JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 40, 165-176 (2024)
DOI: 10.6688/JISE.202401_40(1).0010

Adaptive Model for the Placement of Network Services

BRAJESH KUMAR UMRAO[1,+], MAHENDRA PRATAP YADAV[2]
AND DHARMENDRA KUMAR YADAV[3]
[1]*Department of Computer Science and Engineering*
ITER, Siksha 'O' Anusandhan Deemed to be University
Bhubaneswar, Odisha, 751030 India
[2]*Department of Computer Science and Engineering*
IIIT Pune, Maharashtra, 411041 India
[3]*Department of Computer Science and Engineering*
MNNIT Allahabad, Prayagraj, 211004 India
E-mail: brajeshumrao84@gmail.com[+]; mahendra@iiitp.ac.in; dky@mnnit.ac.in

The popularity of Network Function Virtualization (NFV) increases due to offering flexible and efficient methods for placing the chains of Virtual Network Functions (VNFs). The majority of recent studies consider consistent (immutable) bandwidth and flow processing while implementing the NFV-enabled network. The consistent flow processing and bandwidth requirement has become crucial in NFV-enabled network, where the network behaviour is highly dynamic. Due to the regular fluctuation in network traffic, the bandwidth demand, flow processing requirements, and the available Point-of-Presence (PoP) the network changes continuously. This article presents an adaptive model for the placement of VNF chains in order to mitigate the limitations mentioned above. The objective of re-arranging the prior placement of service chain is to maintain the performance of flow processing despite the variation in flow processing and resource requirements. Meanwhile, the proposed adaptive model minimizes the number of changes (*i.e.*, re-allocation of VNFs) in the network. The simulation results show that the proposed model delivers the network services without disruption. Moreover, it also maintains the performance of flow processing and minimizes the 12-25% modifications in the previously placed VNFs.

Keywords: NFV, VNF, bandwidth demand, flow processing, Point-of-Presence, adaptive, placement, flow

1. INTRODUCTION

Network Function Virtualization (NFV) is a modern technology, whose prime objective is to migrate the functions like routing, caching, and proxy from proprietary devices (middleboxes) to software programs running on x86 servers or Virtual Machines (VMs) [1, 2]. The softwarization of network functionality provides several benefits, such as it minimizes the operational expenses and maintenance cost of the network, provides cheaper updation on the network functions, and offers flexibility in the placement and chaining of network services across the network.

NFV has made the advances in the different facet of network, from designing of service chains to the management and orchestration of the service chains. Despite the different advances in NFV, several research challenges are still there. The placement of service chains is one of them. The service chain placement is a process of determining the locations for placing the VNFs in the NFV-enabled network, and routing the network traffic across

---

Received May 27, 2022; revised October 30, 2022; accepted August 1, 2023.
Communicated by Changqiao Xu.
[+] Corresponding author.

the network according to the specification provided in the Service Function Chaining (SFC) [3]. Software-Defined Networking (SDN) may be used in NFV for routing the traffic requests, and it enables service chin placement in a flexible way [4]. It has been investigated that placement of network services is NP-hard problem [1]. This problem has been studied for different objectives, such as optimizing the operating cost of the network, and maximizing usage of network resources.

The existing approaches consider the consistent arrival of traffic requests, and available network resources in the network while deploying the service chains in the network. However, the variation on available network resources depends on the number of network requests. The behaviour of the incoming traffic requests toward the network is highly dynamic with respect to time. Therefore, considering the constant flow processing, bandwidth requirement, available server and link resources, and arrival of network requests is the main constraint in the NFV-enabled network.

To overcome the above-mentioned limitations, we present an Integer Linear Programming (ILP) model for orchestrating network services. This model enables the service provider to (re)plan the previously allocated VNFs to deal with the fluctuating network's resources and varying network requests. To this end, the adaptive model (re)arranges the chaining between the VNFs for satisfying the traffic requests. Now, we summarize our contribution as follows:

- We present an adaptive ILP model for the placement of network services. The ILP model is implemented using IBM ILOG CPLEX Optimizer.
- The performance of the model is evaluated using Internet2 topology and real-world traffic traces.
- The performance of the proposed model is evaluated in term of number of changes in placed VNF, number of changes in service chains, and efficiency of the proposed model in terms of average execution time.
- Now, we compare the performance of the proposed adaptive model with the state-of-the-art solutions from the articles [1, 4].
- The experimental results demonstrate that the adaptive model reduces 12-25% changes in re-positioning the VNFs, and 1.5-2 times in re-mapping the logical links on the physical links.

