Network Function Virtualization-Aware Orchestrator for Service Function Chaining Placement in the Cloud

Hassan Hawilo, *Member, IEEE*, Manar Jammal, *Member, IEEE*, and Abdallah Shami, *Senior Member, IEEE*

***Abstract*—Network function virtualization (NFV) has been introduced by network service providers to overcome various challenges that hinder them from satisfying the growing demand for networking services with higher return-on-investment. The association of NFV with the leading technologies of information technology virtualization and software defined networking is paving the way for flexible and dynamic orchestration of the VNFs, but still, various challenges need to be addressed. The VNFs instantiation and placement problems on data center's (DC) servers are key enablers to achieve the desired flexible and dynamic NFV applications. In this paper, we have addressed the VNF placement problem by providing a novel mixed integer linear programming (MILP) optimization model and a novel heuristic solution, Betweenness centrality Algorithm for Component Orchestration of NFV platform (BACON), for small- and large-scale DC networks. The proposed solution addresses the VNF placement while taking into consideration the carrier-grade nature of the NFV applications and at the same time, minimizing the intra- and end-to-end delays of the service function chain (SFC). Also, the proposed approach enhances the reliability and the quality of service (QoS) of the SFC by maximizing the count of the functional group members. To evaluate the performance of the proposed solution, this paper conducts a comparative analysis with an NFV-agnostic algorithm and a greedy-k-NFV approach, which is proposed in the literature work. Also, this paper defines the complexity and the order of magnitude of the MILP model and BACON. BACON outperforms the greedy algorithms especially the greedy-k-NFV solution and has a lower complexity, which is calculated as $O((n^3 - n^2)/2)$. The simulation results show that finding an optimized VNF placement can achieve minimal SFCs delays and enhance the QoS accordingly.**

***Index Terms*—Network function virtualization, network softwarization, cloud computing, mobile computing, service function chain, high availability, next generation network, next generation mobile networks, 5G mobile communication.**

I. Introduction

THE demand for high-bandwidth network connectivity has been growing significantly over the past few years. It has gained further momentum with the surge in the number of internet-connected mobile devices ranging from smartphones, tablets, laptops to sensor networks and Machine-to-Machine (M2M) connectivity. The network traffic has exceeded the capacities of the existing mobile service providers' networks [1]. Since the network traffic is expected to increase in the near future, Network Service Providers (NSPs) should invest in bandwidth-oriented infrastructure to satisfy the demand [2]. While studies show that the return-on-capital with such investments is minimal [3], the network upgrading highly depends on the network infrastructure. This dependency along with the exponential cost of the network equipment may lessen the revenue margins of the NSPs when an upgrade or new service is released. NSPs' challenges are not only bounded to the cost of expensive hardware devices, but they are also affected by the increase in the energy costs coupled with the shortage of personnel with expertise to design, implement, and orchestrate a progressively complex hardware-based infrastructure. Moreover, maintenance of the network infrastructure is another primary concern of the service providers. The scope of these issues is not limited merely to the revenue loss but also to the ripple effects that manifest through lags in time-to-market as well as in the general hindrances to innovation within the telecommunications industry. Therefore, network operators seek to reduce or even forfeit their dependency on proprietary hardware.

To achieve these targets, network service providers are investigating the integration of virtualization technology within the telecommunications industry. Virtualization technology emerges as a mean for Information Technology (IT) specialists to enhance the capital investments with higher returns-on-capital. Virtualization also facilitates the hardware and software decoupling process where multiple isolated software programs can share the underlying hardware [4]. As an initial step, a group of seven telecommunication operators established an industry specification group for Network Function Virtualization (NFV) under the European Telecommunications Standards Institute (ETSI). Once they proposed their solution in October 2012, several telecommunication equipment providers and IT specialists subsequently have joined the group [5].

NFV is the concept of migrating the network functions from dedicated hardware equipment to software-based applications. NFV is the technology that can exploit the advantages of the IT virtualization evolution. Equipment and software components are consolidated on standardized IT platforms (e.g., high volume servers, switches, and storage) while network functions within the proprietary hardware can be simultaneously decoupled. Through NFV, Virtual Network Functions (VNFs) can be instantiated at various locations, such as

Manuscript received September 12, 2018; revised January 6, 2019; accepted January 11, 2019. Date of publication January 31, 2019; date of current version February 14, 2019. *(Corresponding author: Hassan Hawilo.)*

The authors are with the Department of Electrical and Computer Engineering, Western University, London, ON N6A 3K7, Canada (e-mail: hhawilo@uwo.ca; mjammal@uwo.ca; abdallah.shami@uwo.ca).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JSAC.2019.2895226

Data Centers (DCs), network nodes, and end users' premises depending on the network requirements [3]. Exploiting the advantages of the cloud computing services, Software Defined Networking (SDN), and NFV facilitates the opportunity to design and implement scalable, elastic, and programmable next-generation networks [6], [7]. However, the latter desired networks introduce various deployment and orchestration challenges that should be resolved to realize their benefits and pave the way for wider commercial adoption by the industry [8], [9].

ETSI defines the basic architecture standards for the NFV Management and Orchestration (NFV-MANO) framework. Each NFV networking service consists of one or more VNF [10]. VNFs implement various functionalities that provide the networking services defined by the Network Service Descriptor (NSD). According to the NSD VNF Forwarding Graph (VNFFG), the logical path connecting the VNFs is defined as a Service Function Chain (SFC). Having well-defined standard interfaces for the VNFs provides the NSPs with the freedom to design and implement their proprietary services to meet the customers' needs while avoiding vendor lock-in of their NFV platforms. Moreover, it drives the innovation and evolution of the NFV networking services and provides the capability of flexible management and orchestration of the VNFs lifecycle based on functional/non-functional constraints.

Despite all the significant literature studies on NFV, VNFs deployment and orchestration still need to be further investigated and exploited to satisfy the carrier-grade requirements for the networking services [11]–[14]. Researchers have been addressing various aspects of NFV challenges. For instance, VNFs orchestration and management challenges have been addressed in many literature studies [15]–[22]. They propose different optimization models and heuristic solutions for managing the VNFs placement problem. Besides, other researchers direct their efforts to realize the development of the NFV management platforms [23]–[26]. However, the literature studies discard the fact that the VNFs are running as software applications on commodity servers that provide them not only with the flexibility and programmability of a distributed software application but with the benefits of the microservices architecture as well. Although the majority of the research projects have considered the carrier-grade nature of the NFV, their solutions do not reflect the carrier-grade requirements of cloud-based application, such as performance, fault resilience, high availability, scalability, QoS, VNF Components (VNFCs) structure, and governments' geo-restrictions [27]–[30]. VNFs are the building block of NFV and are constructed by chaining various VNFCs to provide the desired services. The VNFCs take advantage of microservices architecture and the emerging implementation of Service-Oriented software Architecture (SOA). Each VNFC is foreseen as a microservice by itself, which enables heterogeneous VNF structures and allows more flexibility in terms of hosting environment and manageability. However, the intra-connections of VNFCs are directly affected by their placements, which affect and define the performance of a VNF service. Moreover, the interconnections of the VNFs that represent the logical container of the VNFC are directly affected by the VNFs' logical placements, which in return affect the service chain performance. With this in mind, VNFs' placement and service chaining are still important challenges that need further investigation to achieve the anticipated benefits of NFV, such as lower Operation and Capital Expenditure (OPEX and CPEX), on-demand scaling, and real-time network programmability while satisfying the above carrier-grade requirement.

To address the inadequacies of VNFs placement and SFCs orchestration, this paper introduces a novel VNF placement orchestration using a Mixed Integer Linear Programming (MILP) optimization model and associates it with a heuristic solution, Betweenness centrality Algorithm for Component Orchestration of NFV platform (BACON). The VNF placement orchestration is based on capturing all the carrier-grade requirements of an NFV application, such as the functionality, latency, and availability constraints. The main objective of the orchestration is finding the VNFs placements that satisfy the functional and non-functional constraints while minimizing the intra-communication delays between the VNF instances and enhancing the Quality of Service (QoS) of the computational path (SFC). The main contributions of this work can be summarized as follows:

i) Propose an intelligent orchestrator that selects the best placement for the VNFs in a given NFV application to minimize the intra-communication delays between the VNF instances and enhance the QoS of the computational path (SFC). The optimized placement achieves higher number of VNF instances participating in a service chain with different serving components. This outcome generates more active redundant computational paths that can be optimally used to achieve the desired QoS in terms of performance and high availability of service chains per request.
ii) Capture the carrier-grade's functionality constraints that affect the SFCs of the NFV application, such as the application's availability.
iii) Capture the VNFs' dependencies constraints to generate a successful interacting SFCs.
iv) Minimize the end-to-end delay of the SFC.

