Orchestration of VNF Placement and Routing in 5G Networks

Gelareh Hasel Mehri, University of Tehran, 2020

***Abstract*—The fifth generation of wireless networks has been proposed as a platform to provide considerably higher network capacity, enable massive device connectivity with reduced latency and cost, and achieve considerable energy savings. In this regard, network design, management, and upgrades require innovative technologies and architectures. Enterprise and service provider networks are increasingly making use of Virtualized Network Functions (VNFs) and Software-Defined Networking (SDN) to reap the benefits of reduced CAPital EXpenditures (CAPEX) and OPerating EXpenses (OPEX). In this research, the problem of placing virtual network functions on existing resources and data routing through these functions to execute a particular service chain is investigated. A network consisting of switches, servers, and links with determined specifications was designed and simulated to this end. On the next step, an optimization problem was proposed, which was subjected to minimizing the total cost of utilizing links and severs of the network. The service provider view and the cloud provider requirements were both taken into account while modeling the cost function. Besides network constraints and topology constraints, a QoS constraint was considered, which exerted an upper bound on the congestions of the links. Due to the NP-hard nature of the problem, a small-scale network was simulated using MOSEK in MATLAB, and optimal results to MILP were driven. In the following, a heuristic algorithms was presented to solve the NP-hard optimization problem, and its performance was evaluated and compared to optimal results.**

***Index Terms*—5G, Software-Defined Networking, Network Function Virtualization, Routing, Heuristic.**

I. Introduction

IMPLEMENTATION of design, management, and operation of network infrastructure requires evolutions with the help of new technologies and architectures. Research fields such as Network Function Virtualization (NFV) and Software-Defined Networking (SDN) can satisfy this need for networks' evolution. The integration of NFV and SDN makes it possible to innovate in the networks by providing an efficient and flexible framework for collaborative control and scheduling of network functions.

This higher flexibility, as a result, poses challenges to the problem of VNF placement and routing. For example, cost control, Quality of Service (QoS), latency, congestion, energy consumption, justice, etc., are among the challenges raised in this topic and are considered as constraints in the optimization problem of VNFs placement and routing.

This dissertation aims to simultaneously solve the problem of VNFs' placement and traffic routing with the help of SDN in 5G networks. After presenting the integrated studying of these two problems, also called orchestration, the system's performance is analyzed using available constraints and awareness.

The remainder of this article is organized as follows. Chapter two gives a review of the basic concepts introduced in the fifth generation of telecommunication networks. Then, in the third chapter, the problem of orchestration of VNFs' placement and routing through them using SDN is presented, and the existing solutions are examined. Chapter four examines the current heuristic algorithms and offers two heuristic methods for the proposed optimization problem. The performance of one of them is compared with the optimal state by performing various simulations. Finally, the dissertation concludes with a summary of the results and suggestions in Chapter Five.

II. Conceptology and Related Works

This chapter aims to introduce the concepts and technologies that form the basis of the following chapters. The two notions of Software-Defined Networking and Network Function Virtualization, which are among the most dominant innovations presented in the fifth generation, are briefly presented.

A. Network Function Virtualization

Services in the telecommunications industry are traditionally based on network operators using specific equipment and physical devices for each function of a particular service. In addition, the service components have a precise order that should be reflected in the network topology and the placement of the service components. Together with the requirements for high quality, stability, and strict adherence to the protocol, these considerations have led to a long production cycle, very little service change, and severe dependence on specialized hardware.

On the other hand, users' needs for various and new services (short-term) with high data rates continue to increase. However, despite the high increase in demands, the increase in initial and operating costs can not lead to higher subscriptions because service providers have realized that rising prices only lead to lower consumers due to the high level of competition between other providers.

Therefore, Telecommunications Service Providers (TSPs) must constantly purchase, store and use new physical equipment. This usage of new physical equipment requires high and rapidly changing skills for the technicians using and managing them and mass development of network equipment such as base stations. All these lead to high start-up and maintenance

G. Hasel mehri was with the Department of Electrical and Computer Engineering, University of Tehran, Iran, e-mail: gelare.haselmehri@ut.ac.ir.

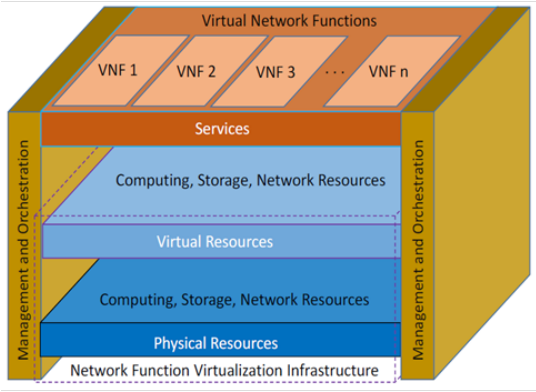

Fig. 1. NFV Architecture by ETSI

costs. TSPs have to find ways to build dynamic and service-aware networks to reduce production cycles, operating and initial costs, and increase the speed of service change.

NFV has been proposed to solve these challenges using virtualization technology to provide a new way to design, set up and manage network services. In November 2012, operators selected the European Telecommunications Standards Institute (ETSI) as the NFV industrial specification group center.

