Exploring Microservices as the Architecture of Choice for Network Function Virtualization Platforms

Hassan Hawilo , Manar Jammal, and Abdallah Shami

Abstract

NFV is an emerging key technology that overcomes many challenges facing network service providers, such as reducing the capital and the operating expenses and satisfying the growing demand for mobile services. Integrating NFV with MEC and cloud environments requires an architecture that enables efficient implementations and deployments of NFV entities. Microservices architecture is a promising implementation of service-oriented architecture with recognized advantages in terms of modularity and continuous delivery. This article envisions microservices architecture as the solution of choice for building NFV platforms that are hosted in a dynamic environment ranging from MEC to cloud environments. This article addresses the major challenges and requirements of the microservices architecture to fully-exploit the potential of its adoption in NFV. It also proposes potential solutions that alleviate these issues. The article also discusses the need for agile and modular NFV entities along with MEC to realize various applications. To this end, the article discusses explicitly a novel NFV microservices entities scheduler optimization model. The proposed scheduler aims at minimizing network delays while taking into consideration various functional and non-functional constraints. The evaluation of the simulation results demonstrates that the proposed model minimizes the computational paths' latencies and improves the performance and availability of the NFV service chains.

Introduction

Network service providers (NSPs) are certainly facing challenges in satisf ying the rapid increase in network connectivity demands while maintaining the required quality of service (QoS). Also, over-the-top application providers are continuously harvesting the traditional NSPs' revenue streams. These changes in the competition landscape narrow the return-on-investment margin and overwhelm the networking infrastructure of the NSPs. With the inevitable presence of networking infrastructure in any application stack of information and communications technology (ICT), NSPs leverage their ability to deliver reliable service and enhance extensive customer intimacy to explore new business opportunities. This can increase NSPs' average revenue per user. NSPs are also seeking accretion of new applications into their service models to enhance and expand their enterprise services portfolio beyond the connectivity realm. To achieve this desired vision, NSPs have projected the need for a programmable automated infrastructure that drives real-time, flexible, and user-application-centric network connectivity services. However, the dependency of the current network on an extravagant proprietary complicated infrastructure prevents the NSP from realizing automated programmable networks without overwhelming their capital and operating expenditure (CAPEX and OPEX) budgets. Virtualization technology emerges as an intriguing solution for this challenge. Virtualization technology was originally introduced as a solution to achieve a smaller footprint and efficient utilization of computing resources in enterprise data centers (DCs). To this end, NSPs investigate the opportunity to employ virtualization within their infrastructure to lower their CAPEX and OPEX investments.

A major milestone was reached when a group of NSPs under the European Telecommunications Standards Institute (ETSI) introduced network function virtualization (NFV). NFV is the technology that migrates the networking functions from the proprietary hardware to virtual network functions (VNFs). The latter is implemented as software applications running on commercial-off-the-shelf (COTS) information technology (IT) infrastructure. NFV utilizes various IT virtualization techniques based on commodity hardware (computing resources, storage, and networking) to consolidate network function applications. This consolidation enables NSPs to take advantage of the lower cost and innovative dynamics of traditional IT infrastructure. In that context, a powerful companion technology to NFV is software-defined networking (SDN), the technology that introduces real-time network programmability. With the effective integration between these two paradigms, NSPs can expect major improvements in component modularity and implementation agility. This improvement will have a direct impact on CAPEX, OPEX, and time-to-market applications releases, and rapid innovation will emerge in the ICT industry.

In the ETSI definition of the basic architecture standards for the VNFs, each VNF consists of one or more virtual network function components (VNFCs) [1]. VNFCs implement various functionalities that provide the service defined by the

Digital Object Identifier:
10.1109/MNET.2019.1800023

The authors are with Western University, Canada.

0890-8044/19/$25.00 © 2019 IEEE

VNF descriptor (VNFD). This architecture allows the standardization group to have well-defined interfaces for the VNFs' services while granting the VNFCs implementation freedom to the VNF software providers. Having well-defined standard interfaces of VNFs provides stable software releases while enabling interoperability of VNFs between various software provider vendors. The VNFCs implementation freedom drives the innovation and evolution of the VNF services and provides the capability of flexible management and orchestration of the VNFCs lifecycle based on functional and non-functional constraints.