The rest of this paper is structured as follows. Section II presents the related work for the NFV placement approaches. Section III discusses the problem motivation. In Section IV, the problem formulation, modeling, and constraints are defined. Section V defines the optimization model of the proposed NFV placement approach and its computational complexity. Section VI presents the heuristic algorithm, BACON. Section VII presents the simulation setup, its results, and comparative analysis. Finally, Section VIII concludes the paper.

II. Background and Related Work

NFV is the technology that promises to revolutionize the telecommunication industry by providing substantial benefits to the next-generation networks. As NFV captures the interest of the leading telecommunication industrial equipment/service providers and academic researchers, intensive research projects are focusing on this technology.

Evolved Packet Core (EPC) is one of the basic network entities that are considered for virtualization. Taleb *et al.* [31] implement virtualized EPC (vEPC) using the cloud computing environment and demonstrate the feasibility of providing vEPC as a service. The authors also propose a comparative analysis of various architectures. Baba *et al.* [32] present and implement a vEPC architecture based on the VNFs. The architecture satisfies the requirements of the machine-to-machine service computing with reduced resources. The authors achieve 27% CPU time reduction with the proposed architecture. A smart VNF placement to deploy multi-tier cloud applications is proposed by PACE [33]. However, PACE overlooks many of the requirements that affect the VNF placement to achieve the desired QoS in multi-tier cloud-based applications. These requirements include the VNF dependency hierarchy, delay tolerance, and anti/co-location constraints. An efficient and scalable VNF provisioning framework is proposed in E2 [34]. E2 is a framework that manages the VNFs by combining traffic engineering and the best VNF placement. It is suitable for a private cloud that serves a single type of applications and provides specific functionalities, such as traffic offloading to proprietary switches. E2 has discarded the various placement constraints, such as the instances' inter and intra-dependency and the delay tolerance between components. Bari *et al.* [35] propose an optimization algorithm for the VNF placement with a simplified set of constraints. The latter only considers the deployment cost, the resources requirement, and the processing delay. This optimization algorithm discards the placement constraints that satisfy the carrier-grade requirements of the VNF applications, such as the VNF chaining, reliability, and delay tolerance constraints.

Mohammadkhan *et al.* [36] formulate a mixed integer linear programming optimization model for VNFs placement and traffic flow routing while minimizing the resource utilization. However, the proposed solution has focused on minimizing computational resources while ignoring non-functional constraints such redundancy, dependency, and availability. Sahel *et al.* [37] focus on the network service chaining problem by formulating an integer linear programming model and a heuristic algorithm. The proposed solution is based on two segments: a decomposition selection with backtracking phase and a mapping phase; leading consequently to sub-optimal solutions. Nguyen *et al.* [38] formulate a quadratic programming model and propose a heuristic solution for the VNF placement and routing problems. However, the latter does not consider the VNF chaining and dependencies in their solution. The authors also consider that the networking service is provided by one VNF. Gadre *et al.* [39] propose an agile VNF placement solution based on a divide-and-conquer algorithm. The formulation considers that the VNFs are hosted on network switches. Hosting VNFs on virtual switches could accelerate the processing of the user's service chain request, but it contradicts with the principles of SDN and NFV.

Eramo *et al.* [20] propose an integer linear programming model for VNF migration and placement that minimizes the total expenses and revenue loss. The proposed work has overlooked various constraints in there considerations, such as the delay tolerance and dependencies between components. Ahvar *et al.* [40] formulate an integer linear programming for the VNF placement to minimize the cost of the NSP. However, the proposed ILP has considered the resources constraints, such as the decision variables without including other functional and non-functional constraints. Gupta *et al.* [41] introduce "COLAP", a predictive framework to place the participating VNFs of a SFC in a cloud environment while optimizing the service latency. In summary, this work has considered the service latency as the main metric while overlooking the VNF instances' dependencies and availability metrics. Zhang *et al.* [42] formulate the VNF placement problem as bin-packing and open Jackson network problems to achieve better resource utilization. The proposed solution has considered computational utilization as main metric while ignoring non-functional constraints such redundancy, dependency, and availability. Ayoubi *et al.* [21] propose a cut-and-solve approach for the VNF placement problem. The approach consists of two sub-problems and maximizes the policy-aware traffic flows count. This work has considered the service chain latency as the main metric while overlooking the dependencies and availability metrics. Qu *et al.* [43] formulate a MILP model and a heuristic approach to overcome the scalability of an optimization model while maximizing the reliability and minimizing the SFC end-to-end delays. The authors have proposed an algorithm that selects a subset of VNFs that are needed to generate a SFC and its redundant. The user traffic in the proposed algorithm is managed through the main SFC and its redundant, simultaneously, which results in a costly SFC deployment. The redundant path has a longer SFC leading to higher delay and thus affecting the QoS, in the case of SFC request's migration or failure. Despite the high demand of resource allocation for the proposed algorithm, the authors have discarded the delay tolerance between components.

Hantouti *et al.* [44] have discussed SDN architectures for SFC and provided an analysis of the traffic steering techniques in the context SDN-based SFC approaches. The work has presented a comprehensive analysis while identifying relevant research challenges and classifying the traffic steering techniques according to their efficiency in real-life networks. Bagaa *et al.* [45] have proposed an algorithm to define the optimal number of core network virtual elements to meet the demand of the mobile traffic while maintaining the QoS and maximizing the profits of the cloud operators. Furthermore, the authors have developed an algorithm to place the core network virtual instance in a federated cloud. Benkacem *et al.* [46] have formulated a VNF placement algorithm to minimize the cost while maximizing the Quality of Experience (QoE) of the virtual streaming service. The authors have applied the bargaining game theory to achieve an optimal tradeoff between the cost efficiency and QoE in the proposed solution.

Laghrissi *et al.* [47] have addressed the problem of non-uniform distribution of signaling messages in irregular network topologies. They have proposed a solution to map the non-uniform distribution of signaling messages in the physical domain into a new uniform environment through the utilization of Schwartz-Christoffel conformal mappings. Taleb *et al.* [48] have proposed a VNF placement algorithm to cope with the

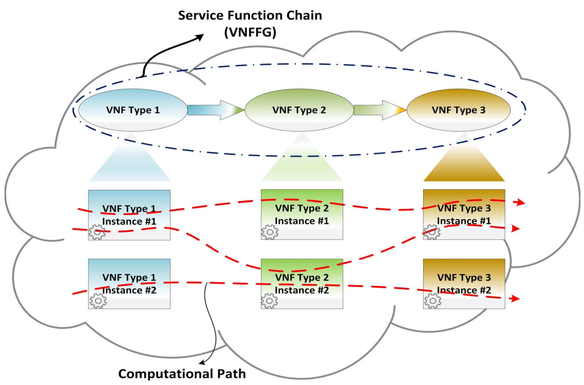

Fig. 1. Service function chain and computational path of NFV of different VNF types.

surging mobile traffic while minimizing the cost in terms of the total number of instantiated VNFs to build a Virtual Network Infrastructure (VNI) in a cloud environment. The proposed algorithms objective functions are minimizing path between users and their respective data anchor gateways and optimizing their sessions' mobility. Bagga *et al.* [11] proposed a placement algorithm for the mobile network functions over federated cloud. The proposed algorithm instantiate the Packet Data Network Gateways (PDN-GW) virtual instances and select the adequate virtual PDN-GWs for user equipment receiving specific application service. Laghrissi *et al.* [49] developed a tool that facilitates the development of spatio-temporal models of mobile service usage over a particular geographical area. Furthermore, the tool help in defining the mobile users' behavior in terms of mobility patterns and service consumption.

Most of the aforementioned approaches propose solutions through private cloud interfaces, which are completely owned and controlled by the cloud service providers. Also, the previous literature studies discard the fact that different applications can be hosted within the VNF entities in the cloud platform. So far, the proposed solutions for the NFV-SDN framework are mostly applicable to small-scale networks within a private cloud. Private clouds are groups of data centers owned by the network service providers. The latter has full control over the entire infrastructure (physical servers, underlying core networks, virtual environments, and orchestrators). These solutions overlook multi-tenant support and co-existence with variant applications that are already using the cloud. Additionally, most of the above literature studies have focused on the VNF functionalities and placements from the perspective of single-tier applications (services) where a single type of VNFs is responsible for serving the users' requests (traffic). However, most NFV applications (services) are multi-tier applications (services) where a set of different types of VNFs work collaboratively to serve users' requests (traffic).

Majority of the stated research has discarded various carrier-grade requirements, such as the performance, fault resilience, high availability, scalability, QoS, and governments' geo-restrictions. In order to achieve the desired objectives of NFV, further studies should be conducted on the VNF's functionalities and placements from the perspective of multi-tier applications orchestration while satisfying the carrier-grade requirements. To mitigate the above inadequacies and pave the way for advancing NFV, SFC realization, and wider adoption within NSPs, this paper proposes an intelligent VNF placement orchestrator. The latter proposes a MILP model and a heuristic solution, BACON, and satisfies various carrier-grade requirements of NFV platforms. The MILP model acts as a solver for small-scale NFV platforms and a benchmark for BACON that addresses large-scale NFV platforms.