The main idea of NFV is to separate physical network equipment from the functions that run on it. Therefore, a TSP can present a network function, such as a firewall, as a simple software. This allows many types of network equipment to be fixed on large servers, switches, and repositories located in data centers, distributed network nodes, or at the user side. This way, a data service can be decomposed into a set of Virtualized Network Functions (VNFs) and run on software running on one or more industry-standard physical servers. VNFs can then be placed at different network locations without purchasing and installing new hardware.

1) NFV Architecture: According to ETSI, the NFV architecture includes three key elements: Network Function Virtualization Infrastructure (NFVI), Virtual Network Functions (VNF), and NFV MANagement and Orchestration (NFV MANO). These elements are shown in figure 1.

NFVI is a combination of hardware and software resources that make up the environment in which VNFs are located. Physical resources include commercial off-the-shelf (COTS) hardware, storage, and networking (including nodes and links) that process, store, and connect to VNFs. Virtual resources are an abstraction of computing, storage, and network resources. Abstraction is achieved using a virtualization layer (based on a virtual machine observer), separating virtual resources from physical resources.

A network function (NF) is a functional block in network infrastructure with specific external interfaces and functional behavior. Examples of NFs are home network elements, such as local input, and typical network functions, such as servers, DHCP firewalls, etc. Thus, a VNF is an implementation of an NF deployed on virtual resources such as a VM. A single VNF may consist of multiple internal components, so it can be deployed in multiple VMs, in which case each VM hosts a single component of the VNF. A service is a provision provided by TSP that consists of one or more NFs.

The MANO framework presents VNF provisioning capabilities and related operations, such as VNF configuration. These capabilities also include synchronizing and managing the lifecycle of physical resources or software that supports VNF infrastructure. It also includes databases used to store information and data models that define the deployment and life cycle properties of functions, services, and resources. MANO NFV focuses on all the specific virtualization management tasks that are essential to the NFV framework. In addition, the framework defines interfaces that can be used to communicate between different components of the MANO NFV and integrates with traditional network management systems.

B. Software-Defined Networking

As network size and complexity increase, network infrastructure requires more dynamic and flexible operations, programmability, and modifiable devices. Currently, the best technology to achieve this behavior is Software-Defined Networking (SDN). SDN provides a central control in the network where the data layer is separated from the control layer. Separating the control layer from the data layer adds the ability to improve performance programmability and remotely manage infrastructure to the network, allowing the network and business applications to change network policies as users and applications change.

In SDN, by optimizing the control activities, the possibility of optimal infrastructure management is created. Instead of being able to understand and interpret different protocols, network devices need to receive instructions from the SDN controller. So any network element can change instantly. The network can also meet urgent business needs and help personalize the network. The figure 2 shows the difference between the traditional network architecture and SDN. In conventional networks, the control panel and data plane are integrated into each network device which is not the case in SDN.

C. SDN-based NFV

NFV and SDN have a lot in common, as they both support the move to open-source software and standard network hardware. In particular, just as NFV intends to run NFs on industry-standard hardware, the SDN control layer can be implemented as mere software running on industry-standard hardware. In addition, both NFV and SDN are looking to achieve automation and virtualization to achieve their respective goals. NFV and SDN may complement each other to a large extent, and so combining them into one network solution may lead to more value. An SDN-based NFV structure is shown in figure 3.

This structure includes a control module, forwarding devices, and an NFV platform. The packet sending logic is determined by the SDN controller and is implemented on the forwarding devices through forwarding tables. Efficient protocols, such as OpenFlow can be used as standard interfaces to communicate between the central controller and distributed forwarding devices.