The rest of the paper is organized as follows: Section 2 presents the motivation of the proposed work with the support of experiment. Section 3 comprises of the existing work related to our proposed approach. Section 4 provides the detailed description of adaptive model for provisioning the network services. Section 5 consists of the experimental setup and result analysis. Section 6 includes the concluding remarks with some light on the future direction.

2. MOTIVATION

The NFV technology is aimed to enhance efficiency, performance, and flexibility of network service deployment. The complexity of the network service deployment depends on the constraints and the requirements that must be satisfied during the deployment of network services. For example, compute resources of the x86 servers, end-to-end delay of network services, required bandwidth between the servers or PoPs. The network service deployment problem is identified as an NP-hard. The existing solutions assume the constant flow processing, bandwidth requirement, availability of server resources, and incoming network requests. In the NFV-enabled network, flow processing, bandwidth requirement, and available server resources depend on the variation of network requests with respect to time. The network requests are highly dynamic with respect to time.

3. RELATED WORK

The VNF placement and traffic routing problem is solved with different objectives. Some researchers consider that the VNF placement and traffic routing problem is the extension of Virtual Network Embedding [5, 6]. The network services comprise a chain of VNFs. The VNF chain can be represented as a directed graph. The directed graph is mapped on the substrate network. One or multiple VNFs can be mapped on a server and each link of the directed graph is mapped on the physical link. The problem mentioned above has been investigated as NP-hard [1, 2, 6]. The existing techniques for solving the VNF placement and chaining problem can be categorized as an exact, heuristic, and meta-heuristic. The search space increases exponentially with increasing problem size [1-3, 7]. The major portion of the existing research work is used an ILP formulation for solving the VNF placement problem [8-11]. There are some solutions based on Mixed Integer Linear Programming [12-14]. Marotta *et al.* [12] formulated a MILP model for the placement of VNFs and routing problem consideration with the demand uncertainties and latency constraints. Additionally, they proposed a time-efficient heuristic that minimizes the solution complexity. Sang *et al.* [13] formulated the MILP model and presented a greedy heuristic to solve the proposed VNF allocation problem. Qu *et al.* [14] obtained the optimal solution for the VNF placement and traffic routing problem using MILP formulation. They developed a heuristic algorithm that uses a greedy approach to get the k-shortest path. The problem objective maximizes the service acceptance rate and minimizes the end-to-end delay. Some researchers selected single-objective meta-heuristic approaches to place the VNFs in an NFV-enabled network. Kim *et al.* [15] presented a Genetic Algorithm (GA) based Algorithm for deploying the VNFs in the cloud environment. The problem objective is to minimize the total power consumption and meet the required service latency. Xing *et al.* [16] proposed a Gray Wolf Optimizer (GWO) for placing the VNFs in the NFV-enabled network. The objective of the optimizer is to minimize the end-to-end delay of requested services.

Chantre and Fonseca [17] identify the redundant placement of VNFs in NFV-enabled networks. They selected Particle Swarm Optimization (PSO) to mitigate the problem of redundant VNF placement and minimize the end-to-end delay of requested services. Farshin and Sharifian [18] mixed the features of both Ant Colony Optimization (ACO) and GWO to get the solution for the VNF placement and traffic routing problem. They used ACO to create the best routing path, and then VNFs are deployed on servers that fall on the selected route. The service chains are distributed over the different cloudlets. These cloudlets are connected to each network router so that each service chain somewhat utilizes the cloudlet resources. GWO tunes the controllable components of ACO. The mixed meta-heuristic performs better load balancing but suffers from high time complexity.

There is limited research that uses the multi-objective meta-heuristic approaches for mapping the service chains in the NFV-enabled network. Khebbache *et al.* [19] presented a Non-dominated Sorting Genetic Algorithm (NSGA) to solve the multi-objective VNF placement problem. The objective of the proposed meta-heuristic is to reduce the total mapping expenses and bandwidth usage. The experimental results demonstrate that the proposed NSGA functions better than some heuristic approaches. Cao *et al.* [20] built another version of NSGA for mapping the service chains over an NFV-enabled network. The objective of NSGA is to maximize the link utilization and reduce the bandwidth consumption simultaneously. The experimental shreds of evidence demonstrate that NSGA provides a better trade-off between two conflicting objectives than the heuristics.