III. Motivation

VNFs are hosted in a cloud environment where they are executed either within Virtual Machines (VMs) or within containers. The allocation of the VNFs' execution environment on the hosting servers in data centers directly affects the quality of service provided by these VNFs [50]–[53]. Therefore, having an optimal allocation for the VNFs is essential to satisfy the carrier-grade requirements.

A. VNF Placement Requirements

The ETSI defined framework does not provide a definition for the VNFs' placement management entity. Mainly, the mapping of the VNFs to their hosts is managed by the cloud service provider or is delegated to the users (VNFs' owners). Furthermore, NFV is associated with service function chains that are directly affected by the VNF placement. At the Infrastructure as a Service (IaaS) level, the cloud service provider may offer a certain level of guaranteed resources performance and availability of the VMs assigned to the tenants. However, this approach does not guarantee the QoS of the VNFs deployed on these VMs. In fact, tenants would have to deploy and manage their VNFs in an efficient manner to achieve the desired quality of service. Netflix utilization of the Amazon Web Services (AWS) is an example on how tenants deploy and manage their cloud applications to meet the QoS requirements [54]. Netflix has contributed to various open source software entities that integrate with AWS and other cloud services to enhance and achieve the desired quality of service.

VNF schedulers that are agnostic of the intricacies of the tenant's application may result in inefficient placements. In these placements, computationally chained VNF components may be placed where the delay constraints can be violated, which hinders the application's functionality in terms of scalability and traffic offloading. A carrier-grade-aware (NFV-aware) application architecture that defines the computational paths, the participating components (VNFs), and the prospected service function chains is needed to enhance the scalability and traffic offloading of the application components (VNFs) [55]. It is necessary to note that the prospected service chain represents the path that should be generated to process the users' requests. The main objective of designing a carrier-grade application-aware (NFV-aware) architecture is to ensure that the system and its services are capable of serving various workloads with insignificant or zero degradation in QoS while maintaining the carrier-grade requirements with minimal SFC delay.

IV. Problem Formulation

In order to take advantage of NFV technology, it is necessary to understand the architecture of its VNFs, their corresponding SFCs, and QoS requirements. This section describes the VNFs architecture and proposes the constraints to satisfy the requirements of QoS and meet the SLA.

A. VNF Architecture

NFV services (applications) are typically developed using a VNF-based architecture where each service consists of one or more VNFs. These VNFs are chained logically to create the service chain as described in the VNFFG. The VNFs' functionalities are combined to provide high-level abstracted services. As described by the VNFFG, the participating VNFs in the service function chain are configured to represent the functional dependencies and form the service computational paths. Fig. 1 illustrates the VNFs' service function chain.

The dependency relation is captured at the service representation level where the delay tolerance and communication bandwidth attributes are defined. The delay tolerance determines the maximum latency at which a VNF instance can maintain communication with its dependent ones without declaring any service or computational path outage or degradation.

B. Requirements of VNFs Scheduler

Each VNF instance of the service is scheduled on a server in the cloud using VMs mappings. Each VM can be hosted on one server and can have at least one VNF instance running on it. Sudden demand spark or failure events can occur in the cloud, such as natural disasters, run-time failures, and global broadcasting events. In order to deal with these events, users' requests/traffic are balanced between various computational paths, or soft failovers to the redundant computational paths groups are triggered. Therefore, increasing the number of the computational paths is translated in better quality of service. The number of computational paths can be increased by adding VNFs on various tiers of the NFV service. However, adding more VNF components can overwhelm the OPEX and CAPEX of the users' investment. Besides, increasing the number of the VNF components while overlooking their optimal placements can result in underutilized VNFs. To address these challenges, this paper proposes a novel NFV-aware scheduling technique to achieve the carrier-grade QoS of an NFV service. The scheduler finds the optimal physical server to host the VNF component while minimizing the delay between the VNFs' components of the service function chain. This technique allows the maximum number of the VNFs to communicate without violating the functional and non-functional constraints. In other words, this technique generates the maximum number of computational paths to serve the users' requests while satisfying the quality requirements.

To consider a successful generation of computational paths, VNFs should be hosted on servers that can satisfy their computing requirements (CPU, memory, storage and networking resources) in the service chain without violating the delay tolerance among their dependent ones. In order to achieve the optimal count of the computational paths, this paper proposes a mixed integer linear programming model to schedule the VNFs while minimizing the traffic delays between the VNFs constituting the service chain. The MILP model provides an NFV-aware placement solution that generates mappings between the cloud physical servers and the VMs on which the tenants' VNFs are hosted while satisfying the following constraints:

(a) *Capacity constraints*: These constraints generate a servers' list that satisfies the resource demands of each VNF to meet the Service Level Agreement (SLA). In the proposed scheduler, the computational resources consist of CPU and memory.

(b) *Network-Delay constraints*: These constraints prune the above list to generate another servers' sub-list that satisfies the latency requirements to avoid any service degradation between the communicating VNFs.

(c) *Availability constraints*: These constraints prune the candidate servers generated by the capacity and delay requirements according to the following constraints:

   i) *Co-location constraint*: It requires that the dependent VNFs should be placed on the same server of their sponsor if the delay tolerance of these dependent VNFs is ephemeral.

   ii) *Anti-location constraint*: It requires that the dependent VNFs should be placed on different servers if their delay tolerances can compensate the communication cost.

   iii) *Redundancy constraint*: With this constraint, VNFs of the same type cannot reside on the same server. In this case, these VNFs should be placed as far as the delay tolerance allows.

(d) *Dependency constraints*: These constraints define the structure of the computational path between the defined VNFs.

V. Mathematical Formulation

In the MILP model, the set of VNFs participating in the SFC is denoted as V. V^A denotes a subset of V where its VNFs should satisfy the anti-location constraint. V^C denotes a subset of V where its VNFs should satisfy the co-location constraint. For each VNF, a subset of V is defined as dependent VNFs and denoted as V^D. v and *v'* represent a single VNF instance that belongs to a given VNF set. V_v^D is defined as the set of dependent VNFs of VNF v. The available set of servers in a given DC is denoted as S while the total number of servers in this set is denoted as N^S. s and *s'* represent a single server that belongs to a given server set. R denotes the set of computational resources types (CPU and memory). r represents a resource type in the computational resources set (CPU or memory). The computational resources r of a specific VNF v are denoted as V_{vr}^{Res}. The available resources r of a server s are denoted by S_{sr}^{Res}. The communication delay tolerance between the VNF components v and *v'* is defined as $T_{vv'}$. The communication delay between servers s and *s'* is denoted by $D_{ss'}$. The delay between two dependent VNFs v and *v'* is defined as $D_{vv'}$.

P_{vs} is the binary decision variable that defines the placement state of a VNF v on server s as follows:

$$P_{vs} = \begin{cases} 1 & if\ VNF\ instance\ v\ is\ placed\ on\ server\ s \\ 0 & otherwise \end{cases} \tag{1}$$

A. Model Formulation

The objective function and the constraints of the proposed MILP model are formulated as follows:
Objective function:

$$Minimize \quad \sum_{v'}^{V^D} D_{vv'}\ \forall\ v \in V \tag{2}$$

Subject to:
Availability/Dependency Constraints:

$$0 \leq P_{vs} \leq 1 \quad \forall\ v\ \epsilon V,\ \forall s \epsilon S \tag{3}$$

$$\sum_{s=0}^{N^S} P_{vs} = 1\ \forall v \epsilon V \tag{4}$$

$$P_{vs} + P_{v's} \leq 1 \quad \forall v, v' \epsilon V^A,\ \forall s \epsilon S \tag{5}$$

$$P_{vs} + P_{v's} \geq 2 \quad \forall v, v' \epsilon V^c,\ \forall s \epsilon S \tag{6}$$

Capacity Constraints:

$$\sum_{v=0}^{V} P_{vs} \times V_{vr}^{Res} \leq S_{sr}^{Res} \quad \forall r \epsilon R, \quad \forall s \epsilon S \tag{7}$$

Network Delay Constraints:

$$D_{ss'} \times (P_{vs} + P_{v's'} - 1) - D_{vv'} \leq 0 \quad \forall v \in V,\ v' \in V_v^D,\ \forall s, s' \in S \tag{8}$$

$$D_{vv'} \leq T_{vv'} \quad \forall v \in V,\ v' \in V_v^D \tag{9}$$

As shown above, the NFV-aware placement constraints are grouped into availability, dependency, capacity, and network connection constraints. Constraint (3) defines the decision variable of the VNFs placement as a binary variable. Constraint (4) ensures that the defined VNF instance can only reside on one server at most. The anti-location constraint is defined in (5) and the co-location constraint is defined in (6). The capacity constraint (7) ensures that the candidate servers should have enough resources to host the assigned VNFs. Constraint (8) is defined as a network connection constraint. The latter ensures that a counted connection is established after the successful placement of the connected VNFs. Constraint (9) reflects the delay tolerance between the VNF types and maps the delay of the hosting servers to their VNFs instances.