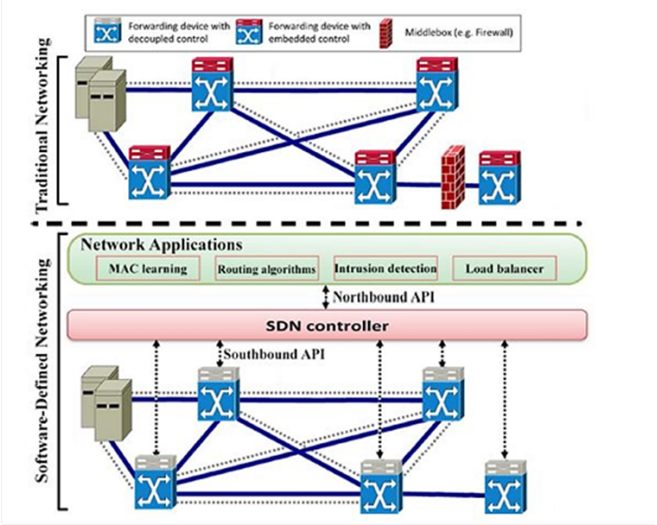

Fig. 2. SDN vs Conventional Network Architectures

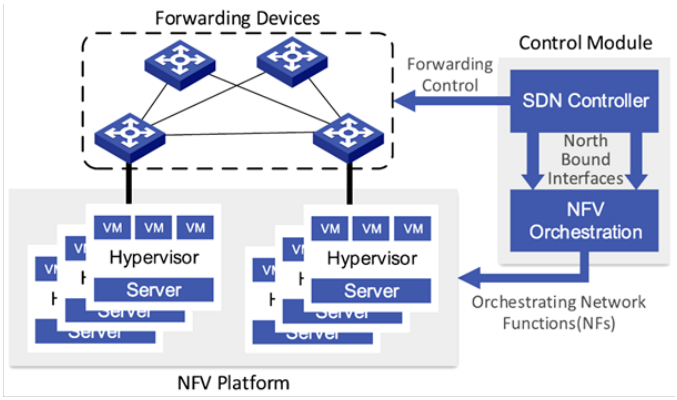

Fig. 3. SDN-based NFV Architectures

NFV platform uses suitable servers to implement NFs with high bandwidth and low cost. Virtual machine hypervisors run on servers to support virtual machines that execute network functions. This platform allows for customizable and programmable data layer processing functions, including firewall middleboxes, IDSs, and proxies that run as software on virtual machines. Network functions are delivered to the network operator in pure software.

SDN controller and NFV orchestration form the logic control module. The NFV synchronization system provides VNFs and is controlled by standard interfaces by the SDN controller. After providing the network topology and policy requirements, the control module calculates the optimal allocation of the functions (assigning network functions to some VMs) and converts the logic policy specification into optimal routing paths. The NFV orchestration system implements the assignment of the function, and the controller directs the traffic through them by installing the forwarding rules on the required and appropriate sequences of VMs and forwarding devices.

D. Related Works

Research [1] represents a perspective involving several stakeholders, including the user, service provider, and infrastructure provider. Therefore, VNF placement is considered to maximize the number of accepted requests from the set of received requests and maximize the satisfaction of the subscribers. The model presented in this study also differentiates service requests by prioritization levels and ensures that service quality objectives are met for accepted service requests.

The Jasper method [2] is an automated approach to simultaneously scaling, locating, and routing network services to minimize constraints' violations such as CPU memory and link capacity limitations. Considering a set of secondary constraints like total latency, resource consumption, etc., the Jasper method is a Pareto optimization.

In [3], optimizing the dynamic placement of network functions and flow routing in a chain of network functions is considered. In order to maximize the acceptable flow rate and minimize the energy cost for several service chains, a multi-objective optimization problem is formulated as a complex integer linear programming (MILP) problem and proves to be NP-hard.

[4] formulates the path optimization problem to minimize the number of routing rules for when a chain's service functions are located within a single virtual machine. It does not take into account the processing order specified in the service chains.

[5] Formulates the problem of accommodating more streams in a domain, while minimizing the use of links and CPUs, and proposes several exploratory methods. This approach aims to maximize the resource capacity of each link and CPU, rather than maximizing acceptable flow rates.

Some studies, such as [6], consider both placement and flow routing but address them separately. For example, a heuristic method is used for placement, and then the result is used as an input to guide the flow.

Research [7] uses an online and offline formulation. The main focus of offline formulation is to limit the number of transmission rules due to limitations in the TCAM memory of SDN switches. The online formulation is available for online load balancing on existing switches. This study does not examine the possibility of a dynamic service launch.

In [8], the optimal positioning of VNF chains is investigated, and it is shown that the problem can be complete NP-complete for particular cases. [9] examines the placement and routing of a network of VNFs, focusing on minimizing the link capacity of the network used. In [10], the authors provide an NFV network model suitable for 24 ISP operations. They define the VNF chain routing optimization problem and provide a linear integer linear programming formulation.

[11] Creates a reliable, low-latency routing optimization framework called READ for NFV data networks. READ involves formulating a MILP that Leads to VNF localization and traffic routing while maximizing the reliability of network-supported services.

In [12], the problem of locating virtual network functions and routing in physical hosts is studied to minimize the overall

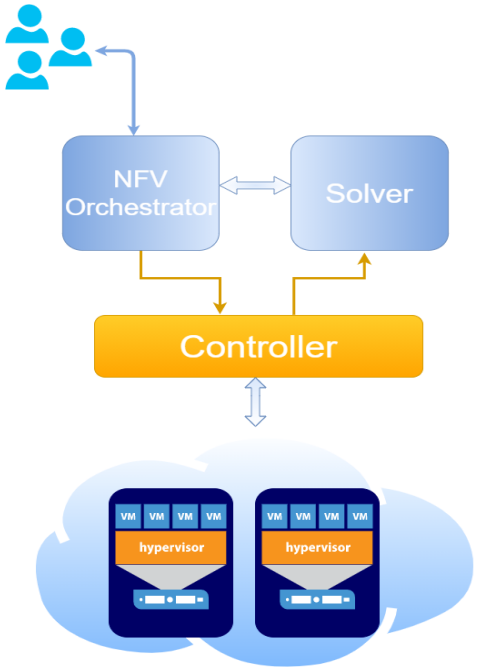

Fig. 4. System Model Architectures

latency defined as queue latency in network links. In this regard, this study considers both VNFs placement and the flow routing aspect.

In [13], the effects of changes in intermediate box traffic are studied, and solutions for SDN-based intermediate boxes placement are proposed to achieve the desired load balance. The problem of Traffic-Aware Middlebox Placement (TAMP) is formulated as an optimization problem to minimize the maximum link load ratio.