NSPs intend to deploy NFV services in cloud environments to take advantage of their business and service models, such as pay-as-you-go and scale up or down on-demand. Furthermore, NFV is expected to complement mobile edge computing (MEC) technology to provide accelerated content delivery and better application responsiveness, such as intelligent edge data caching, to enhance the quality of user experience (QoE). MEC has been introduced by ETSI as a technology that enables the deployment of services and applications in the edge network to achieve the closest proximity to the end user [2]. With these intentions, new software development perspectives should be adopted by NFV to ease the VNFs deployments and their integration with the cloud and MEC environment.

Since VNFs are constructed by chaining various VNFCs to provide networking services, this article envisions microservices architecture, the emerging implementation of service-oriented software architecture (SOA), as the solution of choice for developing a VNF. In the foreseen design, each VFNC is a microservice component by itself. Microservices architecture allows the VNFs services to be more flexible in the hosting environment where the virtualized functionalities can adopt various manageability scopes to meet functional and non-functional constraints. To fully exploit the potential of adopting the microservices architecture in NFV, it is necessary to define the major challenges introduced by this architecture and address them accordingly.

This article discusses the adoption of microservices architecture in NFV and provides a guideline to design a placement scheduler for the VNFCs. The main contributions of this work can be summarized as follows:

- Define the major challenges of adopting microservices within NFV platforms.
- Define the requirements for microservices architecture to fully-exploit the potential of its adoption in NFV.
- Propose potential solutions that alleviate the challenges of adopting microservices within NFV platforms.
- Discuss explicitly a novel optimization model for the NFV microservices entities' scheduler. The model aims at minimizing the computational paths network delays while taking into consideration various functional and non-functional constraints.

The rest of this article is structured as follows. The next section gives a conceptual definition of microservices architecture. We then discuss microservices architecture in NFV. Following that we describe the role that microservices-NFV is playing to complement MEC for enabling various mobile edge applications. Then, challenges of NFV implementing microservices architecture are introduced. Following that, the modeling of VNFCs placement scheduler technique is explained. We then present and discuss the simulation results of the designed scheduler. Finally, we conclude the article.

Microservices architecture has evolved to mitigate monolithic architecture challenges by introducing distributed systems with lightweight components. Each component performs a specific workload in an independent manner. Components are defined as microservices in this architecture.

Microservices Architecture

In the last decade, the ICT industry has witnessed major breakthroughs in terms of how the world interacts and exchanges information. With the inevitable dependency on mobile smart devices that ranges from personal use to Internet of Things (IoT) connected devices, new paradigms of applications have emerged, such as social media, video-on-demand applications, and software as a service. These paradigms are associated with the advances in computing resources services. Cloud computing accompanied with virtualization is introduced as an infrastructure foundation to meet the rapid and increasing demands of computing resources with minimal CAPEX and OPEX investments. Adopting cloud computing services in an application development requires remodeling the application architecture to exploit the benefits of cloud services, such as scaling on-demand.

Traditionally, web-based applications are developed using a monolithic architecture. The latter is a software with a vertically integrated stack that is executed in a single process. This practice of software development facilitates application deployment and networking where multiple instances can easily reside behind a load-balancer to satisfy the application service demands. However, the change in the application nature and the increase in the complexity and demand of the provided services introduce various challenges to monolithic applications. The tightly coupled codebase is typically a result of the monolithic application, which imposes high-risk associated with any code change or addition of new features. Applying any change to one component can seemingly affect the whole system functionality. Moreover, the monolithic application does not support component reusability, which hinders the scalability of an individual component. This can cause an inefficient utilization of computing resources.

