Orchestrating Network Function Virtualization Platform: Migration or Re-Instantiation?

Hassan Hawilo, Manar Jammal, and Abdallah Shami
Department of Electrical and Computer Engineering, Western University, Canada
hhawilo, mjammal, abdallah.shami@uwo.ca

***Abstract*— Network function virtualization (NFV) provokes the evolution of network functions to overcome various challenges facing the network service providers (NSPs). To exploit the advantages of the virtualization technology, NFV platforms should use the cloud environment to provide their services. Typically, an NFV service is represented by a service function chain (SFC) that consists of multiple virtualized network functions (VNFs). Hosting and orchestrating these VNFs in a cloud environment are challenging tasks. In this paper, we discuss the VNF orchestration problem from the perspective of VNF's migration and re-instantiation mechanism to achieve carrier grade-aware NFV services in a cloud-based platform. This paper also provides detailed insights on the NFV system modeling, building blocks, and various challenges hindering its cloud adoption. Also, a novel mixed integer linear programming (MILP) optimization model is proposed as a solution to facilitate the NFV platform orchestration in a cloud environment. The model decides between triggering either VNF's migration or re-instantiation while achieving minimal downtime of the VNF, satisfying carrier grade requirements, and finding an optimal placement for the migrated or re-instantiated VNF that minimizes the SFC delays. The proposed model is compared to two availability-agnostic greedy algorithms. The simulation results show that finding an optimized decision whether to migrate or re-instantiate a VNF while associating it with an optimal placement can achieve a minimal VNF's downtime and SFCs delays and can enhance the quality of service accordingly.**

Keywords— Network Function Virtualization, Software Defined Networking, Cloud Computing, Service Function Chain, Network Softwarization.

I. Introduction

The rapid increase in demand for network connectivity generated by the mobile computing devices has imposed various challenges on the network service providers (NSPs). NSPs are trying to keep pace with the connectivity demands while maintaining the required quality of service (QoS) conditions. To realize the prospected vision, NSPs perceive the need for programmable infrastructure that can be automated to deliver flexible user-application-centric services. However, achieving fully automated programmable networks can pose an exhausting budget load. Virtualization technology presents an intriguing solution for this challenge. It is rapidly becoming a staple in information technology (IT) industry as a guarantee of lower footprint and efficient utilization of computing resources. To exploit the advantages of virtualization, a group of NSPs with the European Telecommunications Standards Institute (ETSI) reveals their network function virtualization (NFV) solution [1]. NFV enables the migration of network functions from the expensive proprietary hardware to software applications termed as virtual network functions (VNFs), which use commercial-off-the-shelf (COTS) infrastructure [2]. When combined with the implementation flexibility of the cloud services and the programmable network dynamics of software-defined networking (SDN), NFV can inherit the advantages of the virtualization technology [3][4].

As NFV services are carrier grade in nature, NSPs face various challenges to satisfy the carrier grade requirements for cloud-based NFV applications, such as high availability (HA), performance (low latency), and QoS. In terms of HA and performance, carrier grade services should aim at achieving five nines (99.999%) or more of service availability while ensuring very low latency especially for mission-critical applications. This means undergoing less than six minutes of downtime per year whether it is planned or unplanned outage and minimizing the violations of the service level agreements (SLAs). Although implementing NFV-cloud services can subsidence the NSPs' capital expenditure (CAPEX) and operational expenditure (OPEX), this practice introduces various system orchestration challenges that defy the carrier grade anticipated availability and performance requirements.