III. System Model

The primary purpose of this section is to indicate the required application blocks and provide a platform for implementing VNF placement and routing algorithms. Tree main parts of the structure are introduced that interact with each other during VNFs' placement and implementation of a service chain. Figure 4 shows this structure. These parts include controller, NFV orchestrator, and solver entity.

The *controller* unit is built to act as a centralized observer and has a network-wide view. This unit maintains and retrieves the status and performance of servers and network links to provide resource capacities (e.g., server CPU and memory, link bandwidth, etc.) to the resource optimizer. The controller also receives the VNF deployment result from the NFV synchronizer to provide suitable servers for the VNFs and create and update switching tables according to the VNF placement result. OpenFlow is a standard protocol used to configure flow tables for infrastructure network resources.

The *NFV orchestrator* unit is responsible for the overall management of the life cycle of VNFs. In order to meet the needs of users, examples of services are provided that can include switching elements, firewalls, and web security, intrusion detection and prevention systems, functions in home routers, traffic analysis functions, security, and key distribution functions. This component updates the services presented to customers with the provided NFs and reciprocally retrieves information about the services selected by customers.

The coordinator needs to communicate with the *Solver* component to ensure that VNFs are configured, located, and chained. This component has the task of selecting the optimal server locations in the cloud infrastructure based on the services to be deployed. It implements appropriate algorithms for placing selected VNFs. To ensure optimal placement of VNFs in the service chain, the solver needs up-to-date resource status information, including existing cloud resource capacity and VNF requests as input. Next, the solution to the VNF placement problem is returned to the NFV orchestrator.

The cloud resource infrastructure in the system model is shown as an undirected weighted graph and is denoted by $G = (V; I)$, where V represents the set of nodes and E represents the set of available links. The nodes consist of two general categories: 1) servers $N \subset V$ used to host VNFs as virtual machines, and 2) switches $X \subset V$ that chain VNFs between geographically dispersed servers. Hence $X \cup N = V$.

Each $v \subset V$ node is assigned a vector of available resource capacities (e.g., CPU memory and storage for servers and virtual LANs and current inputs for switches). Here, the vector contains the CPU attribute. Also, each link $(u; v) \subset E$ has an available bandwidth capacity denoted by $bw(u; v)$.

The placement of VNFs includes allocating VNFs of a service chain to the appropriate servers of the cloud infrastructure. Here we specifically, consider a service chain containing k VNFs that need to be set up to start a specific function.

In addition, each VNF in a service chain transfers the traffic to the next VNF of the chain, creating connections between successive elements of the chainset that include the input and output demands between the VNFs. Each VNF in the service chain is also identified by a vector of computational requirements named D (CPU, memory, storage, etc.) that must be satisfied. The size of the claims is proportional to the type of VNF.

A. Problem Statement

In this section, the VNF placement and routing processes performed by the solver component are presented and modeled as a Mixed Integer Linear Programming (MILP). To solve the MILP, a multi-constrained objective function is considered.

It is assumed that a service chain S consisting of a prespecified set of VNFs is admitted to the network. The service chain's VNFs are represented as network nodes and thus complete the network graph. It is also considered that each VNF of the S chain denoted by s_k is connected to all available servers $n \in N$. Thus, the new augmented graph of the network is shown as $G' = (V', E')$, in which $V' = s_k \cup V$ and $E' = (s_k, n) \cup E, n \in N$. The connection between two consecutive VNFs in the service chain is interpreted as a flow starting from VNF k, passing through the augmented graph, and finally entering VNF $k + 1$.

Problem variables:

$h_{uv}^{s_k,s_{k+1}}$: is a binary variable that if the flow between VNFs k and $k+1$ is routed through the link (u,v), it will be equal to one; otherwise, it will be zero.

$f_{uv}^{s_k,s_{k+1}}$: indicates the amount of traffic that passes between the VNFs k and $k+1$ through the link $(u,v) \in E'$.

Now, the cost function for the MILP can be defined.

1) Objective Function: To model the problem of optimizing VNF placement and routing, an appropriate objective function must be defined. As mentioned earlier, NFV has been introduced as a technique to reduce operating costs. Therefore, the objective function of the problem can be of the cost type.

Each server comes with the cost of activating the server to place VNFs, which is shown by $R_n, n \in N \subset V$. In addition, when the service chain is placed between multiple servers, the use of links connecting the servers also comes with an activation cost, which is specified by $R_{(u,v)}, (u,v) \in V$.

To define the cost function, one might only consider the interests of the Telecommunications Service Provider (TSP), or take the interests of the Infrastructure Provider (InP) into accounts, too.

The (InP) implements and manages physical resources in the form of data centers and physical networks. These physical resources are essential to provide virtual resources and will be leased to one or more TSPs through programming interfaces. Examples of InPs can be public data centers such as Amazon or private servers owned by TSPs. If a particular InP is idle, in whole or in part, to provide resources for a particular TSP, negotiations and consequently alliances with other InPs can be formed to provide multi-domain VNFs. InPs can also determine the mechanisms through which the resources are allocated to TSPs.