Microservices architecture has evolved to mitigate monolithic architecture challenges by introducing distributed systems with lightweight components. Each component performs a specific workload in an independent manner. Components are defined as microservices in this architecture. A microservice is a kind of software that is contained in its process and typically uses web-based protocols, such as transmission control protocol (TCP), hypertext transfer protocol (HTTP), or remote procedure call (RPC) protocol to communicate. Although microservices archi-

tecture is proposed as a solution to have efficient scalable distributed systems, it introduces new challenges.

Microservices and NFV: A Match Made with Modularity Cloud9

Leading ICT equipment vendors have rushed to build and release various proof-of-concepts designs and prototypes of VNFs running on COTS computing resources. However, these prototypes are based on traditional network function development and monolithic stack development that can only scale vertically and are limited to the computing performance of the underlying bare-metal servers [3]. Since networking functions' applications thrive on the power of computing resources, NSPs are faced with the challenge of re-engineering VNFs to enable horizontal scaling. Being in the process of fully adopting cloud computing to build the telco-cloud, NSPs aim at adopting the best performing architectures in the web-scale development world where scalable and distributed applications reside, such as Amazon, Google, and Netflix platforms. Microservices architecture is considered the best-fit architecture to help NFV achieve its goals. Defining VNFCs as microservices provides the following advantages:

VNFC Bounded Context: Each VNFC performs a limited set of functionalities, which results in a small code base limiting the scope of bugs. Furthermore, the standalone nature of microservices facilitates direct testing of functionalities in isolation with respect to the VNF provided service.

VNFC Modularity: This means gradual transitions to updated versions of VNFCs. The newer versions of VNFCs can be deployed simultaneously with the old ones. The VNFCs that depend on the old versions can be gradually modified to interact with the updated VNFCs, which is known as rolling upgrade. With this approach, NFV can adopt VNFCs' continuous integration and can greatly ease VNF software maintenance.

VNF Innovation and Evolution: By exploiting the independency characteristic, new NFV microservices can be easily introduced to the production services without disrupting their operations.

VNF Flexibility and Scalability: VNF building blocks, VNFCs, can be scaled up or down independently according to the service demand.

VNFC Interoperability: With microservices architecture, VNFCs can be deployed in a heterogeneous manner. Various VNFCs provided by different vendors or developed using different programming languages and frameworks can still be interconnected as long as they implement the right communication interfaces.

Microservices NFV and Mobile Edge Computing

Designing high bandwidth networks with negligible latency is the intent of the service providers to serve many emerging applications, such as Internet of Everything, device-to-device (D2D) communication, voice-over-LTE (VoLTE), on-demand video streaming (4K and 8K videos), augmented reality, and various Internet protocol (IP) multimedia subsystem (IMS) services. Implementing such broadband mobile networks requires efficient utilization of the assigned spectrum for wireless communication and the distribution network infrastructure. It also requires placing the data-hosting application servers in closest proximity to the end-users to achieve negligible latency. With spectrum being the scarce resource, mobile network service providers are tending to deploy heterogeneous networks where macro and micro base station cells coexist with small base-station cells (pico-cells and femto-cells). Heterogeneous networks enhance spectrum utilization to achieve higher data rates for the end-users (user equipment). Mobile edge computing (MEC) is introduced to minimize the latency of serving data through hosting the application servers with the closest proximity to the end-users, especially data caching servers. In such networks, the substantial growth of signaling traffic on the core network (CN) can be generated due to the reduced cell size and increase in user density and mobility. Signaling traffic growth is flourishing due to the emergence of new services on mobile technology platforms.