VNFs can undergo different planned and unplanned outages (such as maintenance, natural disasters, and overload). To this end, different mechanisms can be used to mitigate these issues, such as migration and re-instantiation. However, achieving a mandate that maintains the performance and the availability of VNFs' services requires an intelligent orchestration paradigm that chooses either to migrate or to out-scale (re-instantiate) the VNFs. Each of these techniques is associated with its own delay overhead costs that contribute to the system downtime and latency. For instance, migration technique is a preferable solution when application states should be reserved to achieve service continuity. It is also used for the applications that efficiently use the scaling resources, such as monolithic application. As for re-instantiation technique, it is a preferable solution when it is not necessary to preserve the application states, such as stateless microservices application components. However, the migration or re-instantiation decision does not only depend on the application type, but it is also affected by other factors, such as the rebuild delays of the associated application state and the governance registration delays. Therefore, the non-optimized decision of migration or re-instantiation affects the application downtime and latency. In turns, this practice generates violations of the carrier grade

978-1-5090-4026-1/17/$31.00 ©2017 IEEE

requirements, service degradation, and loss of revenue. As for NFV applications, they are represented by a computational path (SFC). The latter is the path where a user data bearer, application's request, or desired service should follow a chain of intercommunicating (dependent) VNFs until it is successfully processed. It is illustrated in Fig. 1. Therefore, the traditional application's migration and re-instantiation techniques of the cloud are not applicable to NFV applications. To this end, an intelligent NFV-aware orchestrator should be introduced to capture the NFV application requirements and constraints, such as the SDN network convergence delays and the SFC rebuilt delays.

To address the inadequacies of NFV-aware orchestration, this paper introduces a novel NFV-aware orchestrator using a mixed integer linear programming (MILP) optimization model. Once a sudden interruption (resource scaling or failure) affects a VNF status, the optimization model captures the VNF's functionality, latency, and availability constraints and generates an optimal decision (either migration or re-instantiation) that minimizes the VNF's downtime and SFC latency. Since migrating or re-instantiating a VNF means that a new server should be selected to host it, the proposed model generates an optimal VNF placement that satisfies the functional and non-functional constraints and minimizes the delay of the computational path (SFC). To this end, the proposed approach enhances the QoS by minimizing the VNF downtime and satisfying the carrier grade requirements (availability and performance) of the provided services. The main contributions of this work are summarized as follows:

- Propose an intelligent orchestrator that selects the best decision (migration or re-instantiation) to resume a VNF workload while minimizing the downtime of the VNFs.
- Capture the carrier grade's functionality constraints that affect the SFCs of the NFV application (the overhead delays of the migration and re-instantiation processes, such as service governance and SDN convergence delays).
- Capture the VNFs' dependencies constraints that constitute successful SFCs.
- Minimize the delay of the computation path of the SFC. This objective is achieved by finding an optimal placement for the migrated or re-instantiated VNF.

The rest of this paper is organized as follows. Section II presents related work for NFV orchestration in a cloud environment. Section III describes the proposed approach where system modeling and orchestration of the NFV platform are discussed. In Section IV, the proposed optimization model and its constraints are defined. The system evaluation and simulation results are discussed in Section V. Finally, the conclusion and future work are defined in Section VI.

II. Background

Nowadays, NFV services and SDN are capturing the interest of researchers in academia and industry. Several approaches are proposed to address the NFV challenges. Eramo *et al*. propose a consolidation migration technique for VNF instances that minimizes the revenue loss [5]. Although the authors define the cost of migration in terms of the network bandwidth and the SFC direct dependencies, they discard different functional and non-functional constraints, such as SDN network convergence and the delay tolerance of the VNF dependencies. Also, they do not consider optimizing the placement of the migrated VNFs. Cohen *et al.* propose a near optimal NFV placement technique [6]. However, the proposed algorithm has a limited problem scope. The authors only consider monolithic NFV components while overlooking inter- or intra-dependencies and service chains. Mehraghdam *et al.* introduce an optimization model that enhances the count of the service chains for a given set of VNFs [7]. Mijumbi *et al.* introduce an online SFC greedy orchestration algorithm to efficiently schedule VNF in an NFV platform [8]. Addis *et al.* propose an optimization model that aims at achieving efficient CPU utilization for VNFs in SFCs [9]. Ayoubi *et al.* propose a cut-and-solve based approach to optimally solve the VNF assignment problem [10].