The TSP leases resources from one or more InPs that use the resources to set up VNFs. They also define a chain of these functions to create services for end-users. In a more general case, TSPs may lease their virtual resources to other TSPs. In such cases, the vendor TSP plays the role of an InP. In cases where InP is private or home, such as when provided by network nodes or TSP servers, InP and TSP may be an entity. An adequate cost function to VNFs placement and routing problem must measure the solution's efficiency both in terms of cloud resource usage and the service provider's cost.

From the service provider's point of view, the ultimate goal is to rent as few resources as possible and minimize the sum of resources activation costs. On the other hand, the owner of the cloud infrastructure must use the available computing and network resources optimally, using a proper balancing method, to maximize its income in the long run. Utilization of these resources must be done optimally to meet the needs of the cloud provider. A suitable load balancing scheme can be presented by using less loaded computing and network capacities through monitoring the available capacity (for example, through the controller).

Therefore, if $C(n)$ represents the capacity of server n and $bw(u,v)$ represents the bandwidth of link (u,v), two coefficients are introduced in Eq. 1 that are equal to the inverse values of the existing node capacity and the link bandwidth. In this way, resources with lower capacities are selected for VNF placement, which allows the cloud provider to make use of the available resources in a balanced manner.

$$A_n = \frac{1}{C(n)}, B_{(u,v)} = \frac{1}{bw(u,v)} \tag{1}$$

The objective function of Eq. 2 aims to minimize the cost of activating the links and servers selected for the service chain placement and flow routing, taking into account both the benefits of the service provider and the benefits of the infrastructure provider. Weights A_n and $B_(u,v)$ are also included to ensure load balance between available sources. These coefficients are eliminated if only the benefit of the service provider is considered.

$$\sum_{k\in K}\sum_{n\in N}\sum_{s_p\in S} A_n.\mathbf{R}_n.h_{s_p n}^{s_k,s_{k+1}} + \sum_{k\in K}\sum_{(u,v)\in E} B_{(uv)}.\mathbf{R}_{(uv)}.h_{uv}^{s_k,s_{k+1}} \tag{2}$$

2) Constraints: Eq. 3 ensures that the total capacity requested by VNFs, denoted by $D(s_p)$, does not exceed the capacity of the selected servers $C(n)$.

$$\sum_{s_p\in S} D(s_p).h_{s_p n}^{s_k,s_{k+1}} \leq C(n), \quad \forall n \in N, s_k, s_{k+1} \in S \tag{3}$$

In Eq. 6, it is guaranteed that the maximum congestion on each network link does not exceed a threshold. The queue formed at each link is considered to be $M/M/1$. In M/M/1 queue, congestion is obtained from equation $\lambda/\mu - \lambda$, in which λ represents the rate of entry into the queue and μ is the rate of processing. If the maximum tolerable congestion on each link is considered to be equal to the buffer size of that link, $bfr(u,v)$, then the inequality in 4 must be satisfied for each link.

$$\frac{\lambda}{\mu - \lambda} < bfr(u,v) \tag{4}$$

Comparing 4 with the formulation of the present problem, it is concluded that the bandwidth of the link (u,v), denoted by $bw(u,v)$, is equal to the processing rate of that link. Also, the entry rate for each link (u; v) is obtained from Equation 5.

$$\sum_{k\in K}(f_{uv}^{s_k,s_{k+1}} + f_{vu}^{s_k,s_{k+1}}) \tag{5}$$

By replacing the problem equivalents with values in 4, the maximum congestion on each link will be in the form of 6.

$$\sum_{k\in K}(f_{uv}^{s_k,s_{k+1}} + f_{vu}^{s_k,s_{k+1}}) \leq \frac{bfr(u,v)}{1+bfr(u,v)} bw(u,v), \quad \forall (u,v) \in E' \tag{6}$$

Flow conservation is satisfied through the set of constraints 7, 8, and 9. These constraints represent the sum of the input and output flows for intermediate, destination, and source nodes, respectively.

$$\sum_{w \in V'} f_{uw}^{s_k,s_{k+1}} - \sum_{w \in V'} f_{wu}^{s_k,s_{k+1}} = 0, \forall k \in K, \forall w \in V' \backslash s_k, s_{k+1} \quad (7)$$

$$\sum_{w \in V'} f_{s_k w}^{s_k,s_{k+1}} - \sum_{w \in V'} f_{w s_k}^{s_k,s_{k+1}} = bw(s_k, s_{k+1}), \forall k \in K \quad (8)$$

$$\sum_{w \in V'} f_{s_{k+1} w}^{s_k,s_{k+1}} - \sum_{w \in V'} f_{w s_{k+1}}^{s_k,s_{k+1}} = -bw(s_k, s_{k+1}), \forall k \in K \quad (9)$$

Equation 10 ensures that flows are not splittable.

$$f_{uv}^{s_k,s_{k+1}} + f_{vu}^{s_k,s_{k+1}} = bw(s_k, s_{k+1}).h_{uv}^{s_k,s_{k+1}}, \forall k \in K, (u,v) \in E' \quad (10)$$

Equation 11 ensures that each VNF must be assigned to one and only one server.

$$\sum_{n \in N} h_{s_p n}^{s_k,s_{k+1}} = 1, \forall s_p \in S, k \in K \quad (11)$$

Finally, constraint 12 ensures that the placement of VNFs forms a connected graph [14].

$$h_{uv}^{s_k,s_{k+1}} = h_{uv}^{s_{k+1},s_{k+2}}, k \in K, (u,v) \in E' \quad (12)$$

IV. Methods and simulations

As mentioned in Chapter 3, switches, servers, and VNFs of the service chain are considered as network nodes, and it is assumed that there is a link between all switches and servers and all servers and VNFs in the service chain. As a result, the network graph will be presented in Figure 5.