Nowadays, on-demand video streaming and social media applications are responsible for 65 percent of mobile data traffic, and it is expected to reach 90 percent by 2022 according to the Ericsson mobility report [4]. Therefore, the existence of applications and data caching servers in mobile edge networks is essential to offload the data traffic from the core network and minimize networking latency while serving the maximum number of users with high bitrates. In current and legacy mobile networks, the application servers and the content data should be accessed from centralized data centers and content distribution network (CDN) nodes. The latter nodes are placed at the mobile core network and the point of presence (PoP) that constrains the backhaul networks. Given the evolution at the level of base stations, D2D, and storage technology, deploying the application and caching servers at macro, micro, pico and femto base stations become feasible. However, flexible, agile, and automated network entities should exist side by side with the MEC entities to achieve the desired application and data caching schemes for the above designs. NFV and SDN are proposed to achieve these objectives for networking entities, but so far, they are examined and researched in the context of monolithic applications. NFV and SDN-based network services and components should be proposed and provided as microservices to scale, complement with MEC, and enable advanced application and data caching deployment criteria. Implementing NFV and SDN networking microservices entities at the network edges offloads the networking orchestration traffic from the core network and enables elastic network federations that can be self-sustained while providing high bandwidth connectivity with negligible latency for the end-users. The centralized core networking entities can then synchronize and orchestrate the network federations' inter-traffic.

Challenges of NFV Implementing Microservices

NFV adopting microservices paves the way for the arrival of the telco-cloud. To ensure a wider adoption of NFV by the ICT industry, NFV should overcome the challenges introduced by the soft-

Challenge	Description	Solution
VNFCs networking complexity	VNFCs as microservices are chained together using various protocols, mainly web-based protocols. This approach can result in a complex network activity that can rapidly increase manageability complexity with a higher risk of network exposure to security issues.	• VNFC application states should be extracted and reserved in data stores (Persistence Centralization). • NFV platform should utilize SDN while implementing the following functions within the controller: – Decentralized governance – Governor units – Network segmentation – Continental federations • VNFCs should be logically grouped into various functional groups and serving units. • An optimal placement of orchestration entities should be provided. • Various VNFCs structures that comply with service availability forum (SA-Forum) standards to achieve the carrier grade high-availability requirements should be defined. • Redundancy models and automated management of the replicas at the network segments level should be provided. • An efficient SDN query collision resolution should be provided. • A virtual centralized network-provisioning layer especially for the operations support system (OSS) should be provided.
VNFCs service discovery	Real-world NFV applications can be decomposed into hundreds of microservices (VNFCs) and tens of thousands of running instances. Service discovery challenge is a major hurdle that can impede the scalability of the NFV applications and platforms.	
VNFCs service monitoring, logging, and meta-data collection	NFV applications are carrier-grade in nature that thrives on high QoS. Real-time metrics and meta-data are needed to be collected and processed on-the-fly to facilitate the NFV orchestration and achieve the desired QoS.	
Infrastructure convergence	Converged infrastructure that drives software-defined infrastructure in modern DC introduces challenges for NFV microservices architecture.	
Routing convergence	Existing routing protocols cannot keep up with the hyper-scale DCs in terms of scalability and efficiency. Supporting NFV applications along with the current load of cloud applications is a challenge for all cloud service providers.	
Placement of VNFCs	The criterion used to place the VMs and containers on physical servers is the main contributor to the increase in the signaling traffic between servers. Therefore, having the optimal allocation for the VNFCs is essential to satisfy the carrier-grade requirements.	

TABLE 1. Challenges and solutions of NFV microservices architecture adoption.

warization of network functions and the development architecture. This would aid NFV in meeting all the expectations of hyper-scaling while satisfying the carrier-grade requirements. Prime challenges include the following issues. Table 1 summarizes this section.

VNFC Networking Complexity

NFV microservices architecture is based on the creation of (as many as needed) small independent VNFCs that are chained together using various web-based protocols. This approach can result in complex network activities that are difficult to manage and rapidly impose a negative effect on network manageability. Real-world applications can be decomposed into hundreds of microservices and tens of thousands of running instances, as is the case with Netflix and Twitter [5, 6]. VNFCs provide networking services that handle various networking traffics and latency-sensitive workloads. Therefore, networking complexity escalates further on various levels. Network chaining complexity is a challenge that NFV-microservices should overcome through intelligent networking management, possibly with SDN integration [7].