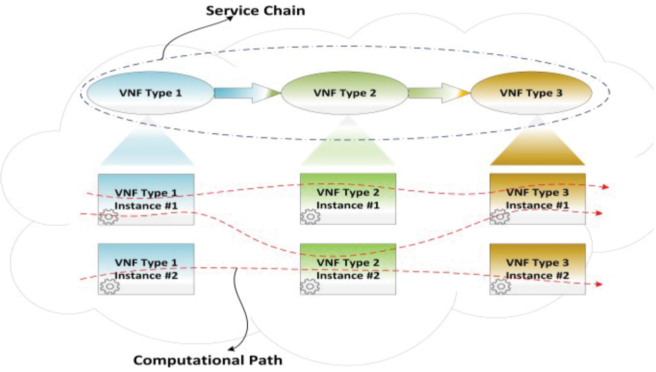


Fig. 1: Some possible computational paths of the service function chain.

Although the proposed approaches address NFV orchestration challenges, each solution focuses on either migration or re-instantiation. They discard the necessity of a model that can select the best carrier grade-aware technique. Also, they overlook various constraints that affect the SFC of the NFV platforms, such as the overhead cost of the VNFs migration or re-instantiation decision, inter/intra- dependencies, multi-tiered VNFs delay tolerance, SDN network convergence, and service governance discovery delays. To mitigate these inadequacies, this paper proposes an intelligent orchestrator that uses MILP model to show the impact of migration and re-instantiation on the VNFs' downtime and SFCs' delays and select the best approach accordingly.

III. Approach

NSPs are in pursuit to make the best of the cloud service models for hosting NFV applications to overcome challenges and realize the Telco-cloud. Cloud service providers (CSPs) offer the cloud users different Software, Platform, and Infrastructure as a Service (IaaS, PaaS, and SaaS) models to build their services. However, it is up to these users (NSPs) to design and orchestrate their NFV applications to achieve any desired objectives. For instance, Amazon Web Services are used to serve the hyper-scale user base of Netflix, which is responsible for 35.2% of North America networking traffic as of 2015 [11]. Netflix introduces various tools and technologies (e.g. Eureka) to achieve its application desired QoS, performance, and HA. This section defines the system modeling of the NFV platform and the VNFs carrier grade requirements and discusses the migration and re-instantiation solutions and challenges.

A. System Modeling

The first step toward achieving an efficient NFV platform in the cloud is to define the building blocks and the granularities of this platform. In the following, the NFV platform/application is described to extract and reflect the different platform entities that affect the carrier grade's metrics. Besides, the VNFs inter- and intra-dependencies relations of an NFV application are identified for a better understanding of the VNFs' interactions.

1) Service abstraction layer

While the monitoring services of the CSPs expose various metrics of the cloud-based applications' entities, the cloud users (NSPs) are responsible for these metrics' interpretation as well as the management of the application's entities [12]. With this in mind, maintaining the QoS of the NFV cloud-based applications becomes a joint responsibility between the CSPs and NSPs. The CSPs offer the virtual machines (VMs) and containers placements that account to the requirements of the NFV application, and the NSPs should deploy and orchestrate their applications to comply with the carrier grade standards.

2) Carrier grade requirements

These requirements capture the performance and availability considerations that can facilitate the adoption of an NFV application.

a) Performance requirements of NFV applications

Designing a performance-aware NFV application can be achieved using an optimal and intelligent management of the VNF entities. In that case, the management approach can exploit the scalability characteristics of the cloud environment, such as vertical and horizontal resources scaling. The vertical scaling of resources is the process where more computational resources (virtual central processing unit (vCPU), memory, and/or storage capacity) are assigned to the same virtual environment (VM or container). This scaling improves the performance of the VNF instance, but it is limited to the physical server resources. As to horizontal scaling, it is the process where a new instance of a VNF is instantiated. Although this type of scaling maximizes the service reliability, it is associated with different challenges, such as placements' management of the instances, workload distribution among these instances, and maintaining the interdependency and redundancy relationships of the VNFs.

b) Availability requirements of NFV applications

In order to meet the service availability requisites of the carrier grade, NFV should ensure resiliency to failure and service continuity. Failure resiliency can be provided by sanctioning an automated on-demand mechanism in the NFV framework to reconstitute the VNF after a failure. As for the service continuity, it can be provided by an instant data recovery (VNFs migration) with non-observable state loss. Concurrently, the NFV orchestrator manages the VNFs' resources and deployments based on the network demand to meet the desired QoS.