Despite the various advances in VNF placement and chaining, existing solutions did not consider fluctuations in localization and bandwidth due to varying network traffic demands. A contemporary approach to dealing with these fluctuations is executing the VNF placement algorithm again and rearranging them according to obtained results. This solution is effective but expensive in terms of computations and does not respond appropriately to the dynamic flow. The given work in the papers [21, 22] is very close to an efficient solution to this problem. In these papers, the authors apply the genetic approaches to locate the network functions with scalable computing abilities. In this solution, they have considered the isolated network functions. Therefore, traffic steers with the exact requirements for greater VNFs capacity without considering any possible global optimization. Considering the limitations mentioned above, this article presents an adaptive model to re-adjust the network infrastructure against varying network traffic demands by identifying bottlenecks in the traffic processing, rearranging the placement and chaining of network functions, and aiming to reduce the disruption in the flow processing.

4. ADAPTIVE MODEL FOR PROVISIONING THE NETWORK SERVICES

To deal with the fluctuating nature of network traffic and rearrange the VNF allocation without degrading the network performance or wasting the physical resources of the network. It is necessary to review the available formulations and heuristics for the VNF placement problem. To this end, we have selected an adaptive version of the ILP formulation proposed by Bari *et al.* [11]. They have formulated the static placement and chaining of VNFs by applying constraints in a linear system. We present an adaptive ILP formulation for the placement and chaining of virtual network functions.

4.1 System Model

Model Input: The proposed ILP model considers a set of traffic requests T and a physical network $G = (V, L)$, where V and L represent the set of switches and a set of links in the network, respectively. We assume that the VNFs are deployed on x86 servers that located in the network. The locations of the servers in the network are called the Point-of-Presences (PoPs). The available set of servers in the network are denoted by S. We define a binary variable $h_{v,s} \in \{0,1\}$ that expresses the association of a server $s \in S$ to a network switch $v \in V$. The compute capacity of a server s is represented by c_s. The propagation delay and bandwidth of a physical link $(u, v) \in L$ are denoted by δ_{uv} and $\beta_{u,v}$, respectively. There is a function $\eta(u)$ that returns a set of adjacent switches of the switch u.

Each network service comprises different types of VNFs in a pre-determined sequence. There is a symbol P that denotes the possible types of VNFs. Each type of a VNF $p \in P$ consumes the computing resources of the underlying server, which is denoted by k_p. The processing capacity and processing delay of each type of VNF $p \in P$ is denoted by c_p and δ_p, respectively.

A network traffic request $t \in T$ can be represented by 5-tuple $t = \langle u^t, v^t, \mu^t, \beta^t, \delta^t \rangle$, where u^t and v^t are the source and destination points of a traffic t in the network. μ^t represents a sequence of VNFs through which a network traffic must pass. The bandwidth demand and maximum allowed delay of a network traffic t is denoted by β^t and δ^t, respectively. In the proposed ILP model, the sequence of VNFs u^t can be represented by a directed graph $G^t = (V^t, L^t)$, where V^t and L^t are the set of traffic nodes and the set of links between the nodes. Here, we define a function $\eta^t(u_1)$ to find the adjacent of $u_1 \in V^t$. Also, we define a binary variable $q^t_{up} = \{0, 1\}$ that denote the type node u is p in a network traffic t.

VNFs Enumeration: We enumerate all the VNFs in the network by computing the maximum number of VNFs each type that can be deployed on the servers. This number can be computed by dividing the resource capacity of a server to the resource requirement of VNF type. For example, a server has 16 CPU cores and the required CPU cores are 4 and 8 for proxy and IDS, respectively. Therefore, we can deploy 4 proxy and 2 IDS on the server. The set of enumerated VNFs in the network is denoted by the symbol M. Each VNF $m \in M$ is associated to a particular server $s \in S$. We define a function $\sigma(m)$ (*i.e.*, $\sigma(m) = s$ if VNF m is associated to the server s) to represent this relationship. We have another function $\Omega(s)$ (*i.e.*, $\Omega(s) = \{m \mid \sigma(m) = s\}$, where $m \in M$ and $s \in S$) that represent the reverse association. Now, we define a binary variable $d_{mp} = \{0, 1\}$ that represent the VNF type. There is a function $\lambda(m)$ (*i.e.*, $\lambda(m) = \{p \mid d_{mp} = 1\}$ that return the VNF type m. We define a binary variable $x_m = \{0, 1\}$ to show whether m is active or not.