This figure shows two switches, three servers, and a service chain containing five VNFs. It should be noted that to analyze the system's scalability in some simulations while maintaining the overall structure, the number of network servers may change. It is also assumed that the order of VNFs within the service chain is prespecified and fixed, and the service chain's admission to the network has already taken place.

For each server and network link, a certain amount of CPU capacity and bandwidth are assigned, respectively, which bandwidth amounts are shown as the weight of the links in the network topology graph. The link's bandwidth connecting the switches is significantly higher than that of the links connecting servers and switches.

MILP formulations provide a flexible and accurate mathematical method for formulating general network problems. However, MILP problems, especially for large-scale experiments, are computationally untraceable. Therefore, this solution is more suitable for smaller-scale experiments, where fewer computational requirements are required.

Conclusions of NP-hardness of a problem in theoretical computer science makes heuristic methods the only currently acceptable option for various complex optimization problems that must be frequently solved in real-world applications. Therefore, the optimization problem raised in the previous chapter is compared with the existing well-known optimization problems, and some of the existing heuristic solutions to deal with this problem are stated.

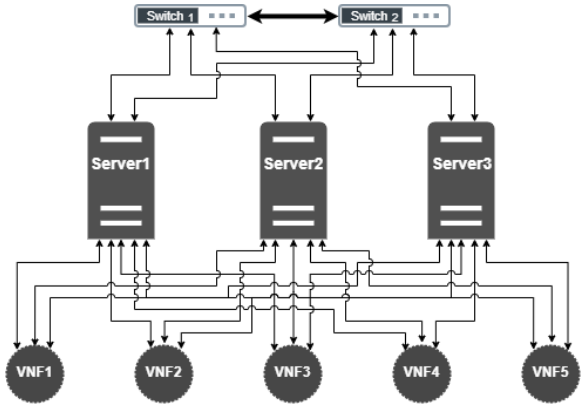

Fig. 5. Infrastructure Graph

The Bin Packing Problem (BPP) is a famous discrete optimization problem with many applications and naturally fits in with many problems, such as bandwidth allocation. In BPP, objects of different volumes must be placed in a finite number of bins of the same capacity to minimize the number of bins used. One type of BPP is the BPP with variable sizes, or VSBPP, which is conventionally defined as a set of objects with specified weights and several bins of different types. Each type of bin consists of identical bins with a specified capacity and fixed cost. The purpose of VSBPP is to place each object in a bin so that the sum of the weights in that bin does not exceed the bin's capacity, and the total cost is minimized. As mentioned, BPP is NP-hard, and consequently, VSBPP will be untraceable. VSBBP-related research background is relatively low compared to BPP.

The shortest path problem is one of the most fundamental network optimization problems. This problem has arisen in practice and is considered a sub-problem in many network optimization methods. The shortest path problem is about finding the path between two nodes so that the edges' total weight that makes it up is minimized. This problem is used to find routes between physical locations, such as traffic routes on Internet maps such as Google Maps. The most well-known methods to solve this problem are the Belman-Ford method, Floyd-Warshall method, Viterbi method, and finally, the Dijkstra method, which is among the most important methods for solving the shortest path problem.

By carefully examining the model system proposed in this research, it can be said that the main problem consists of two sub-problems. The first sub-problem is the location of VNFs on network servers, and the second sub-problem is routing among these VNFs.

Comparing the first sub-problem with VSBPP, it is clear that each server is in the role of a box, and VNFs are the same objects. Are in VSBPP. As a result, placing objects inside the box will be equivalent to placing VNF on network servers. Then, by comparing the second sub-problem with finding the shortest path, the similarity of these two problems is revealed. In this way, a path with the lowest cost must be established between two consecutive VNFs in the service chain that have been deployed. The ultimate goal is to minimize the total cost of using the servers and links in the network structure.

A. Heuristic Method

In this part, a heuristic method to solve the problem of placing and routing through VNFs is presented. The Best First Decreasing (BDF) heuristic method is used in the following, which provides good performance for the classic BPP. Each new object is placed in the busiest place with enough space to accommodate it in this method. This method is an online method. Since all locations are examined in each step, the method's execution time will be of order $O(n^2)$.

In the method presented for VSBPP, VNFs are first sorted in ascending order of size and then positioned one by one. For each VNF, it first tries to be assigned to the best server ever opened. To determine the best server, a standard function is defined based on the considerations of the objective function of the initial optimization problem. The best server in this method is a server that maximizes the standard function that calculates the empty capacity of servers. If VNF cannot be placed on one of the preselected servers, then a new server is selected, and VNF is placed on it.

An issue specific to VSBPP that does not appear in classic BPP with similar bins is what criteria to use when selecting a new bin when needed. Here, new servers are selected in a non-descending order of cost per unit of capacity. Selecting servers with these criteria usually leads to good results, but when locating the latest VNs, performance may decline slightly, even if the cost-to-capacity ratio is good. That way, a new server has to be selected and opened for one of the end-of-list VNFs. The selected server may have much more capacity than the few VNFs left to take advantage of this capacity [15].