Achieving a carrier grade service aims at providing performance and availability-aware computational paths. For example, if an active healthy component becomes faulty, its requests failover to another healthy component of the same type. As a result, the computational path maintains the desired availability while relaxing the component availability requirements.

To this end, the NFV platform relies on an optimal and efficient orchestration of its VNFs to satisfy the carrier-grade (low latency and HA) requirements. The carrier-grade-aware orchestration is associated with live near real-time VNF dynamic scaling and reconstitution. The latter tasks are performed using either migration or re-instantiation techniques. Therefore, an intelligent orchestrator is needed to select the best carrier grade-aware technique that provides seamless services and avoids/minimizes SLA violations.

3) VNF placement and networking considerations in cloud

A cloud consists of interconnected data centers (DCs) that are distributed across different geographical regions. Each DC consists of multiple racks that are intra-connected through switches. Each rack hosts a set of servers with various resources configurations. These servers are grouped in shelves and connected through the top of the rack (TOR) switches. It is necessary to note that the infrastructure topology affects the networking latency between the servers. The cloud orchestrator then generates a mapping between the VMs/containers and the servers. VNF instances are in turn executed within the VMs and/or containers. The services of an NFV application are provided by chaining various VNFs. This chain determines the dependency between different VNFs. Each dependency relation is associated with delay tolerance and communication bandwidth attributes. These attributes define the maximum allowed latency between the chained VNFs to maintain QoS.

The signaling traffic between the cloud servers is defined by the criterion used to allocate the VMs and containers on them [2]. A suboptimal allocation can impede the functionality of the carrier-grade applications. Also, the SFC's routing decisions are directly affected by the VNFs' allocations. Although a basic standard for the architecture of the NFV framework is defined by the ETSI, the latter does not provide a module to facilitate the VNFs' placement management [13]. Since the VNFs' placement is much more complicated than that of cloud applications, the legacy orchestration techniques are not sufficient for NFV applications. This issue remains a "millstone around the neck" of the NFV orchestrator that can be mitigated using an optimized VNFs-to-hosts mapping model.

To this end, the VNF placement and orchestration decisions (migration or re-instantiation) are directly correlated. Once a migration or re-instantiation process is triggered, a new hosting server should be allocated for the affected VNF. The allocation stage reshapes the SFCs of the NFV platform and affects their delays accordingly.

4) Live migration and re-instantiation techniques

Once a VNF service is affected, the orchestrator triggers live migration or re-instantiation mechanism to maintain the carrier grade requirements (HA and low latency). Both mechanisms can be triggered upon scaling up or down of resources, (un)scheduled maintenance, or faulty nodes.

a) Live migration mechanism

Live migration is the process of moving a VNF from its original server to another one without interrupting its activity. This mechanism migrates the states of the VNF's resources to the destination server.

b) Re-instantiation mechanism

Re-instantiation is the process of initializing a recovery or a new healthy image that has the same type as the affected VNF. In this process, the states of the hosting environment are reset and associated with a cold boot of the VNF.

The decision of performing a live migration or re-instantiation technique is affected by various constraints. As NFV is an emerging technology accompanied with SDN, traditional networking constraints should be redefined to reflect the networking evolution, such as the SDN controller placement and the network convergence. Also, the SFCs of NFV is associated with microservices application architecture, which introduces additional constraints to the NFV platform models. In the following, we introduce an optimization model that aims at satisfying the aforementioned requirements and challenges. The model also intelligently decides between live migration and re-instantiation while enhancing the QoS, minimizing the service interruption of an NFV services and SFCs' latency, and finding an optimal carrier grade-aware placement of VNFs.