B. Model Complexity

In order to determine the complexity of the proposed MILP model, we use reduction method. In this section, we reduce the problem to a bipartite matching one in order to build our model accordingly [56]. Any scheduling problems can be interpreted as a triplet $a \mid b \mid c$, where a represents the problem environment, b represents the problem constraints, and c represents the objective function of the problem [57]. These triplet fields vary depending on the scheduling problem nature. Since the proposed placement approach addresses the allocation problem of a VNF components set (V) on the available servers (S) with an objective function to minimize the communication delay between the dependent components, it can be formulated as a special case of the transportation problem. The formulation for the problem can be represented as $S_s \mid V_v \mid \sum D(x)$ where the S_s is the problem environment consisting of s different parallel servers, V_v defines the VNF job v that can proceed on a single server s, and $D(x)$ represents the objective function to be optimized. In this special case, the problem is known as a constrained bipartite matching problem. $G = (V, S, a)$ represents the bipartite graph that consists of VNF components nodes as set V, server nodes as set S, and arc a connecting the two sets. The arc $a = \{v, s\}$ assigns the VNF component v of set V to server s of set S, and it represents the decision variable P_{vs} defined in the previous section. Said that and using the Hopcroft-Karp algorithm, the bipartite maximal matchings are determined in polynomial time to the number of edges and vertices [58]. Thus, this type of bipartite matching problem that is formulated using linear programming models is categorized as NP-hard problem, and by reduction, the proposed MILP model is NP-hard. Therefore, the proposed MILP model would be solvable for small-scale DC networks [59]. With this in mind, this paper proposes a heuristic approach, BACON, to address the large-scale DC networks.

VI. BACON: NFV-Aware Placement Algorithm

Due to the computational complexity of the proposed MILP model (NP-hard) and given the available computing processing power, the optimization model imposes a limitation on scaling to large-scale data center network. Therefore, this section proposes a novel heuristic solution, Betweenness centrality Algorithm for Component Orchestration of NFV platform (BACON). BACON is based on the betweenness centrality of a node in a graph that works around the complexity and the time-consuming execution of the MILP model. Given a set of servers S and a set of VNFs participating in a SFC, BACON finds a feasible near optimal VNF placement solution compared to the MILP optimal solution. The generated solution satisfies the previous constraints while relaxing the objective function. BACON executes different subroutines to find the placement solutions.

Prior to the placement subroutine, BACON analyzes the types of the participating VNF in a given SFC. The VNF types are then divided into sub-groups according to their inherited dependency from the VNF Forwarding Graph (VNFFG). Each sub-group consists of three VNF types and is assigned a criticality attribute based on the communication delay tolerance of the participating VNF types. If BACON finds an undercount group, it shares VNF types from another sub-group. It is necessary to note that a group is considered as an undercount one when it contains less than three VNF types. After the grouping step, BACON builds a graph to represent the model system. The graph is built while considering that all the available servers in the data center are connected through a logical communication link in a mesh topology. BACON constructs the weighted graph $G(V, E, w)$ where the vertices V represents the set of available servers in a given data center, the edges $E(v, v')$ represents the logical communication link between the servers, and the weights $w(v, v')$ represents the data communication delay between the servers. Since the SFC is divided into sub-groups of three components, the count of the vertices is triple the number of the servers. Thus, BACON covers all the placement possibilities of a sub-group. Once the graph is built, BACON calculates the Betweenness Centrality (BC) of the vertices (servers).

A. Calculation of Betweenness Centrality

The calculation of the betweenness centrality is based on the number of the shortest paths from the source node (s) to sink node (t) that passes through a specific node. *Betweenness centrality:*

$$B(v) = \sum \frac{\alpha_{st}(v)}{\alpha_{st}} \quad \forall v \neq s,\ v \neq t$$
$$where \begin{cases} \alpha_{st}(v) = \text{Number of shortest paths from s to t passing through } v \\ \alpha_{st} = \text{Total number of shortest paths from s to t} \end{cases} \tag{10}$$

Calculating the BC identifies the servers that can be anchors for the median nodes in the defined sub-groups. Median nodes are the VNF instances of the mediator VNF type in a given sub-group. For example, in Fig. 1, the mediator VNF type in the given sub-group is VNF type 2. The placement of the median nodes of the sub-group is based on the critically attribute. BACON starts by placing the most critical VNF components of the sub-groups' median VNF types on the servers with the highest BC while satisfying the functional constraints. This placement criterion guarantees that the highest critical VNF components in a sub-group are placed in the most branched servers with the minimal communication delays. It also guarantees that the critical component has the maximum count of the computational paths between the sub-group members without violating the communication delay tolerance.

Once the median VNF components of the sub-group are placed on the servers, BACON hosts the members of the other sub-group on the servers. The group members that interconnect the sub-groups are placed on the servers with the highest BC. These servers belong to the intersection subset of the candidate servers of the interconnected sub-groups median as follows:

$$S_m = S_{SG'} \cap S_{SG''}$$
$$where \begin{cases} S_m = \text{A set of candidate servers to place the sub-group members} \\ S_{SG'} = \text{A set of candidate servers to place the sub-group } SG' \text{ median members} \\ S_{SG''} = \text{A set of candidate servers to place the sub-group } SG'' \text{ median members} \end{cases} \tag{11}$$

BACON ensures that the group members have the best-fit servers with the most branching communication paths without violating the communication delay tolerance not only between the members of a sub-group but also between the interconnected members of the other sub-groups. Finally, BACON returns the VNF components set where each component is associated with a host. The generated placement is considered the best effort to achieve the minimum delay between the VNF components while maximizing the count of the possible computation paths. BACON is represented in Fig. 2.

Algorithm 1 BACON

```
INPUT: V = (V1, V2, ..., Vv)
       V^D = (V1, V2, ..., Vn)
       S = (S1, S2, ..., Sk)
OUTPUT: S^P
        where S^P ⊂ S
begin:
C = VNFComponentCriticalityRank(V)
SubGroups = DivideIntoSubGroups(V, V^D, 3)
SubGroups.AssociateCriticality(C)
G = BuildGraph(S, S.Delays)
for s_i ∈ S do
  | s_i.Centrality = G.BetweennessCentrality(s_i)
end for
S.DescendingSort(Centrality)
SubGroups.DescendingSort(Criticality)
for g_i ∈ SubGroups do
  | v = g_i.GetMedainVNF
  | VNFType = v.GetType
  | for v_i : VNFType.InstancesCount do
  |   | for s_i ∈ S do
  |   |   | if s_i.AvailableResources >= v_i.Resources then
  |   |   |   | s_i.Host(v_i)
  |   |   |   | S^P.Add(s_i)
  |   |   |   | Break
  |   |   | end if
  |   | end for
  | end for
end for
for g_i ∈ SubGroups do
  | S' = g_i.CandidateServers ∩ g_{i+1}.CandidateServers
  | M = g_i.GetVNFMembers
  | for m_i ∈ M do
  |   | VNFType = m_i.GetType
  |   | for v_i : VNFType.InstancesCount do
  |   |   | for s_i ∈ S' do
  |   |   |   | if s_i.AvailableResources >= v_i.Resources then
  |   |   |   |   | s_i.Host(v_i)
  |   |   |   |   | S^P.Add(s_i)
  |   |   |   |   | Break
  |   |   |   | end if
  |   |   | end for
  |   | end for
  | end for
end for
return : S^P
end
```

Fig. 2. BACON: The proposed heuristic algorithm.

The highest order of magnitude in BACON is the subroutine that calculates the betweenness centrality of the vertices nodes. Examining the subroutine closely, the worst-case scenario can be calculated by finding all the combinations of the sub-groups while holding the median node then calculating the betweenness centrality of the median nodes. The results in order of magnitude are as follows:

$$O(\frac{n!}{(n-2)! \times 2!}) = O(\frac{n^2 - n}{2}) \tag{12}$$

Iterating n times over the median node, the worst case is then:

$$O(\frac{n^2 - n}{2}) \times n = O(\frac{n^3 - n^2}{2}) \tag{13}$$

Given n as the total number of available servers *"S"* in a given data center then

$$O(\frac{S^3 - S^2}{2}) \tag{14}$$

is the highest order of magnitude for BACON.

VII. NFV-Aware Placement Simulation

At the root level, the cloud consists of data centers distributed across various geographical areas. Each data center consists of multiple racks communicating through aggregated switches. Each rack has a set of shelves hosting servers, which can have different resources capacities. Servers residing on the same rack are connected with each other through the same network device, such as the Top Of the Rack (TOR) switch. Finally, the VMs/containers are hosted on the servers. This tree structure determines the network delay constraints and consequently, the delay between the communicating VNFs. This architecture divides the cloud into different latency zones. For the simulation, we have considered 3-tier data center with:

- Access Switches or TOR Switches: Connecting the servers in the same rack.
- Aggregation Switches (AS): Connecting the TOR switches.
- Core Switches: Connecting the AS and acting as gateways to the external networks.