VNFC Service Discovery

Despite the benefits that microservices architecture introduces to NFV, VNFC management and development are still intricate challenges. A task such as the deployment of applications is trivial with monolithic applications, but with microservices architecture's additional subtasks, it becomes a complicated job. Software development and information technology operations (DevOps) tools along with containers have become mature enough to automate complicated development on remote servers, such as one-click install applications in cloud environments [8]. However, service discovery of VNFCs is a major hurdle that impedes the scalability of the NFV application and platforms. As VNFCs scale on-demand in a cloud environment, a real-time automated service discovery mechanism should be developed to create dynamic service chains to permit the dynamic scaling of VNFs.

VNFC Service Monitoring, Logging, and Meta-Data Collection

Typical NFV applications are carrier-grade in nature, and they thrive on high QoS. Real-time metrics and meta-data should be collected and processed on-the-fly to facilitate the NFV service entities (VNFs and VNFCs) orchestrations that achieve the desired QoS. Therefore, guaranteeing NFV application QoS is a challenge with microservices architecture. The orchestration and management entities in the NFV platform require clear visibility of the collected system metrics data to perform versus VNFC health checks. Further analysis of VNFC health checks can craft the NFV provided service topology, but any variation in the performance metrics across various VNFCs or NFV infrastructure (NFVI) resources hinders this capability. With the on-demand automated scaling ability and delay sensitive VNFC services, collecting and analyzing the generated metrics and meta-data across the NFV microservices platform to give a holistic view of services chains and networks control flows remain an open issue.

VNFC Security

Implementing VNFCs with microservices architecture presents new security challenges that did not face the traditional monolithic applications. These security challenges are exacerbated due to the extensive usage of various communication channels between all the VNFCs that create more roads for data hijacks and interception while in transit. For instance, establishing mutual trust and distributing component secrets are major security

concerns [9]. Implementing all the security measures on hyper-scale microservices intensifies the security challenges.

Infrastructure Convergence

Convergence of infrastructure is a promising approach currently being utilized in modern DCs to allow ICT service providers to scale their infrastructure with efficient resource utilization [10]. Converged infrastructure drives the software-defined infrastructure in modern DC, such as Google DCs [11]. However, this kind of computing infrastructure is not flawless. Some of the challenges that should be addressed in software-defined infrastructure to enable NFV microservices architecture are as follows.

Computing Resources Convergence: Converged infrastructure includes a variety of computing resources in hosts. Various standards, communication types, file system protocols, and interface buses are used to connect hosts over COTS networking equipment. DC operators have exclusive control rights of the network, leaving users with narrow to no exposure to the control functionalities of the underlying network infrastructure. With this limitation of network control exposure, users cannot optimize VNFCs to the best performance.

Networking Resources Convergence: Converged infrastructure combines all kinds of traffics into a unified infrastructure without any segregated network. This approach of unified network infrastructure imposes risks on high priority traffics. Applying QoS and traffic separation through various networking bearers occurs through network adapters and switch partitioning. Although this approach is a solution, it introduces various manageability and traffic processing challenges, especially in a virtualized environment. In a virtualized environment, the physical network adapters are shared between various applications, such as VNFCs and DC management entities that should deliver their services in real time.

Simply providing more bandwidth in a converged infrastructure is not a solution to host NFV applications. DC infrastructure orchestrators should integrate and expose various network-controlling functionalities to maintain the desired QoS and assure interoperability of VNFCs.