IV. Optimization Model Formulation

The proposed model is solved using IBM ILOG CPLEX optimization tool [14]. The model aims at minimizing the downtime of migrating or re-instantiating a VNF while satisfying different placement, availability, and re-instantiation/migration constraints.

1) Computational resources constraint

Using this constraint, the proposed model selects a set of servers that can satisfy the VNFs' resources demand. In this model, the resources are CPU cores and memory.

2) Network delay constraint

Using this constraint, the proposed model filters the servers to select the ones that do not violate the delay tolerance between the dependent VNFs in an SFC.

3) Availability constraints

Each VNF can be either a sponsor and/or a dependent one. In order to maintain the availability of the SFC chain, the proposed model defines the following constraints:

a) Affinity constraint: This ensures that the sponsor VNF and its dependents should be hosted on the same server if the dependents have tolerance time lower than sponsor's recovery time.

b) Anti-affinity constraint: In contrary, the dependent VNFs and their sponsor should be deployed on different servers if the dependents have higher tolerance time compared to their sponsor's recovery time.

4) SDN network controller convergence constraint

Using this constraint, the model selects a set of servers that minimizes the convergence delay of the SDN network controller. This delay is the time needed by the controller to reflect the changes (such as new VNFs' placements) in the computational path of the VNFs of a SFC in case of migration or re-instantiation process.

5) Service discovery delay constraint

Using this constraint, the model selects a set of servers that minimizes the service discovery delay. The latter is generated from the VNFs' migration or re-instantiation process. It is defined as the VNF registration time with a service broker, which is responsible for collecting and maintaining meta-data information of the federated VNF cluster.

A. Notations and decision variables:

In this model, the set of VNFs is denoted as V, the total number of VNFs is denoted as N_v, the set of servers is denoted as S, the total number of servers is denoted as N_s, the computational resources are denoted as Res, the set of computational resources types is denoted as R, the SDN controllers set is denoted as C, and the set of dependent VNF is denoted as V^D. The original placement of the VNFs is denoted by $X^{original}$. Also, the tolerance time and recovery time are denoted as T^T and T^R respectively. SO and CO represent the hosting server's delay overhead and the network convergence delay overhead of the selected SDN controller respectively. As for delays, the delay generated from the VNF placement is denoted by D^p, the delay between server S and S' is denoted by $D^{SS'}$, the delay between the hosting server and the SDN controller is denoted as D^{CS}, and the delay resulting from the overhead of migration or re-instantiation decision is denoted as D^{Dec}. Note that Dec represents either a migration or re-instantiation decision. As for the binary decision variables, they are defined as follows:

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

$$Y_v^{Dec} = \begin{cases} Y_v^{Migration} = 1 & if\ VNF\ "v"\ will\ be\ migrated \\ Y_v^{Migration} = 0 & otherwise \\ Y_v^{Re\text{-}Instantiation} = 1 & if\ VNF\ "v"\ will\ be\ re\text{-}instantiated \\ Y_v^{Re\text{-}Instantiation} = 0 & otherwise \end{cases}$$

B. Mathematical Formulation

The objective function is:

$$min \quad \sum_{v}^{N_V} DownTime_v \tag{1}$$

It is subjected to the following constraints:

- *Boundary Constraints:*

$$X_{vs}, Y_v^{Dec} \in \{0,1\} \quad \forall\ v \in V,\ s \in S \\ Dec \in \{Re\text{-}Instantiation, Migration\} \tag{2}$$

$$DownTime_v \geq \quad 0 \quad \forall v \in V \tag{3}$$

- *Placement Constraints:*

$$\sum_{v}^{N_V}(X_{vs} \times Res_{vr}) \leq Res_{sr} \quad \forall\, s \in S, r \in R \tag{4}$$

$$\sum_{s}^{N_S} X_{vs} = 1 \quad \forall\, v \in V \tag{5}$$

- *Availability Constraints:*

$$\begin{gathered}(X_{vs} + X_{v's}) \leq 2 \quad or \quad (X_{vs} + X_{v's}^{original}) \leq 2 \\ \forall\, s \in S, v \in V, v' \in V^D, T_{v'}^T \leq T_v^R\end{gathered} \tag{6}$$

$$\begin{gathered}(X_{vs} + X_{v's}) \leq 1 \quad or \quad (X_{vs} + X_{v's}^{original}) \leq 1 \\ \forall\, s \in S, v \in V, v' \in V^D, T_{v'}^T \geq T_v^R\end{gathered} \tag{7}$$