In order to generate the delay data-set of the servers in the simulation, we distribute the servers among the DC's racks and their data flow throughout the 3-tier DC network. Each DC network tier represents a specific delay with each unique server-to-server connection. The delays are generated randomly and follow a normal distribution with a specific predefined 99th percentile latency for each tier [60]–[62].

A. Simulation Results and Evaluation

The proposed MILP model and BACON are compared to two greedy algorithms. The first greedy algorithm is an NFV-agnostic algorithm. The other one is an NFV-aware algorithm, "Greedy-k-NFV algorithm" that is proposed by Qu *et al.* [43]. This comparison shows the impact of NFV-aware placement on the computational paths' delays that affect the validity of these paths. It also evaluates the performance of BACON. During the simulation, we have used the vEPC as the simulation use case [12]. The 3rd Generation Partnership Project (3GPP) group introduces the EPC as all-Internet-Protocol (IP) core network architecture [63]. It is designed to unleash the full potentials of mobile networks to provide broadband services. In the simulation, the four major components of the EPC have been considered; Mobile Management Entity (MME), Home Subscriber Server (HSS), Serving Gateway (SGW), and Packet Data Network Gateway (PGW or PDN-GW). Each component represents a VNF type in the input data-sets of the simulation.

The simulation testbed is implemented and deployed on SharcNet computing platform [64]. Wobbie-142 computing server is used to execute the simulation. Wobbie-142 computing server has 24 core-48 thread Intel Xeon E5-2690 v3 (2x sockets configuration) and 768.0 GB of memory. The simulation is executed in two phases:

TABLE I
Small-Scale DC Network Data Set

Set	Count
Available servers in DC	30
VNF of type MME	2
VNF of type HSS	3
VNF of type SGW	2
VNF of type PGW	3

TABLE II
Large-Scale DC Network Data Set

Set	Count
Available servers in DC	300
VNF of type MME	20
VNF of type HSS	23
VNF of type SGW	25
VNF of type PGW	30

- *Phase 1: Small-scale DC network simulation*
  In this phase, the data-set of a small-scale DC network is the input of the MILP model, BACON, and the greedy algorithms in the testbed. The input data is shown in Table I, and the evaluation results are shown in Fig. 3, Fig. 4, Fig. 5, and Fig. 6.
- *Phase 2: Large-scale DC network simulation*
  In this phase, the data-set of a large-scale DC network is the input of BACON and the greedy algorithms in the testbed. The input data is shown in Table II, and the evaluation results are shown in Fig. 7.

B. Components Intra-Communication Delay Comparative Analysis

This section provides a comparative analysis between the proposed NFV-aware MILP model, BACON, and the other greedy placement algorithms for small- and large-scale DC networks.

1) Small-Scale Network Simulation: Fig. 3 shows the connection delays between the VNF instances of types MME and HSS. Fig. 4 shows the connection delays between the VNF instances of types MME and SGW. Fig. 5 shows the connection delays between the VNF instances of types SGW and PGW. As shown in the figures, the MILP model generates the connections with the optimal minimum delay of the intra-connectivity between the entities. BACON achieves a near optimal minimum delay where it deviates slightly from the MILP results. However, BACON has the lowest delays when compared to the other two greedy algorithms especially the "greedy-k-NFV" algorithm [43], which minimizes the communication delay of the SFC entities. The proposed MILP model and BACON do not only minimize the communication delay of the intra-links, but they also provide the best count of the links that satisfy the delay tolerance constraints between the VNFs instances. However, the other greedy algorithms generate placement decisions that violate the delay tolerance constraints between the VNFs instances. Any violation of the

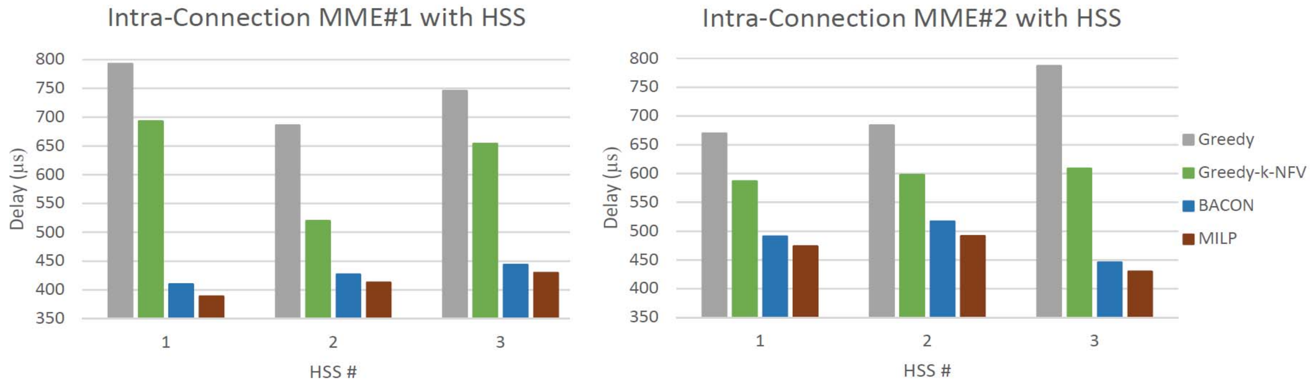

Fig. 3. Intra-connection delay between VNF instances of types MME and HSS.

Fig. 4. Intra-connection delay between VNF instances of types MME and SGW.

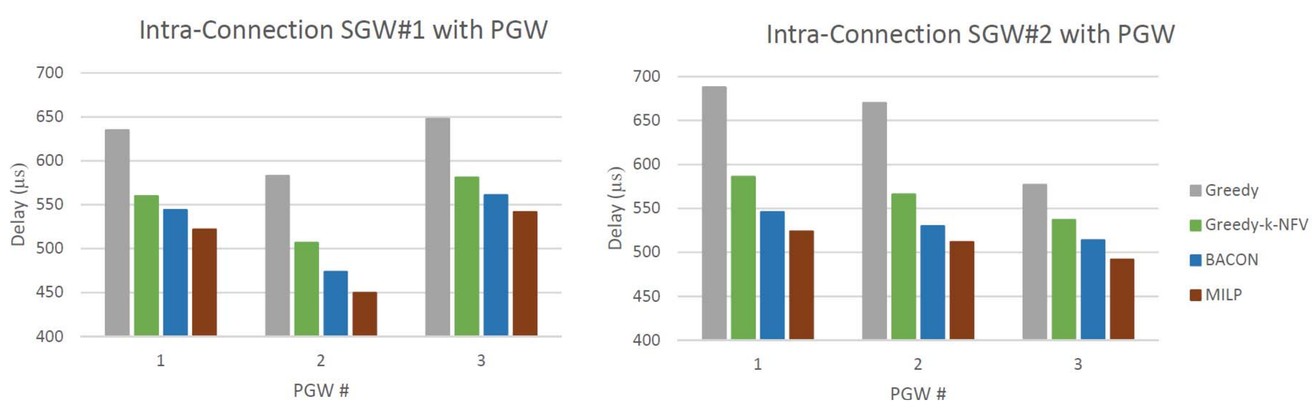

Fig. 5. Intra-connection delay between VNF instances of types SGW and PGW.

delay tolerance constraints terminates the connection between the VNF instances, and the link is considered as an invalid one for a computational path. The computational paths delays are shown in Fig. 6.