Model Output: A set of binary variables are used in the formulation of ILP model. The binary variable $y^t_{um} = \{0, 1\}$ represents the mapping of node u of traffic t to the VNF m. The binary variable $\hat{y}^t_{um} = \{0, 1\}$ represents that the current placement of node u of traffic t has changed to its previous placement. Now, we define another binary variable $z^t_{uv} = \{0, 1\}$ to determine the mapping of a traffic node to the switch in the physical network. Similarly, the variable $\hat{z}^t_{uv} = \{0, 1\}$ indicate that current mapping is changed to its previous mapping. Finally, we define a decision variable $w^{t_{ij}}_{uv} = \{0, 1\}$ that represent the mapping of network traffic link (i, j) on the physical network link (u, v). The decision variable $\hat{w}^{t_{ij}}_{uv} = \{0, 1\}$ indicates that current network traffic link (i, j) mapping is changed to its previous mapping.

4.2 ILP Formulation

The objective function of the ILP model comprises two components. The first component of the objective function minimizes the resource consumption by minimizing the number of allocated VNFs (*i.e.*, z^t_{uv}) and number of allocated links (*i.e.*, $w^{t_{ij}}_{uv}$). The second component of the objective function consists of two parts. The first part of the second component minimizes the changes in the previously allocated VNFs, whereas second part of the second component minimizes the changes in previously allocated traffic links. Here, α and β are used to set the relative significance of the respective components.

Objective Function:

$$Minimize\left(\sum_{t\in T}\sum_{u\in V^t}\sum_{v\in V} z_{uv}^{t} + \sum_{t\in T}\sum_{(i,j)\in L^t}\sum_{(u,v)\in L} w_{uv}^{t_{ij}}\right) - \left(\alpha\sum_{t\in T}\sum_{u\in V^t}\sum_{v\in V} \hat{z}_{uv}^{t} + \beta\sum_{t\in T}\sum_{(i,j)\in L^t}\sum_{(u,v)\in L} \hat{w}_{uv}^{t_{ij}}\right)$$

Subjected to:

$$\sum_{m\in\Omega(s)} x_m \times k_m \le c_s, \quad \forall s\in S \tag{1}$$

$$\sum_{t\in T}\sum_{u\in V^t} y_{um}^{t} \times B^t \le c_{\lambda(m)} \quad \forall m\in\{x \mid x\in M,\ a_x = 1\} \tag{2}$$

$$y_{um}^{t} \times q_{up}^{t} = d_{mp}, \quad \forall t\in T, u\in V^t, m\in M, p\in P \tag{3}$$

$$\sum_{t\in T}\sum_{u\in V^t} y_{um}^{t} = 1, \quad \forall m\in M \tag{4}$$

$$w_{uv}^{t_{ij}} + w_{vu}^{t_{ij}} \le 1, \forall (i,j)\in\{(c,d) \mid c\in V^t, d\in\lambda^t(c), d>c\} u,v\in V \tag{5}$$

$$\sum_{u\in V}\sum_{v\in V}(w_{uv}^{t_{ij}} + w_{vu}^{t_{ij}})\times B^t \le B_{uv}, \forall t\in T,\ \forall (i,j)\in\{(c,d) \mid c\in V^t, d\in\eta^t(c), d>c\} \tag{6}$$

$$\sum_{u\in V}\sum_{v\in V}(w_{uv}^{t_{ij}} - w_{vu}^{t_{ij}}) = c_{iu}^{t} - c_{ju}^{t}, \forall t\in T,\ \forall (i,j)\in\{(c,d) \mid c\in V^t, d\in\eta^t(c), d>c\} \tag{7}$$

$$\sum_{u\in V}\sum_{v\in V}(w_{uv}^{t_{ij}} + w_{vu}^{t_{ij}}) \ge 1, \forall t\in T,\ \forall (i,j)\in\{(c,d) \mid c\in V^t, d\in\eta^t(c), d>c\} \tag{8}$$

The first four constraints are used to express the limitation of network infrastructure. Eq. (1) ensures that the total provisioned resources by all VNFs on a server does not exceed the available computation capacity of the server. Eq. (2) makes sure that the amount of passing traffic through a VNF does not exceed the functional capacity of the VNF. Eq. (3) is responsible for mapping a traffic node the appropriate type of VNF. Eq. (4) ensures the mapping of each traffic node to exactly one VNF.