- SFC *Delay & Re-Instantiation/Migration Constraints*:

$$Y_v^{Re\text{-}Instantiation} + Y_v^{Migration} = 1 \quad \forall\, v \in V \tag{8}$$

$$D_v^P = X_{vs} \times \left[\left(\sum_{d}^{N_S} X_{vd}^{original} \times D_{sd}^{SS'} \right) + D_{cd}^{CS} \right] \quad \forall v \in V, s \in S, c \in C \tag{9}$$

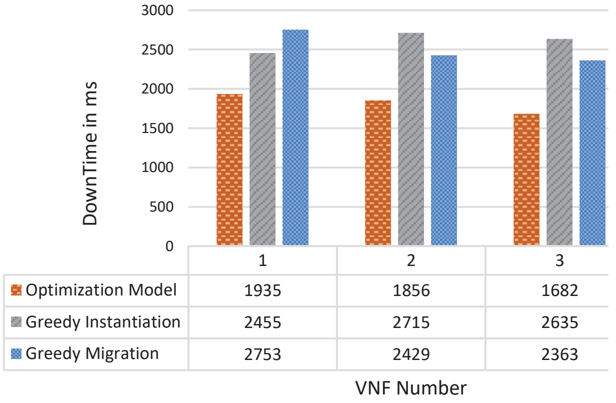
$$D_v^{Dec} = (SO_v^{Dec} + CO_c^{Dec}) \times \mathrm{Y}_v^{Dec} \quad \forall c \in C, v \in V \tag{10}$$

$$\begin{gathered}DownTime_v = D_v^{Dec} + D_v^P \quad \forall\, v \in V, \\ Dec \in \{Re\text{ - }Instantiation, Migration\}\end{gathered} \tag{11}$$

Constraint (2) determines that the placement and re-instantiation/migration decision variables are binary numbers. Constraint (3) determines that the VNF downtime should be a positive number. Constraint (4) determines that the servers should have enough computational resources to host the re-instantiated or migrated VNF. Constraint (5) determines that only one server can host a VNF. To maintain the interdependency relationship between different VNFs, constraint (6) determines that a VNF shares the same server with its dependent VNF(s) if the latter cannot tolerate the absence of their sponsor VNF. In contrary, constraint (7) determines that a VNF and its dependent(s) should share different servers if the dependent(s) can tolerate the sponsor's absence. Constraint (8) determines that a VNF can be either migrated or re-instantiated. Constraints (9) and (10) determine that a VNF should be placed on a server that satisfies the delay requirements while minimizing the migration or re-instantiation overheads. Based on the previous constraints, the model selects either migration or re-instantiation of a VNF while minimizing its downtime. Therefore, constraint (11) shows that the downtime of each VNF is calculated in terms of the placement latency and the overhead delay resulted from either the migration or the re-instantiation process.

V. Simulation Results

The testbed is implemented on a VM that consists of 12 vCPUs and 32 GB of memory. The model is evaluated on a three-tier NFV application where each tier consists of two VNFs, and each VNF consists of one VNF component. The model setup consists of 35 hosting servers, two SDN controllers, and six VNFs. The results of the optimization model are compared with two greedy algorithms to show the impact of the optimized NFV-aware orchestration on the VNF's downtime and the SFCs' delay. The first algorithm is an availability-agnostic migration algorithm. This algorithm generates a set of servers that can satisfy the migration functional constraints (computational resources requirements) and allocate the migrated VNFs on them. As for the greedy availability-agnostic re-instantiation algorithm, it finds a set of servers that satisfies the re-instantiation functional constraints to instantiate new VNF instances that have the same type as the affected VNF(s). In the following, the VNF's downtime and the delay between the instances of the VNF's SFC are the metrics used to compare the proposed MILP model with the above algorithms.