The benefits of increasing the number of computational paths can be quantified by assessing how many members are participating in a functional group of a VNF instance. All group members should share the same VNF type and reside in the same orbital area. The orbital area is defined by the area where the functional group members can maneuver without violating any of the previous constraints. Fig. 8 shows the VNF orbital area. The boundaries of an orbital area are defined by the delay tolerance constraints of the dependent VNF instances. The higher the number of participating members in the functional group, the better its performance and reliability. The SFC performance and availability can be enhanced by the functional group members. From a performance perspective, user data traffic can be offloaded between the functional group members. The traffic offloading process is mainly managed by the health check entities in a system. The health check entities constantly monitor and collect various metrics from the active VNFs instances and balance the traffic to achieve the desired performance. From an availability perspective of the SFC, the functional group members are considered as redundant components that can mitigate the failure of the VNF instances due to a sudden

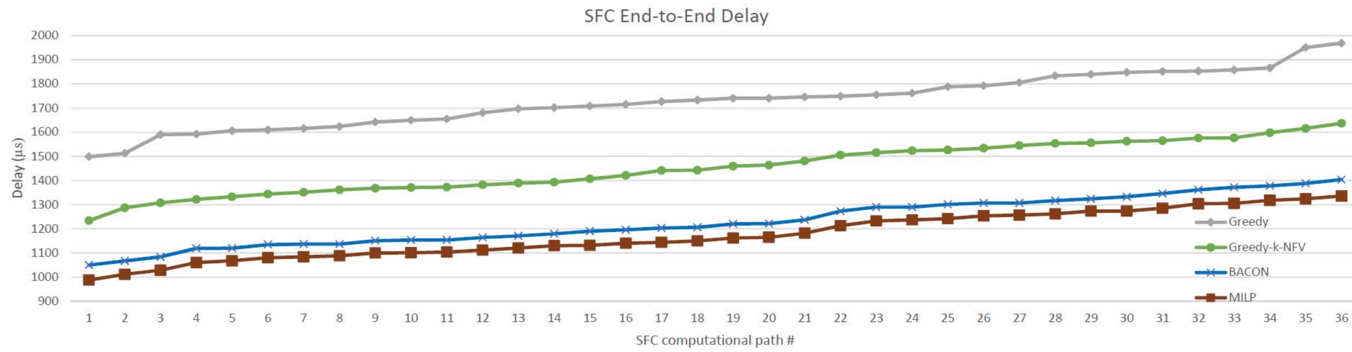

Fig. 6. The end-to-end delays of SFCs in small-scale DC network.

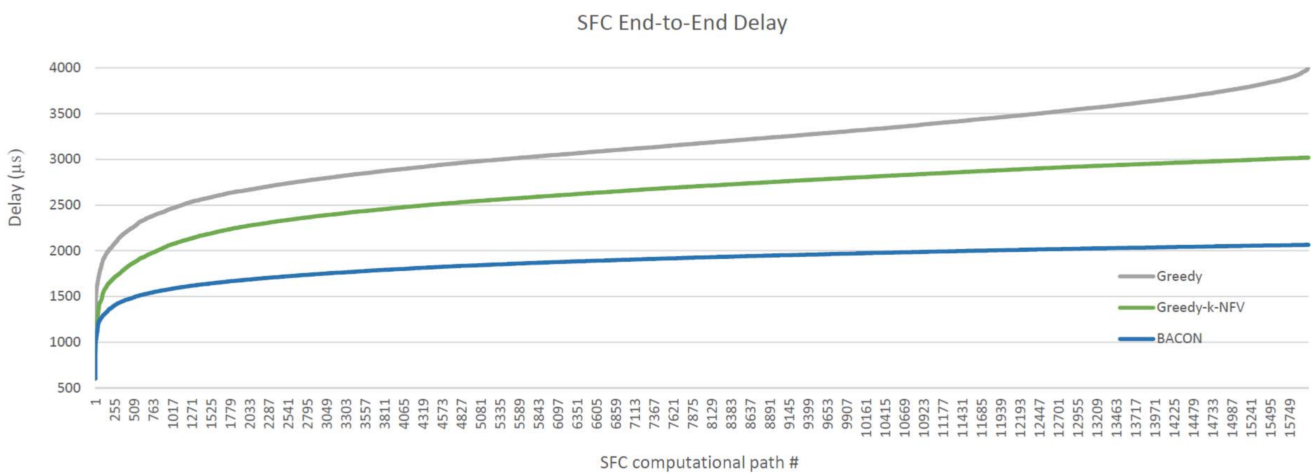

Fig. 7. The end-to-end delays of SFCs in large-scale DC network.

interruption that affects the QoS of the SFC. The proposed MILP model and BACON generate the best count of functional group members. The results of Table III represent the VNF instances count in each VNF-type functional group for the small-scale DC simulation. The results show that the proposed heuristic "BACON" and the MILP model have achieved the best count of VNF members in each VNF-type functional group. BACON and the MILP model have achieved count of two, three, two, and three group members for the following VNF-types; MME-VNF-type, HSS-VNF-type, SGW-VNF-type, and PGW-VNF-type, respectively. In contrary, Greedy-k-NFV algorithm has achieved one, two, one, and one and the Greedy algorithm has achieved one, one, one, and one for these VNF-types; MME-VNF-type, HSS-VNF-type, SGW-VNF-type, and PGW-VNF-type, respectively. Achieving higher member counts (higher VNF count of different types) in a specific functional group enhances the QoE for the service users. QoE is determined by the perception and evaluation of a service from user viewpoint. With the increase in the member counts in a functional group, the number of possible computational paths increments accordingly. These paths can be optimally used by services to facilitate the migration of data traffic between different computational paths in case of any degradation in performance or to migrate any errors while providing a seamless service to the user and maintaining the desired level of QoE.

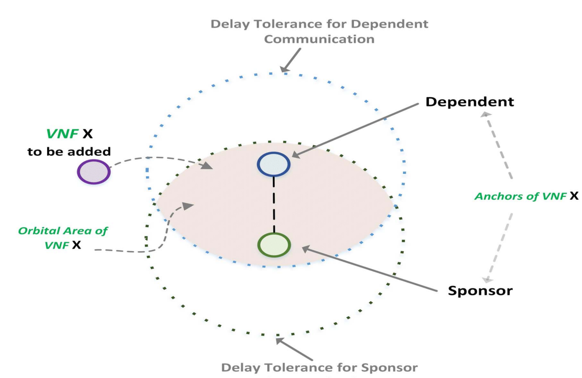

Fig. 8. The placement zones of VNFs depending on their sponsors and dependent VNFs (of different types).

2) Large-Scale Network Simulation: The MILP model has a high order of magnitude that hinders the results generation within a reasonable time given the available computing processing power. Therefore, it is not evaluated on the large-scale network simulation. BACON, the greedy NFV-agnostic, and the greedy-k-NFV algorithms are evaluated on the large-scale network. The simulation results are shown in Fig. 7, and the functional group counts are represented in Table IV. Similar to the small-scale network simulation,

TABLE III

THE MEMBERS' COUNT OF FUNCTIONAL GROUP FOR DIFFERENT VNF TYPES IN SMALL-SCALE DC NETWORK

DC	Greedy	Greedy-k-NFV	BACON	MILP
VNF of type MME functional group	1	1	2	2
VNF of type HSS functional group	1	2	3	3
VNF of type SGW functional group	1	1	2	2
VNF of type PGW functional group	1	1	3	3

TABLE IV

THE MEMBERS' COUNT OF FUNCTIONAL GROUP FOR DIFFERENT VNF TYPES IN LARGE-SCALE DC NETWORK

DC	Greedy	Greedy-k-NFV	BACON
VNF of type MME functional group	3	12	18
VNF of type HSS functional group	7	13	22
VNF of type SGW functional group	4	16	24
VNF of type PGW functional group	9	21	30

BACON achieves the lowest delays of the SFC computational paths and the highest count of the functional group members when compared to the other two greedy algorithms. The results in Table IV show that the proposed heuristic "BACON" has achieved the best count of VNF members in each VNF-type functional group. BACON has achieved members' count of 18, 22, 24, and 30 group members for the following VNF-types MME-VNF-type, HSS-VNF-type, SGW-VNF-type, and PGW-VNF-type, respectively. However, the Greedy-k-NFV algorithm has achieved 12, 13, 16, and 21 and the Greedy algorithm has achieved 3, 7, 4, and 9 for the following VNF-types MME-VNF-type, HSS-VNF-type, SGW-VNF-type, and PGW-VNF- type, respectively.

BACON outperforms the other two greedy algorithms especially the greedy-k-NFV. The greedy-k-NFV is proposed to overcome the scalability of an optimization model while maximizing the reliability and minimizing the SFC end-to-end delays [43]. When compared to the greedy-k-NFV algorithm, BACON has a lower order of magnitude, which allows better scalability of the algorithm. The greedy-k-NFV has the following order of magnitude:

$$O(kN(M + N\log N))$$

$$where \begin{cases} k = \text{Initial set of paths} \\ M = \text{Number of edges} \\ N = \text{Number of nodes in network} \end{cases} \quad (15)$$

The simulation environment consists of DCs with multiple commodity servers to host the NFV applications. Given this simulation setup, the greedy-k-NFV algorithm variable can then be represented as follows:

- $k = S$, the number of servers in a given DC since the VNF instances can be hosted on any server in the DC.
- $M = S^2$, since all servers are connected to each other with a logical mesh network.
- $N = S$, since the node in a network represents a server in the DC.

To this end, the order of magnitude of the greedy-k-NFV algorithm can be represented as:

$$O(S^4 + S^3 \log S)) \quad (16)$$

where S = Number of servers in a given data center

This shows that BACON has a lower order of magnitude:

$$O(\frac{S^3 - S^2}{2}) \quad (17)$$

Thus, BACON outperforms the greedy-k-NFV algorithm as shown earlier.