In this case, a good choice could be a server with a higher cost-to-capacity ratio but a lower total cost. As a result, a control step can be added to the method after initial processing, improving the method response by evaluating such possible exchanges between servers. This step alternately examines each of the selected servers. Then, if there is an unselected server with a lower total cost and a capacity greater than or equal to the capacity used by the selected server, it will transfer the VNFs placed on the previous server to the new server.

In this heuristic method, after locating all VNFs, the Dijkstra method is used to communicate between them. The pseudo-code of this method is 1.

Algorithm 1 Adapted BFD

Input S : Set of VNFs to be accommodated into the servers
Input N : Set of Servers available to load the VNFs

Sort the VNFs in S according to non-increasing order of their volumes
Sort the servers in N according to non-decreasing order of ratio $A_n R_n / C_n$
$\bar{S}$: Set of unpacked VNFs
K : Set of selected bins
Set $\bar{S} = S$
K = $\{\emptyset\}$
while $\bar{S} \neq \emptyset$ **do**
  if the first VNF in $\bar{S}$, naming s, can be accommodated into a server in K **then**
    Accommodate s into the best server of the set K , naming n
  else
    Accommodate s into n' , where n' is the first server in the ordered list $N \backslash K$
    $K := K \cup n'$
    $\bar{S} := \bar{S} \backslash s$
  end if
end while
for all $k \in K$ **do**
  for all $m \in N \backslash K$ **do**
    $U_k = \sum_{s\ loadedin\ k} D_s$
    if $C_m > U_k$ and $A_m R_m < A_k R_k$ **then**
      Move all the items from k to m
      $K = K \backslash k \cup m$
    end if
  end for
end for
if $\bar{S} = \emptyset$ **then**
  for all $s_k \in S$ **do**
    if selected servers of s_k and s_{k+1} are different **then**
      $Dijkstra(s_k, s_{k+1})$
    end if
  end for
end if

B. Simulation Results

To run the simulations, a computer operating Windows 10 with a corei7 Intel processor and 16 GB of memory was used. The proposed algorithms were implemented in MATLAB, and Mosek Toolbox was used for MILP.

In general, each simulation was repeated a hundred times to extract each diagram, and then their results were averaged over time. Some evaluation criteria are used to compare the performance of the proposed heuristic algorithm with the optimal solution.

Figure 6 shows the average cost of allocating service chains based on the number of service chains admitted to the network. It can be seen that for both algorithms, as the number of service chains accepted in the network increases, the total cost

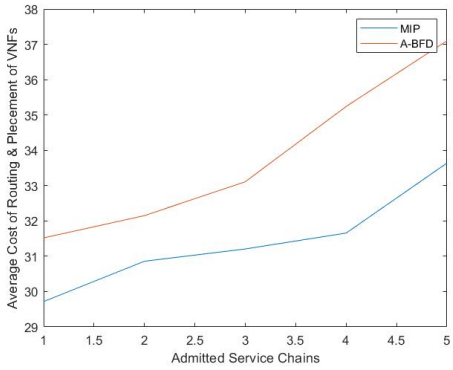

Fig. 6. Impact of increasing the number of admitted service chains on the total cost

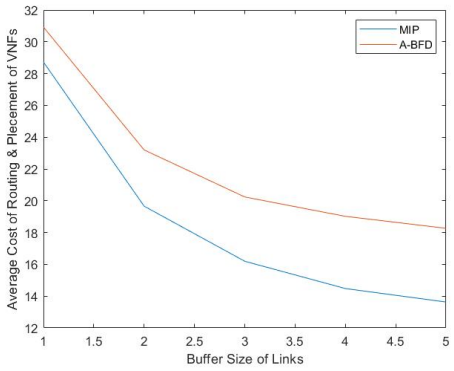

Fig. 7. Impact of increasing the links' buffer size on the total cost

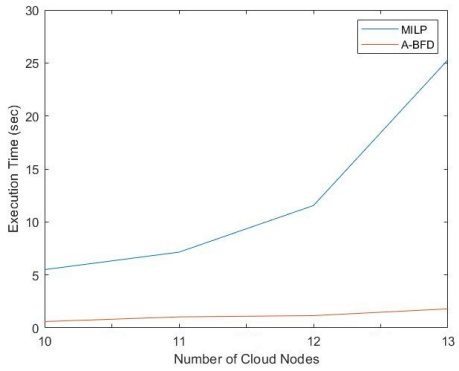

Fig. 8. Impact of increasing the number of VNFs in the service chain on the execution time

also increases. As expected, the MILP algorithm offers the optimal solution at a lower cost.

The figure 7 shows the effect of increasing the buffer size of network links on the average total cost of placing and routing network functions. Here it is assumed that there is one admitted service chain within the network. It can be seen that both heuristic and optimal algorithms show cost reduction behavior while increasing links' buffer size. At the same time, the optimal method at any point in the graph has a lower total cost than the heuristic algorithm. The buffer size of each link determines the amount of congestion that can be tolerated by that link. The reason for the cost reduction is that by increasing the buffer size, the link will withstand higher congestion. By increasing the congestion tolerance of each link, the increase in buffer size affects the usage of each network link. It ultimately leads to a reduction in the total cost of placing and routing VNFs.

