (dynamic) virtual network function allocation scheme to cope with successive network service (NS) requests. Unlike previous research on resource allocation, we assumed that each virtual node processes one or more functions at a time using multiprocessing technologies as in the real environment [6–8].

The composition of this paper is as follows. We first highlight some mapping and scheduling challenges of VNFs in the NFV in Sect. 2. The proposed coordinated mapping and scheduling scheme of VNFs are presented in Sect. 3. Finally, the paper concludes in Sect. 4.

2 Related Works

Resource allocation in NFV requires efficient algorithms to determine on which high volume servers (HVSs) VNFs are placed, and be able to migrate functions from one server to another for such objectives as load balancing, reduction of CAPEX and OPEX, energy saving, recovery from failures, etc. [9].

In the NFV architecture framework the component that performs the resource allocation is the orchestrator. The orchestrator manages VNFs through the VNF manager (VNFM) and the virtualized infrastructure manager (VIM). The orchestrator evaluates all the conditions to perform the assignment of VNFs chains on the physical resources, leaning on the VNF managers and the virtualized infrastructure managers. The resource allocation in NFV has carried out in three stages: (1) VNFs Chain composition (VNFs-CC), also known in the literature as Service Function Chaining [1, 10] (2) VNF Forwarding Graph Embedding (VNF-FGE)2 and (3) VNFs Scheduling (VNFs-SCH). Next section deeply details the NFV-RA problem and its derived sub-stages.

Considering that NFV is still seen as a concept under investigation, little research has been conducted on VNFs scheduling. Riera et al. [4] provided the first formalization of the scheduling problem in NFV as a Resource Constrained Project Scheduling Problem. Recently, Mijumbi et al. [5] proposed an approach to tackle the online VNF-FGE and VNFs-SCH by proposing greedy and metaheuristic (tabu search) approaches aiming at reducing the flow execution time. The algorithms perform both mapping and scheduling at the same time (one-shot) resulting in high acceptance ratio, low average flow time and low embedding cost. This work considers a resource sharing approach that allows a given VM to process multiple VNFs, one after another (possibly) from a queue.

3 Proposed Virtual Network Function Allocation Scheme

The features of the proposed dynamic NFV-RA scheme in this paper are as follows. Unlike previous research on resource allocation, each virtual node processes one or more functions at a time using multiprocessing technologies.

- Support one or more VNF component instances (VNFCIs) that provide the same functionality within a VNF through scale out
- Online RA algorithm supporting integration of VNF-FGE and VNF-SCH
- Multi-tenancy support

To explain the proposed VNF allocation scheme, we define the following variables and functions.

- N : A set of all virtual nodes, $N = \{1, 2, 3, 4, \dots, n\}$
- n : The number of virtual nodes
- m : The number of VNFs
- S : NS, consists of m sequential VNF, $F = \{1, 2, 3, 4, \dots, m\}$
- $\rho_{i,j}$: Processing time of VNF i at node j
- δ_i : The buffer used by the node to which function i is mapped
- B_j : At some point, the available buffer size at node j
- $\beta_{i,j}$: 1 if node j can handle function i , 0 otherwise
- t_i : Deadline time for which service to process
- t_i : Completion time of VNF i
- t_c : Current time of VNF i
- t_a : The time at which mapping and scheduling requests for the service arrived on the physical network
- π_j : Expected completion time of the last function waiting for processing at the corresponding virtual node j
- μ_i : Start time of the first function waiting for processing at the corresponding virtual node j