Eqs. (5)-(8) describe the chaining constraints of the network traffic requests. Each traffic link is mapped to the single direction on the physical network link. Eq. (5) satisfies the single direction link mapping constraints to the link of the physical network. The traffic bandwidth demand does not exceed than the physical link on which it is mapped. Eq. (6) responsible for satisfying the bandwidth constraints of the physical path. Eq. (7) ensures the flow conservation constraints. It means the incoming and outgoing flow must be equal at each network node except the source and destination nodes. Eq. (8) ensures that each node of a traffic is mapped on the path of the network.

4.3 Procedure for Provisioning the Network Services

The proposed algorithm considers that the service chain requests are delivered on demand. Each service chain comprises a sequence of VNFs. Additionally, the CPU and bandwidth require for each VNF in the same service chain. The proposed algorithm is responsible for deploying the VNFs in the requested service chain and satisfying the resource requirements. In other word, a service chain is mapped on VNFs hosted on x86 servers available in the network. Meanwhile, it preserves the links and sequence between the VNFs. The proposed algorithm aims to periodically re-arrange the previously placed VNF chains. During the process of re-arranging the previous placement of VNF chains, proposed algorithm maintains the flow performance despite the variation in flow processing and resource requirements. At the same time, the proposed algorithm minimizes the number of changes in the network.

Algorithm 1: Adaptive Approach for Provisioning the Network Services

```
Inputs: Network Infrastructure $G(V, L, S)$, a set of VNF types $P$, a batch of traffic request $T$.
Constraints: $x_m \times k_m \le c_s, y_{um}^t \times B^t \le c_{\lambda(m)}, y_{um}^t \times q_{up}^t = d_{mp}, y_{um}^t = 1, w_{uv}^{t_{ij}} + w_{vu}^{t_{ij}} \le 1,$
             $(w_{uv}^{t_{ij}} + w_{vu}^{t_{ij}}) \times B^t \le B_{uv}, (w_{uv}^{t_{ij}} - w_{vu}^{t_{ij}}) = c_{iu}^t - c_{ju}^t, (w_{uv}^{t_{ij}} + w_{vu}^{t_{ij}}) \ge 1.$
for $t \leftarrow 1$ to $|T|$ do
    for $u^t \leftarrow u^t$ to $|V^t|$ do
        for $v \leftarrow u^t$ to $|V|$ do
            minimize $\{z_{uv}^t - \alpha \times \hat{z}_{uv}^t\}$
        end for
    end for
end for
for $t \leftarrow 1$ to $|T|$ do
    for $(i, j) \in L^t$ to $|L^t|$ do
        for $(u, v) \in L$ to $L$ do
            minimize $\{w_{uv}^{t_{ij}} - \beta \times \hat{w}_{uv}^{t_{ij}}\}$
        end for
    end for
end for
Output: $z_{uv}^t, \hat{z}_{uv}^t, w_{uv}^{t_{ij}}, \hat{w}_{uv}^{t_{ij}}$
```

5. EXPERIMENTAL SETUP AND RESULT ANALYSIS

We have performed a couple of experiments to evaluate the performance of the proposed adaptive model. The experimental work is carried out in a machine with configuration Intel(R) Core(TM) i7-4790 3.60 GHz processors and 12 GB RAM. We have installed the Ubuntu18.04 operating system. We use the Internet2 [16] topology as a network infrastructure with 12 nodes (switches) and 15 links. There are seven x86 servers in the Internet2 topology. Each server comprises the 16 CPUs. These servers are associated with the switches with nodes 1, 3, 4, 5, 9, 10, and 11 of network. Each physical link of network has uniform bandwidth and delay that is 10 Gbps and 10 ms, respectively. We have four types of VNF image – firewall, proxy, IDS, and NAT. These VNF images are randomly selected while creating a network service. The network services are represented using a line topology. The endpoints are selected randomly in each network service. The 30 network requests are submitted in a batch for processing. The length of each submitted network request is uniform (*i.e.*, three) in a batch.