	1	2	3
▨ Optimization Model	1935	1856	1682
▨ Greedy Instantiation	2455	2715	2635
▨ Greedy Migration	2753	2429	2363

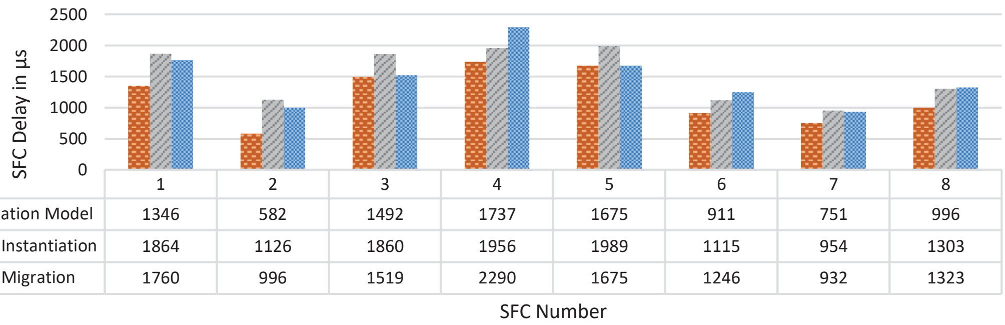
Fig. 2: Downtime of each VNF.

A. Downtime comparative analysis

The proposed MILP model is compared to the availability agnostic migration and re-instantiation algorithms. The results are shown in Fig. 2. The proposed model captures different delay, availability, and placement constraints. It then chooses either to migrate or to re-instantiate the three affected VNFs. The model selects the mechanism that minimizes the VNF's downtime while satisfying availability and functionality constraints between the dependent VNFs. In this simulation, the model decides on re-instantiating the first VNF and migrating the others. Therefore, the proposed model has the lowest downtime values compared to the other algorithms. As for the availability-agnostic migration algorithm, it migrates the affected VNF to a new server that satisfies the computational resources constraints while overlooking any availability or other performance requirements. Similarly, the availability-agnostic re-instantiation algorithm generates new instances of the affected VNF on the servers with enough computational resources regardless of any other QoS requirements.

B. SFC delay comparative analysis

Once a VNF service is affected, a new placement should be generated to re-deploy it whether it is migrated or re-instantiated. Therefore, the proposed model does not only decide on either migrating or re-instantiating a VNF, it also searches for the best placement that can satisfy the functional



	1	2	3	4	5	6	7	8
▨ Optimization Model	1346	582	1492	1737	1675	911	751	996
▨ Greedy Instantiation	1864	1126	1860	1956	1989	1115	954	1303
▨ Greedy Migration	1760	996	1519	2290	1675	1246	932	1323

Fig. 3: Delays of the service function chains.

and non-functional constraints of each VNF while minimizing the SFC's delay. Fig. 3 shows the comparative analysis of the SFC delays between the VNF instances of the 3-tier NFV application. As shown in the figure, the proposed model shows the lowest SFC delays compared to the other algorithms. The model generates a pool of servers that satisfies the functional constraints (computational resources and delay tolerance). It then filters the servers to select the one with the minimal delay between the VNFs instances of the SFC. As for the availability agnostic migration and re-instantiation algorithms, they aim at avoiding service degradation between VNFs instances, but they discard performance constraints that minimize the SFC delays. It is necessary to note that minimizing the SFC delays allows the VNF orchestrator to apply various policies on the systems. These policies vary according to the objective of the NSP. For example, the NSP can introduce policies to achieve green or advanced security analysis networks.

The above downtime and delay comparative analysis show that it is necessary to design an intelligent orchestrator that manages the VNFs lifecycles while ensuring a seamless service that avoids the service degradation and minimizes the SLA violations accordingly.