The NFV-RA problem is to select a virtual node for $j \in N(i)$ each VNF i and to select a completion time t_i . Therefore, the NFV-RA problem is divided into two stages. First,

a virtual node $j \in N(i)$ (mapping problem) to which each VNF should be mapped is selected, and the order in which VNFs are executed in each node is determined (scheduling problem). Figure 2 shows the pseudo code of the proposed dynamic NFV-RA in this paper and Fig. 3 shows the VNFs supported by each node and each node constituting the network. Assume that the request for $S_1 = \{f8 - f2 - f3 - f6 - f5\}$ arrives at time $T1$. Then $S_2 = \{f6 - f8 - f4\}$ arrives. Figure 4 shows the scheduling map of the existing algorithm [5] and Fig. 5 shows the scheduling map proposed in this paper. Figures 4 and 5 show the scheduling map after S_2 has been allocated, respectively. For the proposed algorithm, $f8$ is performed at n_7 to reduce the queue latency at n_1 .

```

function vnf_resource_alloc (S,N,T) {
  //T = {speed, sharing}
  matrix[] []; //graph
  graph *[]; //pointer array of graph
  graph = setup_node (matrix)

  for (NF i ∈ S) {
    N' = {} //node initialization
    If (i == 1)  $t_{i-1} = t_a$  //time initialization

    for (Node j ∈ N) {
       $t_e = \rho_{i,j} + \max(\pi_j, t_{i-1})$ 
      If (( $\beta_{i,j} == 1$ )  $\wedge$  ( $B_j \geq \delta_i$ )  $\wedge$  ( $t_e \leq t_l$ )) then {
        N' = N'  $\cup$  n
        If ((T == sharing)  $\wedge$  ( $t_i \neq \mu_j$ )) then {
          vnf_resource_alloc (NS( $\mu_j$ ),N,T)
        }
      }
    }
  }
  If (N' == 0)
    { Reset the substrate network; return1;}
  sort(N',T)
  mapping (i,select_top(N'))
  update ( $\pi_j$ )
   $t_i = \max(\pi_j, t_{i-1})$ 
  update ( $B_j, t_{i-1}$ )
}
}