5.1 Evaluation

We adopt a similar strategy employed in the state-of-the-art [1] for the experiment. We focus on analyzing the results, which are generated using adaptive model. To evaluate the capabilities of the adaptive model for re-designing the network with minimum distortion, we have submitted the number of network requests between regular and peak hour's workloads. Therefore, re-planning is required for placing the VNFs, which maintains the stability and performance of the network. In order to evaluate the performance of the proposed model, we compare the performance of the proposed model with the research work [1, 4]. A batch of traffic requests are submitted on the adaptive model. The proposed adaptive model provides the locations for placing the VNFs within the network.

5.2 Number of Changes Required in the Network

Fig. 1 demonstrates the number of changes in the previously placed VNFs for the varying traffic requests, and Fig. 2 demonstrates the number of changes in the previously mapped service chains. The number of traffic requests varies from 10% to 80%. Different curves are used to indicates the various scenarios. The value of α and β (respective weighting factors) in the objective function are chosen to be 1 for this experiment. We have observed that a constraint does not bound all possible values of these weighting factors since the network characteristics (*i.e.*, load and size) may influence the effectiveness of any setting for them. We leave the analysis of the relationship among the weighting factors for future research.

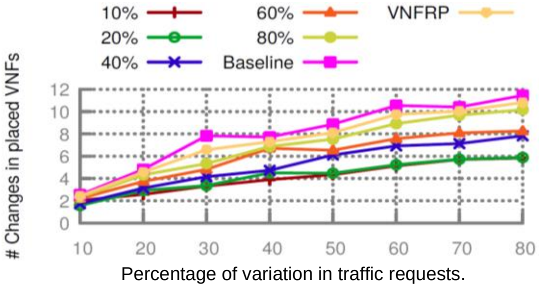

Fig. 1. Reassignment of VNFs due to fluctuating traffic requests.

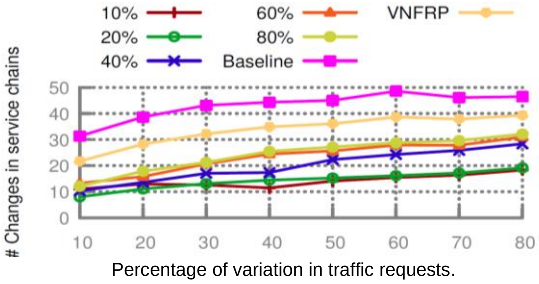

Fig. 2. Reassignment of service chains due to fluctuating traffic requests.

We have perceived that the number of changes required (y-axis) to fine-tune the network for the new traffic demand is proportional to the following: (i) the percentage of service chains with the increased demand; and (ii) the outreached demand values (x-axis). Moreover, the number of changes in the VNF positions are significantly lesser than the reassignment of the service chains. This indicates that the adaptive model is feasible for the real-world applications because the time required for instantiating or migrating a VNF is lower than the programming a routing device. From Fig. 1, we have observed that the adaptive model reduces up to 25% and 12% changes in the previously placed VNFs compared with the baseline and VNFRP approaches, respectively. Moreover, the adaptive model reduces the number of changes in the previously mapped logical links on the physical links up to 1.5 times and 2 times compared with the VNFRP and baseline approaches, respectively.

5.3 Impact of Over-Commitment

We initially submit 40 network requests for deploying on the network under the expected workload. The flow processing demand is increased by 40%. The number of traffic requests are varied from 10% to 40% during the evaluation of over-commitment. The higher number of traffic requests increases the chance of degrading the network performance. Fig. 3 demonstrate the required number of changes in the network to adjust the service chain in response to the varying network traffic demand. Fig. 3 shows that the excessive number of network requests decreases the number of modifications in the network. The 20% traffic requests reduce the 20% overall changes in the network infrastructure compared with 10% traffic requests.

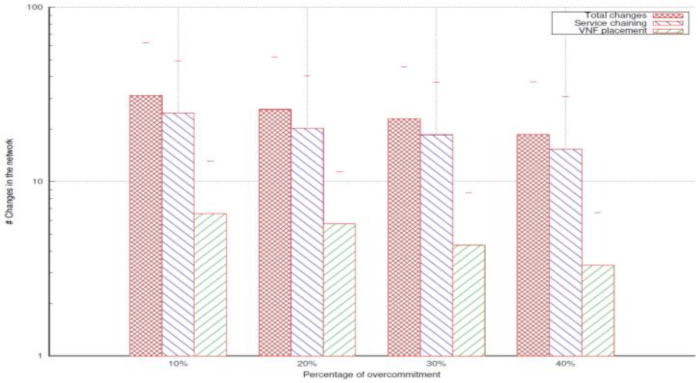

Fig. 3. Impact of over-commitment on the network infrastructure.