Routing Convergence

Multiple distinct architecture choices can be used when designing a data center. Each aims at minimizing the resources required to suit the needs of cloud service providers. It is imperative that cloud service providers are continuously striving to improve their own hardware and software networking infrastructure. Google has gone the extra mile and developed proprietary networking protocols to manage their traffic routes [11]. Existing routing protocols cannot keep up with its hyper-scaled DCs in terms of scalability and efficiency. Supporting NFV applications along with the current load of cloud applications is a challenge for all cloud service providers. They should take a step back and decide on the conflict resolution techniques to be used. In addition, the adoption of microservices architecture with NFV applications requires new approaches at the levels of network hardware and software infrastructure specification. Previously, the use of local area networks (LANs) was sufficient for enterprises when their servers were placed in close proximity. With the wide adoption of cloud computing infrastructure, VLANs used to meet the network demands and create multiple broadcast domains. However, classic VLANs are limited to a 12-bit ID field, which does not satisfy the hyper-scaling level of cloud demands. This led to the emergence and development of generic routing encapsulation (GRE) and virtual extensible LAN (VxLAN). VxLAN and GRE provide virtual LAN connectivity on a hyper-scale over Layer 3 networks. Layer 3 networking equipment (routers) is grouped into various logical groups called autonomous systems (ASs). The latter usually use the open shortest path first (OSPF) protocol to exchange routing information among group members, and the border gateway protocol (BGP) to exchange information with other ASs. When looking closely at these two techniques, OSPF and BGP have evolved to serve current Internet networks with great success. However, the increase in the number of virtualized applications using virtual machines (VMs) and containers has imposed challenges to the current routing protocols. VMs and containers are entities added and dropped out on the fly to meet the cloud application dynamic workloads. These VMs and containers are mobile; they can migrate from one serving node to another in real time [12]. With these properties, VMs and containers highly rely on network traffic mobility and low-latency. Common routing protocols are yet to be proven to serve efficiently this kind of workload because their routing convergence is measured in seconds. Adding NFV applications to the existing cloud workload can disrupt the underlying network because NFV adds hyper-scale overlay networks served by VNFCs. This begs the question: how can SDN emerge as a solution to pave the way for NFV with hyper-scaling VNFCs? It is a challenge for the SDN controller. A first step would be deploying distributed SDN controllers to handle multiple network federation routing convergence, but this area requires further investigation to converge on implementation techniques.

Inter-connecting and Intra-connecting VNFCs

Classic approaches for connecting network functions on premises are achieved through direct connections or through Layer 2 (L2) switches. However, in a virtualized environment, various inter-connection and intra-connection approaches can be held, as illustrated in Fig. 1:

- Two VNFCs are on the same physical server and on the same virtual switch (vSwitch).
- Two VNFCs are on the same physical server but on different vSwitches.
- Two VNFCs are on different physical servers.

Each of the aforementioned cases of VNFC connections has its own advantages and disadvantages. VNFCs establish virtual connections through the virtual network interface controllers (vNICs), which can introduce various hop spanning trees. Optimized traffic routing and VNFC placements should be used to monitor and minimize network traffic latency. Single-root input/output (I/O) virtualization (SR-IOV) compliant

NICs are considered a solution to eliminate intermediate virtual network hops, but they can hinder the VNFCs' mobility in a virtualized environment.

Placement of VNFCs

The criterion used to place VMs and containers on physical servers is the main contributor to the increase in signaling traffic between servers. VM and container allocation is one of the main factors that affect carrier-grade application requirements such as QoS, reliability, and high availability. Migrating networking functions to VNFC microservices is a challenging process because these VNFCs will be executed either within VMs or within containers running on COTS servers in DCs and should satisfy the strict carrier-grade requirements. Therefore, having the optimal (or as close to optimal as possible) allocation of VNFCs is an indispensable step to satisfy QoS requirements.

ETSI has defined a basic framework architecture that does not have a VNFC placement management entity [1]. The virtualization orchestrator handles the VNFC mapping to hosts. The orchestrator is either managed by the cloud service provider or is delegated to VNFCs' owners. Furthermore, VNFC placement directly affects the service chains' routing decisions. This can have a critical impact on the service level agreements (SLAs) in which the cloud service providers guarantee computing resource performance and availability. However, existing SLAs do not guarantee carrier-grade application performance with five nines (99.999 percent) of service availability, which is a critical requirement for virtualized carrier network functions. Therefore, the cloud tenants should orchestrate the VNFC deployment and management in order to achieve the desired QoS. For example, Amazon web services (AWS) are utilized by Netflix to serve the hyper-scale user base that is responsible for 35.2 percent of North American networking traffic [13]. For Netflix to achieve its desired QoS with high service availability, it has developed and contributed to various open source software entities. The Netflix use case is an example of how cloud tenants can introduce their own optimization techniques and approaches to hyper-scale their applications without sacrificing QoS.