```

Fig. 2. Pseudo code of the proposed VNF allocation scheme

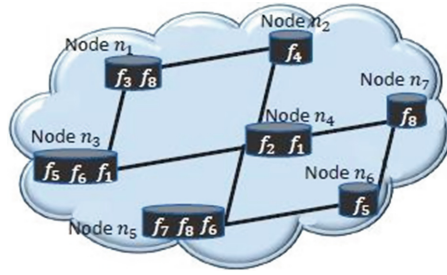


Fig. 3. Node capabilities and architecture of example node

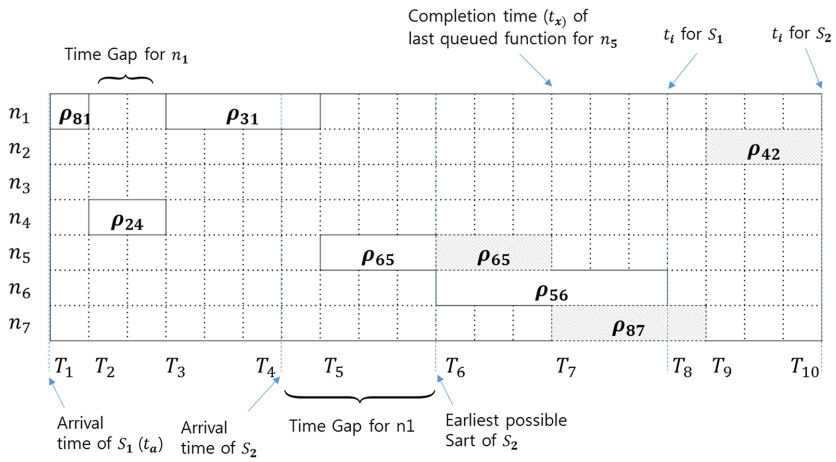


Fig. 4. Static virtual network function scheduling

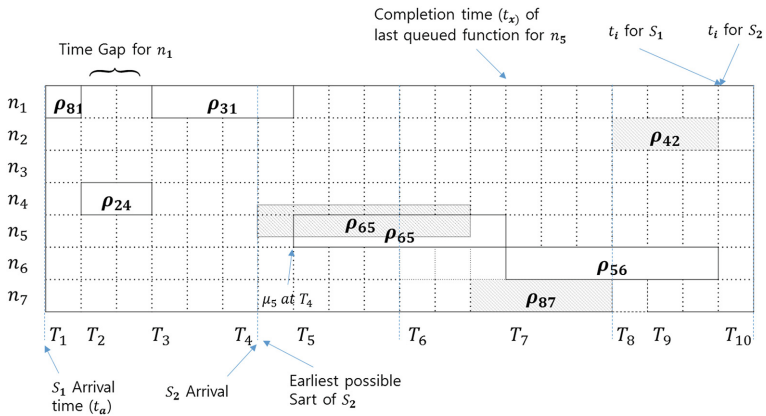


Fig. 5. Proposed virtual network function scheduling

Through the comparison of Figs. 4 and 5, it can be seen that the proposed algorithm reduces the execution time by one time unit.

4 Conclusion

One of the main challenges for the deployment of NFV is the efficient resource (VNF) allocation of demanded network services in NFV-based network infrastructures. However, the effective mapping and scheduling of VNFs are essential to successfully provide NFV services. In this paper, we proposed revised online (dynamic) virtual network function allocation scheme to cope with successive NS requests. Unlike previous research on resource allocation, we assumed that each virtual node processes one or more functions at a time using multi-processing technologies as in the real environment. Compared with existing algorithms, it can be seen that the proposed algorithm reduces execution time by 10% time unit.

References

1. Herrera, J.G., Botero, J.F.: Resource allocation in NFV: A comprehensive survey. *IEEE Trans. Netw. Serv. Manage.* **13**(3), 518–532 (2016)
2. Kim, H.: Network function virtualization (NFV) platform for wellness in high-speed network. In: Kim, K., Joukov, N. (eds.) *Information Science and Applications (ICISA) 2016*. LNEE, vol. 376, pp. 1459–1464. Springer, Heidelberg (2016). doi:[10.1007/978-981-10-0557-2_140](https://doi.org/10.1007/978-981-10-0557-2_140)
3. Mijumbi, R., Serrat, J., Gorricho, J., Bouten, N., De Turck, F., Boutaba, R.: Network function virtualization: state-of-the-art and research challenges. *IEEE Commun. Surv. Tutorials* PP(99), 1 (2015)
4. Riera, J.F., Hesselbach, X., Escalona, E., García-Espin, J.A., Grasa, E.: On the complex scheduling formulation of virtual network functions over optical networks. In: *International Conference Transparent Optical Networks (ICTON)*, pp. 1–5 (2014)
5. Beck, M.T., Botero, J.F.: Coordinated allocation of service function chains. In: *IEEE Global Communications Conference*, pp. 1–6 (2015)
6. Mijumbi, R., Serrat, J., Gorricho, J.-L., Latr, S., Charalambides, M., Lopez, D.: Management and orchestration challenges in network functions virtualization. *IEEE Commun. Mag.* **54**(1), 98–105 (2016)
7. Mijumbi, R., Serrat, J., Gorricho, J.-L., Bouten, N., De Turck, F.: Design and evaluation of algorithms for mapping and scheduling of virtual network functions. In: *IEEE Conference on Network Softwarization (NetSoft)*, pp. 1–9 (2015)
8. Kao, H.-Y., Yang, Y.-M., Huang, C.-H.: Dynamic virtual machines placement in a cloud environment by multi-objective programming approaches. In: *International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS)*, pp. 364–365 (2015)
9. ETSI: *Network Functions Virtualisation (NFV): Virtual Network Functions Architecture* (2014)
10. Munoz, R., Vilalta, R., Casellas, R., Martinez, R., Szyrkowicz, T., Autenrieth, A., Lopez, V., Lopez, D.: SDN/NFV orchestration for dynamic deployment of virtual SDN controllers as VNF for multi-tenant optical networks. In: *Optical Fiber Communications Conference and Exhibition (OFC)*, pp. 1–3 (2015)