5.4 Efficiency in Re-planning

Fig. 4 demonstrate that the average time is required to find the best solution for (re)arranging the previous placement of network services. The proposed adaptive model can find the results within 5 minutes for all cases. We have observed that the VNFRP heuristic rearranges the previous placement less than 2 to 4 times than the proposed adaptive model, but it compromises the solution optimality.

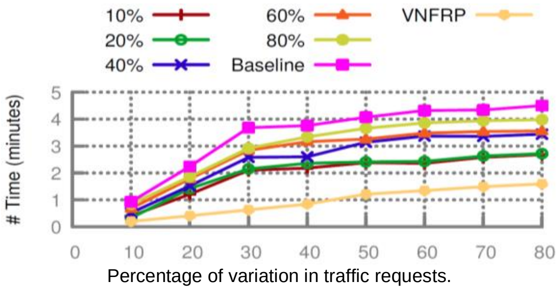

Fig. 4. Average time in re-allocating the network services.

Form the Fig. 4, we have observed that our model solves the problem in less time than the baseline approach for all cases in which all service chains are re-assigned. We have observed that the increased number of network requests required more time to discover the solution. Though the VNF placement and chaining is an NP-hard problem, these results propound that attaining an exact solution is possible for small-scale and medium problems. But, more research is required to evaluate the computing time limits for the large-scale problem.

6. CONCLUDING REMARKS

In this article, we have studied the placement of network services in NFV-enabled network. The existing approaches for this problem have considered the constant flow processing, bandwidth requirement, and server resources, which becomes a main limitation. To mitigate this limitation, we presented an adaptive model for re-adjusting the VNF and logical link for varying demand of network requests. Meanwhile, the proposed adaptive model reduces the number of changes in the network infrastructure. The experimental results demonstrate that the proposed adaptive model reduces the 12-25% in re-positioning the VNFs. Moreover, it also reduces changes by 1.5-2 times in the VNF chaining. In the future, we will devise a method for traffic prediction and integration this method with the adaptive model.

REFERENCES

1. 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, 2016, pp. 725-739.
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. 13, 2017, pp. 172-85.
3. B. K. Umrao and D. K. Yadav, “Placement of virtual network functions for network services,” *International Journal of Network Management*, 2023, p. e2232.
4. M. A. Abdelaal, G. A. Ebrahim, and W. R. Anis, “Efficient placement of service function chains in cloud computing environments,” *Electronics*, Vol. 10, 2021, p. 323.
5. N. Shahriar, R. Ahmed, S. R. Chowdhury, M. M. A. Khan, R. Boutaba, J. Mitra, and F. Zeng, “Virtual network embedding with guaranteed connectivity under multiple

substrate link failures," *IEEE Transactions on Communications*, Vol. 68, 2019, pp. 1025-043.