VNFC placement and management are more complex compared to current cloud applications. This means that the techniques used by the leading companies who have developed the cloud application architectures are not sufficient to orchestrate the NFV platforms. VNFCs are networking function services that overlay networks and process networking packets in real time. Therefore, any potential error or service degradation can escalate issues at various levels of the substrate and overlay networks and can disrupt any dependent services. These issues are on the horizon of the IT and DevOps pioneer enterprises. For instance, the cloud services of Apple iCloud, iTunes, and other products faced disruption with an outage of four hours in 2015 due to an internal DNS error [14].

Having schedulers agnostic of NFV application intricacies may result in inefficient VNFC place ments. Considering this, service chained VNFCs can for the same reason be scheduled on hosts where delay constraints are violated. This placement can hinder NFV application services from scaling and offloading traffics between VNFCs. In light of the previous points, it is a necessity to associate the NFV microservices architecture with a carrier-grade NFV-aware scheduler that defines the service chain's computational paths to enhance the scalability and traffic offloading of the application service. The NFV-aware scheduler would optimally be defined as a management entity within the cloud orchestration platforms to ensure that the NFV services can serve a dynamic workload while satisfying all carrier-grade requirements.

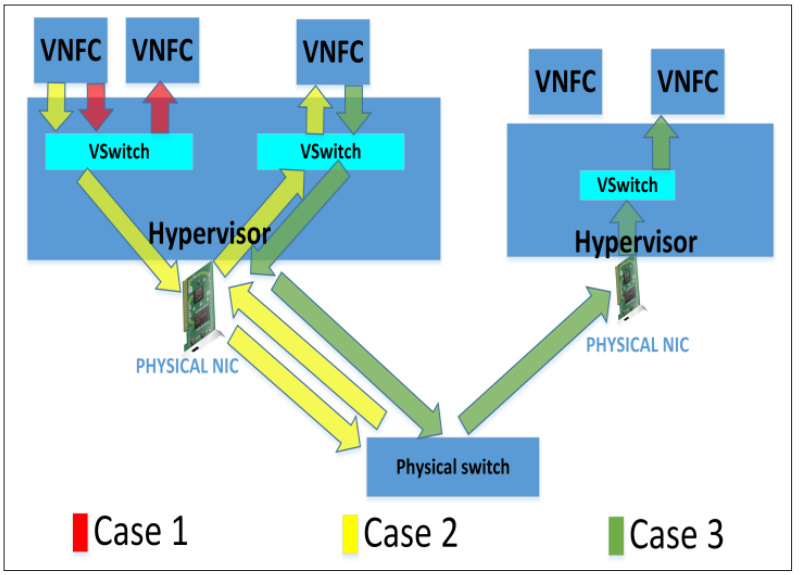

FIGURE 1. Inter- and intra- connections of VNFCs.

VNFC Placement Modeling

In order to provide a scheduling solution that satisfies SLA and QoS requirements, it is necessary to understand the cloud model. The cloud infrastructure consists of interconnected DCs distributed across different geographic areas. Racks are the building blocks of the DC, and they are intra-connected through aggregated switches. They host sets of servers with different resource capacities that are grouped in shelves. Servers belonging to the same rack are connected through the same networking device, top of the rack (TOR) switch. The topology of the network connecting the servers determines the latency constraints between them. By recognizing and modeling various delays between servers, DCs can be divided into different latency zones. As for the VNFC instances, they are executed within VMs and containers that are mapped to the physical servers by the cloud orchestrator. As mentioned in previous sections, NFV applications typically provide their services through various chained VNFs, which are defined as several VNFCs. These chains determine the dependency relations between the VNFCs. The inherited relations are associated with delay tolerance and communication bandwidth attributes that are defined at the abstracted service representation level. The service computational path is restricted by the delay tolerance constraints, which determine the maximum allowed latency between VNFC instances at which this path outage is declared. Therefore, maintaining the maximum number of computational paths requires