C. SFC End-to-End Delay Comparative Analysis

The proposed MILP model and BACON do not only increase the count of the functional group members, but their placements' results show that the computational paths' delays are minimized when compared to the other two greedy algorithms. The computational paths' delays are shown in Fig. 6 and Fig. 7 for small- and large-scale networks respectively. Minimizing the computational paths' delays is a necessity for the SFC orchestration and management entities because time difference between the delay tolerance and computing paths' delays allow the orchestration and the management entities to apply various policies on the systems. These policies vary according to the intent of the network service providers. For example, network service providers can introduce policies to achieve green or security analysis networks.

VIII. CONCLUSION

NFV has been introduced by the leading NSPs as a technology to revolutionize the information and communications technology industry. It has transformed the network functions from proprietary hardware to software-based applications where virtualization can be exploited. The academic and industrial researchers are investigating the possibilities of integrating NFV with the virtualization platforms. This step paves the way to unleash the full potentials of the NFV technology. Therefore, various NFV challenges should be resolved to achieve a wider adoption of this technology.

In this paper, we presented a novel approach to address the placement problem of VNFs and their associated SFCs. An MILP model and a heuristic algorithm, BACON, were proposed to minimize the communication delay between the VNF instances and enhance the end-to-end QoS of the SFC. The proposed MILP model and BACON are implemented to capture the carrier-grade requirements of an NFV application. They are also evaluated on small- and large-scale DC networks data-set. In both cases, the proposed MILP model and BACON outperform the greedy NFV-agnostic and NFV-aware algorithms.

REFERENCES

[1] Ericsson, "Ericsson mobility report on the pulse of the network society," Ericsson, Stockholm, Sweden, White Paper EAB-16:006659 Uen, Nov. 2016.

[2] Cisco, "Cisco visual networking index: Forecast and methodology, 2016–2021," Cisco, San Jose, CA, USA, White Paper C11-481360-01, Jun. 2017.

[3] IDC, "Is your network ready for digital transformation?" White Paper, IDC#US41912917, Indianapolis, IN, USA, Jan. 2017.

[4] M. A. Sharkh, M. Jammal, A. Shami, and A. Ouda, "Resource allocation in a network-based cloud computing environment: Design challenges," *IEEE Commun. Mag.*, vol. 51, no. 11, pp. 46–52, Nov. 2013.

[5] ETSI, "Network function virtualization: An introduction, benefits, enablers, challenges, & Call for action," ETSI, Sophia Antipolis, France, White Paper 1, Oct. 2012.

[6] M. Jammal, T. Singh, A. Shami, R. Asal, and Y. Li, "Software defined networking: State of the art and research challenges," *Comput. Netw. J.*, vol. 72, pp. 74–98, Oct. 2014.

[7] A. Ksentini, M. Bagaa, and T. Taleb, "On using SDN in 5G: The controller placement problem," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2016, pp. 1–6.

[8] A. Ksentini, M. Bagaa, T. Taleb, and I. Balasingham, "On using bargaining game for optimal placement of SDN controllers," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2016, pp. 1–6.

[9] ETSI, "Network functions virtualisation (NFV); management and orchestration," ETSI GS NFV-MAN, White Paper 001 V1.1.1 (2014-12), Dec. 2014.

[10] ETSI, "Network functions virtualisation (NFV); ecosystem; report on SDN usage in NFV architectural framework," ETSI GS NFV-EVE, White Paper 005 V1.1.1 (2015-12), Dec. 2015.

[11] M. Bagaa, T. Taleb, and A. Ksentini, "Service-aware network function placement for efficient traffic handling in carrier cloud," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Apr. 2014, pp. 2402–2407.

[12] H. Hawilo, A. Shami, M. Mirahmadi, and R. Asal, "NFV: State of the art, challenges, and implementation in next generation mobile networks (vEPC)," *IEEE Netw.*, vol. 28, no. 6, pp. 18–26, Nov. 2014.

[13] R. Mijumbi, J. Serrat, J.-L. Gorricho, S. Latré, M. Charalambides, and D. Lopez, "Management and orchestration challenges in network functions virtualization," *IEEE Commun. Mag.*, vol. 54, no. 1, pp. 98–105, Jan. 2016.

[14] H. Hawilo, L. Liao, A. Shami, and V. C. M. Leung, "NFV/SDN-based vEPC solution in hybrid clouds," in *Proc. IEEE Middle East North Africa Commun. Conf. (MENACOMM)*, Apr. 2018, pp. 1–6.

[15] A. M. Medhat, T. Taleb, A. Elmangoush, G. A. Carella, S. Covaci, and T. Magedanz, "Service function chaining in next generation networks: State of the art and research challenges," *IEEE Commun. Mag.*, vol. 55, no. 2, pp. 216–223, Feb. 2017.

[16] S. Mehraghdam, M. Keller, and H. Karl, "Specifying and placing chains of virtual network functions," in *Proc. IEEE 3rd Int. Conf. Cloud Netw. (CloudNet)*, Oct. 2014, pp. 7–13.

[17] R. Cohen, L. Lewin-Eytan, J. S. Naor, and D. Raz, "Near optimal placement of virtual network functions," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, May 2015, pp. 1346–1354.

[18] R. Mijumbi *et al.*, "Design and evaluation of algorithms for mapping and scheduling of virtual network functions," in *Proc. IEEE Conf. Netw. Softwarization*, Apr. 2015, pp. 1–9.

[19] T. Taleb, A. Ksentini, M. Chen, and R. Jantti, "Coping with emerging mobile social media applications through dynamic service function chaining," *IEEE Trans. Wireless Commun.*, vol. 15, no. 4, pp. 2859–2871, Apr. 2016.

[20] V. Eramo, E. Miucci, M. Ammar, and F. G. Lavacca, "An approach for service function chain routing and virtual function network instance migration in network function virtualization architectures," *IEEE/ACM Trans. Netw.*, vol. 25, no. 4, pp. 2008–2025, Aug. 2017.

[21] S. Ayoubi, S. Sebbah, and C. Assi, "A cut-and-solve based approach for the VNF assignment problem," *IEEE Trans. Cloud Comput.*, to be published.

[22] S. Dutta, T. Taleb, and A. Ksentini, "QoE-aware elasticity support in cloud-native 5G systems," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2016, pp. 1–6.

[23] J. Batalle, J. F. Riera, E. Escalona, and J. A. Garcia-Espin, "On the implementation of NFV over an OpenFlow infrastructure: Routing function virtualization," in *Proc. IEEE SDN Future Netw. Services (SDN4FNS)*, Nov. 2013, pp. 1–6.

[24] I. Cerrato, M. Annarumma, and F. Risso, "Supporting fine-grained network functions through Intel DPDK," in *Proc. 3rd Eur. Workshop Softw. Defined Netw. (EWSDN)*, Sep. 2014, pp. 1–6.

[25] J. Martins *et al.*, "ClickOS and the art of network function virtualization," in *Proc. 11th USENIX Conf. Netw. Syst. Design Implement.*, Apr. 2014, pp. 459–473.

[26] T. Taleb, A. Ksentini, and R. Jantti, "'Anything as a service' for 5G mobile systems," *IEEE Netw. Mag.*, vol. 30, no. 6, pp. 84–91, Dec. 2016.

[27] M. Jammal, A. Kanso, and A. Shami, "High availability-aware optimization digest for applications deployment in cloud," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2015, pp. 6822–6828.

[28] M. Jammal, A. Kanso, P. Heidari, and A. Shami, "Availability analysis of cloud deployed applications," in *Proc. IEEE Int. Conf. Cloud Eng. (IC2E)*, Apr. 2016, pp. 234–235.

[29] M. Jammal, H. Hawilo, A. Kanso, and A. Shami, "Mitigating the risk of cloud services downtime using live migration and high availability-aware placement," in *Proc. IEEE Int. Conf. Cloud Comput. Technol. Sci. (CloudCom)*, Dec. 2016, pp. 578–583.

[30] M. Jammal, A. Kanso, P. Heidari, and A. Shami, "Evaluating high availability-aware deployments using stochastic Petri net model and cloud scoring selection tool," *IEEE Trans. Services Comput.*, to be published.

[31] T. Taleb *et al.*, "EASE: EPC as a service to ease mobile core network deployment over cloud," *IEEE Netw. Mag.*, vol. 29, no. 2, pp. 78–88, Mar. 2015.

[32] H. Baba, M. Matsumoto, and K. Noritake, "Lightweight virtualized evolved packet core architecture for future mobile communication," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Mar. 2015, pp. 1811–1816.

[33] L. E. Li *et al.*, "PACE: Policy-aware application cloud embedding," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Apr. 2013, pp. 638–646.

[34] P. Shoumik *et al.*, "E2: A framework for NFV applications," in *Proc. 25th Symp. Oper. Syst. Princ.*, Oct. 2015, pp. 121–136.

[35] F. Bari, S. R. Chowdhury, R. Ahmed, R. Boutaba, and O. C. M. B. Duarte, "Orchestrating virtualized network functions," *IEEE Trans. Netw. Service Manage.*, vol. 13, no. 4, pp. 725–739, Dec. 2016.