6. B. K. Umrao and D. K. Yadav, "Algorithms for functionalities of virtual network: A survey," *The Journal of Supercomputing*, Vol. 77, 2021, pp. 7368-7439.
7. M. C. Luizelli, D. Raz, and Y. Sa'ar, "Optimizing NFV chain deployment through minimizing the cost of virtual switching," in *Proceedings of IEEE Conference on Computer Communications*, 2018, pp. 2150-2158.
8. D. Bhamare, M. Samaka, A. Erbad, R. Jain, L. Gupta, and H. A. Chan, "Optimal virtual network function placement in multi-cloud service function chaining architecture," *Computer Communications*, Vol. 102, 2017, pp. 1-16.
9. Q. Sun, P. Lu, W. Lu, and Z. Zhu, "Forecast-assisted NFV service chain deployment based on affiliation-aware VNF placement," in *Proceedings of IEEE Global Communications Conference*, 2016, pp. 1-6.
10. D. Li, P. Hong, K. Xue *et al.*, "Virtual network function placement considering resource optimization and SFC requests in cloud datacenter," *IEEE Transactions on Parallel and Distributed Systems*, Vol. 29, 2018, pp. 1664-1677.
11. B. Yi, X. Wang, and M. Huang, "Design and evaluation of schemes for provisioning service function chain with function scalability," *Journal of Network and Computer Applications*, Vol. 93, 2017, pp. 197-214.
12. A. Marotta, E. Zola, F. D'Andreagiovanni, and A. Kassler, "A fast robust optimization-based heuristic for the deployment of green virtual network functions," *Journal of Network and Computer Applications*, Vol. 95, 2017, pp. 42-53.
13. Y. Sang, B. Ji, G. R. Gupta, X. Du, and L. Ye, "Provably efficient algorithms for joint placement and allocation of virtual network functions," in *Proceedings of IEEE Conference on Computer Communications*, 2017, pp. 1-9.
14. L. Qu, C. Assi, K. Shaban, and M. J. Khabbaz, "A reliability-aware network service chain provisioning with delay guarantees in NFV-enabled enterprise datacenter networks," *IEEE Transactions on Network and Service Management*, Vol. 14, 2017, pp. 554-568.
15. S. Kim, S. Park, Y. Kim, S. Kim, and K. Lee, "VNF-EQ: dynamic placement of virtual network functions for energy efficiency and QoS guarantee in NFV," *Cluster Computing*, Vol. 20, 2017, pp. 2107-2117.
16. H. Xing, X. Zhou, X. Wang, S. Luo, P. Dai, K. Li, and H. Yang, "An integer encoding grey wolf optimizer for virtual network function placement," *Applied Soft Computing*, Vol. 76, 2019, pp. 575-594.
17. H. D. Chantre and N. L. da Fonseca, "Redundant placement of virtualized network functions for LTE evolved multimedia broadcast multicast services," in *Proceedings of IEEE International Conference on Communications*, 2017, pp. 1-7.
18. A. Farshin and S. Sharifian, "A modified knowledge-based ant colony algorithm for virtual machine placement and simultaneous routing of NFV in distributed cloud architecture," *The Journal of Supercomputing*, Vol. 75, 2019, pp. 5520-5550.
19. S. Khebbache, M. Hadji, and D. Zeghlache, "A multi-objective non-dominated sorting genetic algorithm for VNF chains placement," in *Proceedings of the 15th IEEE Annual Consumer Communications and Networking Conference*, 2018, pp. 1-4.
20. J. Cao, Y. Zhang, W. An, X. Chen, J. Sun, and Y. Han, "VNF-FG design and VNF placement for 5G mobile networks," *Science China Information Sciences*, Vol. 60,

2017, pp. 10-15.

21. W. Rankothge, F. Le, A. Russo, and J. Lobo, "Experimental results on the use of genetic algorithms for scaling virtualized network functions," in *Proceedings of IEEE Conference on Network Function Virtualization and Software Defined Network*, 2015, pp. 47-53.
22. W. Rankothge, F. Le, A. Russo, and J. Lobo, "Optimizing resource allocation for virtualized network functions in a cloud center using genetic algorithms," *IEEE Transactions on Network and Service Management*, Vol. 14, 2017, pp. 343-356.

Brajesh Kumar Umrao is working as an Assistant Professor in the Department of Computer Science and Engineering, Institute of Technical Education and Research, Siksha 'O' Anusandhan Deemed to be University, Bhubaneswar, Odisha, India. He has submitted his Ph.D. thesis in the Computer Science and Engineering Department at Motilal Nehru National Institute of Technology Allahabad, Prayagraj, India. He received his M.Tech in year 2012 from Motilal Nehru National Institute of Technology Allahabad, Prayagraj, India. His research interests include cloud computing, software defined networking, network function virtualization, and deep reinforcement learning.

Mahendra Pratap Yadav is working as an Assistant Professor in the Department of CSE at IIIT Pune, Maharashtra, India. He received his M.Tech. in 2012 and Ph.D. in 2021 from MNNIT Allahabad, Prayagraj, India. His research interests include cloud computing, machine learning, formal methods and virtualization.

Dharmendra Kumar Yadav is working as a Professor in the Department of CSE at MNNIT Allahabad, Prayagraj, India. He received his Ph.D. from IITBombay, India. His research interests include distributed computing, cloud computing, machine learning formal method, and software engineering.