In Figure 8, algorithms scalability is evaluated by examining the execution time of the algorithm based on increasing the number of nodes in the cloud infrastructure. The increase in execution time is because each algorithm examines all possible solutions in the search space, leading to the execution of a large number of operations. This increase in time for the heuristic algorithm is much less than the MILP method, making this algorithm scalable and efficient when developing a network. It should be noted that the MILP's execution time will grow exponentially as the size of the cloud infrastructure increases and may not even be scaled on the same axes as other approximate algorithms. It shows that the MILP algorithm can not be used in online and large-scale experiments but only for small-scale and mainly laboratory uses.

V. Conclusion

The advent of two new technologies, SDN and NFV, has fundamentally changed the use and architecture of the network. These two technologies promise mobile operators to reduce costs, increase flexibility, increase scalability, and reduce the time to launch new applications and services. Considering the benefits these technologies offer, mobile operators are gradually changing how they design their mobile networks to keep pace with growing data traffic, new devices, and network access.

In the presented report, concepts of SDN and NFV were briefly introduced. Then the problem of orchestration or the management of network functions' placement and routing, with SDN management, was raised and modeled as an optimization problem. In particular, a heuristic method to the problem was proposed, and its performance was analyzed and compared with the optimal solution for small scale networks. According to the results of this study, topics such as including various QoS considerations to achieve the desired quality of service and the use of statistical characteristics of entry and exit of service chains to the network are suggested for future researches.

References

[1] Paola Cappanera, Federica Paganelli, and Francesca Paradiso. Vnf placement for service chaining in a distributed cloud environment with multiple stakeholders. *Computer Communications*, 133, 10 2018.

[2] Sevil Dräxler, Holger Karl, and Zoltán Ádám Mann. Jasper: Joint optimization of scaling, placement, and routing of virtual network services. *IEEE Transactions on Network and Service Management*, 15(3):946–960, 2018.

[3] Insun Jang, Dongeun Suh, Sangheon Pack, and György Dán. Joint optimization of service function placement and flow distribution for service function chaining. *IEEE Journal on Selected Areas in Communications*, 35(11):2532–2541, 2017.

[4] Andrey Gushchin, Anwar Walid, and Ao Tang. Scalable routing in sdn-enabled networks with consolidated middleboxes. 04 2015.

[5] Ali Mohammadkhan, Sheida Ghapani, Guyue Liu, Wei Zhang, K. K. Ramakrishnan, and Timothy Wood. Virtual function placement and traffic steering in flexible and dynamic software defined networks. In *The 21st IEEE International Workshop on Local and Metropolitan Area Networks*, pages 1–6, 2015.

[6] Ying Zhang, Neda Beheshti, Ludovic Beliveau, Geoffrey Lefebvre, Ravi Manghirmalani, Ramesh Mishra, Ritun Patneyt, Meral Shirazipour, Ramesh Subrahmaniam, Catherine Truchan, and Mallik Tatipamula. Steering: A software-defined networking for inline service chaining. In *2013 21st IEEE International Conference on Network Protocols (ICNP)*, pages 1–10, 2013.

[7] Zafar Qazi, Cheng-Chun Tu, Luis Chiang, Rui Miao, Vyas Sekar, and Minlan Yu. Simple-fying middlebox policy enforcement using sdn. volume 43, pages 27–38, 09 2013.

[8] Selma Khebbache, Makhlouf Hadji, and Djamal Zeghlache. Virtualized network functions chaining and routing algorithms. *Computer Networks*, 114:95–110, 2017.

[9] Bernardetta Addis, Dallal Belabed, Mathieu Bouet, and Stefano Secci. Virtual network functions placement and routing optimization. pages 171–177, 10 2015.

[10] Bernardetta Addis, Dallal Belabed, Mathieu Bouet, and Stefano Secci. Virtual network functions placement and routing optimization. In *2015 IEEE 4th International Conference on Cloud Networking (CloudNet)*, pages 171–177, 2015.

[11] Long Qu, Chadi Assi, Khaled Shaban, and Maurice 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*, 14(3):554–568, 2017.

[12] Racha Gouareb, Vasilis Friderikos, and Abdol-Hamid Aghvami. Virtual network functions routing and placement for edge cloud latency minimization. *IEEE Journal on Selected Areas in Communications*, 36(10):2346–2357, 2018.

[13] Wenrui Ma, Jonathan Beltran, Zhenglin Pan, Deng Pan, and Niki Pissinou. Sdn-based traffic aware placement of nfv middleboxes. *IEEE Transactions on Network and Service Management*, 14(3):528–542, 2017.

[14] Aris Leivadeas, Matthias Falkner, Ioannis Lambadaris, and George Kesidis. Optimal virtualized network function allocation for an sdn enabled cloud. *Comput. Stand. Interfaces*, 54(P4):266–278, November 2017.

[15] T. Crainic, G. Perboli, Walter Rei, and R. Tadei. Efficient heuristics for the variable size bin packing problem with fixed costs. 2010.