VI. Conclusion

ICT industry is revolutionized with the NFV and SDN technologies. To unleash all the advantages of NFV and SDN, various challenges should be solved. This paper discussed a VNF orchestration problem in an NFV cloud-based platform. It then provided detailed insights on the system modeling and building blocks of a VNF orchestrator. To this end, various challenges hindering NFV cloud adoption were identified and discussed. Furthermore, the paper proposed an MILP optimization model to enhance the NFV cloud-based platform orchestration. The proposed optimization model consists of different performance and availability-aware constraints. These constraints were designed to achieve the minimal downtimes of the affected VNFs. The model also proposed delay constraints that aim at satisfying carrier grade requirements of the SFC and minimizing the delays between different VNFs instances constituting the SFC. Although the proposed model minimizes the carrier grade requirements of the VNF, it is limited by its NP-hardness computational complexity. Therefore, a heuristic solution will be integrated with this model in the future work. The approximate algorithm will aim at solving the problem in a polynomial time while considering the above carrier grade requirements. Also, the proposed methodology can be extended to support other QoS-aware policies (services security and interoperability).

References

[1] ETSI, "Network Functions Virtualisation (NFV); Virtualisation Technologies; Report on the application of Different Virtualisation Technologies in the NFV Framework," *ETSI GS NFV-EVE 004 version 1.1.1*, 2016.

[2] H. Hawilo, A. Shami, M. Mirahmadi, and R. Asal, "NFV: state of the art, challenges, and implementation in next generation mobile networks (vEPC)," *IEEE Network*, vol. 28, no. 6, pp. 18-26, December 2014.

[3] M. Jammal, T. Singh, A. Shami, R. Asal, and Y. Li, "Software Defined Networking: State of the Art and Research Challenges," *Computer Networks Journal*, vol. 72, pp.74-98, October 2014.

[4] T. Taleb, A. Ksentini and R. Jantti, ""Anything as a Service" for 5G Mobile Systems," IEEE Network, vol. 30, no. 6, pp. 84-91, December 2016.

[5] 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 Transactions on Networking*, no.99, pp.1-18, March 2017.

[6] R. Cohen, L. Lewin-Eytan, J. S. Naor, and D. Raz, "Near optimal placement of virtual network functions*," IEEE Conference on Computer Communications (INFOCOM)*, pp. 1346-1354, 2015.

[7] S. Mehraghdam, M. Keller, and H. Karl, "Specifying and placing chains of virtual network functions," *IEEE 3rd International Conference on Cloud Networking (CloudNet)*, pp. 7-13, 2014.

[8] R. Mijumbi, J. Serrat, J.L. Gorricho, N. Bouten, F. De Turck, and S. Davy, "Design and evaluation of algorithms for mapping and scheduling of virtual network functions," *1st IEEE Conference on Network Softwarization NetSoft),* pp. 1-9, 2015.

[9] B. Addis, D. Belabed, M. Bouet, and S. Secci, "Virtual network functions placement and routing optimization," *IEEE 4th International Conference on Cloud Networking (CloudNet),* 2015.

[10] S. Ayoubi, S. Sebbah, C. Assi, "A Cut-and-Solve Based Approach for the VNF Assignment Problem," *IEEE Transactions on Cloud Computing*, vol. PP, no.99, pp.1-1, June 2017.

[11] Sandvine, "2016 Global Internet Phenomena: Latin America and North America," *Report*, 2016.

[12] M. Jammal, A. Kanso, P. Heidari, and A. Shami, "Availability Analysis of Cloud Deployed Applications," *IEEE International Conference on Cloud Engineering (IC2E),* pp. 234-235, April 2016.

[13] ETSI, "Network Function Virtualization: Architectural Framework", http://www.etsi.org/deliver/etsi_gs/NFV/001_099/002/01.01.01_60/gs_NFV002v010101p.pdf, 2013.

[14] IBM, "CPLEX Optimizer," https://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/, May 2017.