[36] A. Mohammadkhan, S. Ghapani, G. Liu, W. Zhang, K. K. Ramakrishnan, and T. Wood, "Virtual function placement and traffic steering in flexible and dynamic software defined networks," in *Proc. 21st IEEE Int. Workshop (LANMAN)*, Apr. 2015, pp. 1–6.

[37] S. Sahhaf *et al.*, "Network service chaining with optimized network function embedding supporting service decompositions," *Comput. Netw. J.*, vol. 93, pp. 492–505, Dec. 2015.

[38] T. M. Nguyen, S. Fdida, and T. M. Pham, "A comprehensive resource management and placement for network function virtualization," in *Proc. IEEE Conf. Netw. Softwarization (NetSoft)*, Jul. 2017, pp. 1–9.

[39] A. Gadre, A. Anbiah, and K. M. Sivalingam, "A customizable agile approach to network function placement," in *Proc. Eur. Conf. Netw. Commun. (EuCNC)*, Jun. 2017, pp. 1–6.

[40] S. Ahvar, H. P. Phyu, S. M. Buddhacharya, E. Ahvar, N. Crespi, and R. Glitho, "CCVP: Cost-efficient centrality-based VNF placement and chaining algorithm for network service provisioning," in *Proc. IEEE Conf. Netw. Softwarization (NetSoft)*, Jul. 2017, pp. 1–9.

[41] L. Gupta, M. Samaka, R. Jain, A. Erbad, D. Bhamare, and C. Metz, "COLAP: A predictive framework for service function chain placement in a multi-cloud environment," in *Proc. IEEE 7th Annu. Comput. Commun. Workshop Conf. (CCWC)*, Jan. 2017, pp. 1–9.

[42] Q. Zhang, Y. Xiao, F. Liu, J. C. S. Lui, J. Guo, and T. Wang, "Joint optimization of chain placement and request scheduling for network function virtualization," in *Proc. IEEE 37th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jun. 2017, pp. 731–741.

[43] 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 Trans. Netw. Service Manag.*, vol. 14, no. 3, pp. 554–568, Sep. 2017.

[44] H. Hantouti, N. Benamar, T. Taleb, and A. Laghrissi, "Traffic steering for service function chaining," *IEEE Commun. Surveys Tuts.*, to be published.

[45] M. Bagaa, T. Taleb, A. Laghrissi, A. Ksentini, and H. Flinck, "Coalitional game for the creation of efficient virtual core network slices in 5G mobile systems," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 3, pp. 469–484, Mar. 2018.

[46] I. Benkacem, T. Taleb, M. Bagaa, and H. Flinck, "Optimal VNFs placement in CDN slicing over multi-cloud environment," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 3, pp. 616–627, Mar. 2018.

[47] A. Laghrissi, T. Taleb, and M. Bagaa, "Conformal mapping for optimal network slice planning based on canonical domains," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 3, pp. 519–528, Mar. 2018.

[48] T. Taleb, M. Bagaa, and A. Ksentini, "User mobility-aware virtual network function placement for virtual 5G network infrastructure," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2015, pp. 3879–3884.

[49] A. Laghrissi, T. Taleb, M. Bagaa, and H. Flinck, "Towards edge slicing: VNF placement algorithms for a dynamic & amp; realistic edge cloud environment," in *Proc. IEEE Global Commun. Conf.*, Dec. 2017, pp. 1–6.

[50] H. Hawilo, A. Kanso, and A. Shami, "Towards an elasticity framework for legacy highly available applications in the cloud," in *Proc. IEEE World Congr. Services*, Jun./Jul. 2015, pp. 253–260.

[51] M. Jammal, A. Kanso, P. Heidari, and A. Shami, "A formal model for the availability analysis of cloud deployed multi-tiered applications," in *Proc. IEEE Int. Conf. Cloud Eng. Workshop (IC2EW)*, Apr. 2016, pp. 82–87.

[52] M. Jammal, H. Hawilo, A. Kanso, and A. Shami, "ACE: Availability-aware CloudSim extension," *IEEE Trans. Netw. Service Manage.*, vol. 15, no. 4, pp. 1586–1599, Dec. 2018.

[53] M. Jammal, H. Hawilo, A. Kanso, and A. Shami, "Generic input template for cloud simulators: A case study of CloudSim," *Softw., Pract. Exper.*, pp. 1–28, 2018.

[54] Netflix. (2016). *Completing the Netflix Cloud Migration*. Accessed: Sep. 22, 2017. [Online]. Available: https://media.netflix.com/en/company-blog/completing-the-netflix-cloud-migration

[55] H. Hawilo, M. Jammal, and A. Shami, "Orchestrating network function virtualization platform: Migration or re-instantiation?," in *Proc. IEEE 6th Int. Conf. Cloud Netw. (CloudNet)*, Sep. 2017, pp. 1–6.

[56] N. Mustafa. (2013). *Reductions to Prove NP Hardness*. [Online]. Available: https://perso.esiee.fr/~mustafan/TechnicalWritings/Complexityslides/lec6.pdf

[57] M. L. Pinedo, "Deterministic models: Preliminaries," in *Scheduling—Theory, Algorithms, and Systems*. Boston, MA, USA: Springer, 2011.

[58] Cornell University. (2017). *Matchings*. [Online]. Available: http://www.cs.cornell.edu/courses/cs6820/2016fa/handouts/matchings.pdf

[59] R. Čorić, M. Dumić, and D. Jakobović, "Complexity comparison of integer programming and genetic algorithms for resource constrained scheduling problems," in *Proc. Int. Conv. Inf. Commun. Technol., Electron. Microelectron.*, 2017, pp. 1182–1188.

[60] C. Guo *et al.*, "Pingmesh: A large-scale system for data center network latency measurement and analysis," in *Proc. ACM Conf. Special Interest Group Data Commun. (SIGCOMM)*, Aug. 2015, pp. 139–152.

[61] Plexxi, "Latency in Ethernet switches," Plexxi, Nashua, NH, USA, White Paper, 2016. [Online]. Available: http://www.plexxi.com/wp-content/uploads/2016/01/Latency-in-Ethernet-Switches.pdf

[62] VMware, "Network I/O Latency on VMware vSphere 5," Technical White Paper, Palo Alto, CA, USA, 2012. [Online]. Available: https://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/techpaper/network-io-latency-perf-vsphere5-white-paper.pdf

[63] *Digital Cellular Telecommunications System (Phase 2+); Universal Mobile Telecommunications System (UMTS);LTE; Network Architecture*, document 3GPP TS 23.002 version 11.6.0, ETSI, 2013.

[64] SHARCNET. (Apr. 2017). *Cluster Wobbie.Sharcnet.Ca*. Accessed: Aug. 5, 2017. [Online]. Available: https://www.sharcnet.ca/my/systems/show/29

Hassan Hawilo received the B.E. degree in communication and electronics engineering from Beirut Arab University, Lebanon, in 2012, and the M.E.Sc. degree in computer and software engineering from Western University, Canada, in 2015, where he is currently pursuing the Ph.D. degree in computer networks and cloud computing virtualization technologies. His research interests include cloud computing, virtualization technologies, software defined network, network function virtualization, distributed systems, and highly available software. He is the Vice-Chair of the IEEE Computer Society, London, ON, Canada.

Manar Jammal received the B.Sc. degree in electrical and computer engineering from Lebanese University, Beirut, Lebanon, in 2011, and the M.E.Sc. degree in electrical and electronics engineering from the Ecole Doctorale des Sciences et de Technologie, Beirut, and from the University of Technology of Compiegne, France, in 2012, and the Ph.D. degree in high availability of cloud applications from Western University, London, Canada, in 2017. She is currently a Post-Doctoral Associate with Western University. Her research interests include cloud computing, virtualization, high availability, cloud simulators, machine learning, software defined network, and virtual machine migrations. She is the Chair of the IEEE Canada Women in Engineering.

Abdallah Shami (SM'09) received the B.E. degree in electrical and computer engineering from Lebanese University, Beirut, Lebanon, in 1997, and the Ph.D. degree in electrical engineering from the Graduate School and University Center, City University of New York, in 2002. In 2002, he joined the Department of Electrical Engineering, Lakehead University, Thunder Bay, Canada, as an Assistant Professor. Since 2004, he has been with Western University, Canada, where he is currently a Professor with the Department of Electrical and Computer Engineering. His current research interests are network optimization, cloud computing, and wireless networks. He is a Senior Member of IEEE. He was the Elected Chair of the IEEE London Section, and a Chair of the IEEE Communications Society Technical Committee on Communications Software. He is an Editor of the IEEE COMMUNICATIONS TUTORIALS AND SURVEY. He has served on the Editorial Board of the IEEE COMMUNICATIONS LETTERS from 2008 to 2013. He is an IEEE Distinguished Lecturer.