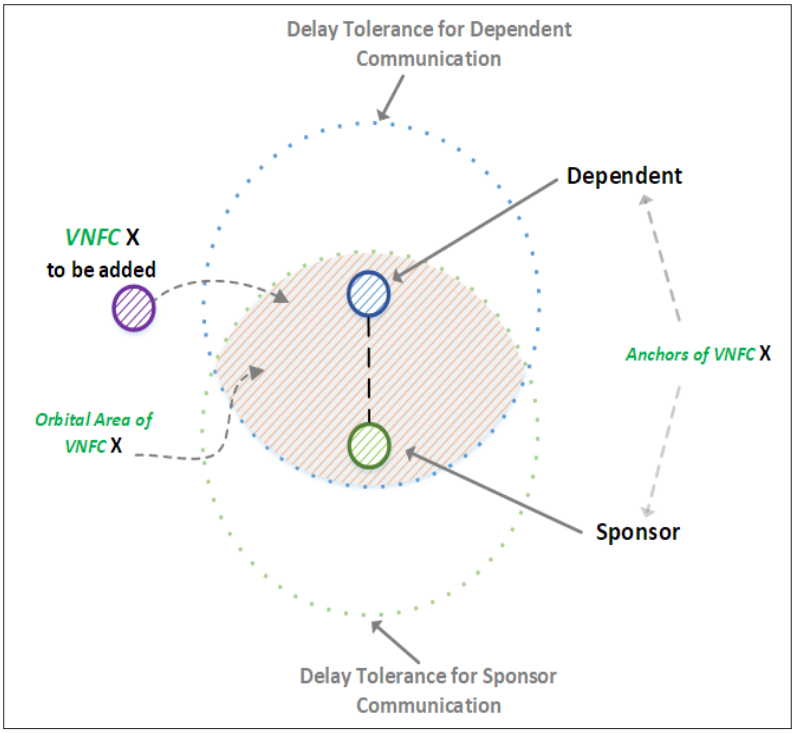

FIGURE 2. The orbital area of a given VNFC.

Input variable	Value
Physical servers	20 servers
MME VNFC	3 instances
HSS VNFC	2 instances
SGW VNFC	2 instances
PGW VNFC	3 instances
Delay tolerance between MME and HSS	320 µs
Delay tolerance between MME and SGW	400 µs
Delay tolerance between SGW and PGW	120 µs

TABLE 2. The model input data.

optimal NFV-aware scheduling models and algorithms. Therefore, the following constraints should be satisfied.

Resources Capacity Constraints: Used to Eliminate Servers That Do Not Satisfy the Resources Demands of VNFC.

Network Delay Constraints: These constraints discard the servers that violate the delay tolerance between VNFCs.

Availability Constraints: These constraints select the servers that satisfy the following:

- Affinity constraint: Defines the set of VNFCs that can reside on the same hosting server.
- Anti-affinity constraint: Defines the set of VNFCs that should reside on different servers. Usually, these VNFCs can tolerate higher outage than the co-located VNFCs.

Redundancy Constraints: These constraints define the number of redundant VNFCs and their redundancy model type. The redundancy models are highly correlated with the cloud environment metrics, such as spin-up time of a VM or container.

Anchors Constraints: VNFC anchors are defined by the functional dependencies that exist between the VNFC microservices. Dependencies may introduce network hierarchy limitations between the VNFC and its anchors.

Orbital Area Constraints: The orbital area is defined by the region where the VNFC can be placed. This area is bounded by the VNFC anchors' constraints associated with the service chain. A VNFC can have multiple peers and dependents in a service chain. Therefore, the orbital areas and distances must be carefully calculated to enable further elastic scalability of the NFV service. Figure 2 illustrates the conceptualization of the VNFCs' anchors in relation to the VNFC orbital area. It demonstrates the placement criterion for a VNFC where the dependents' placements act as anchors and dictate its placement orbital area.

VNFC Placement Simulation

The NFV-aware scheduler should generate optimal placements of VNFCs to pave the way for a carrier grade NFV service. For this purpose, a mixed-integer linear programming (MILP) model is formulated based on the aforementioned constraints and with the following objective function:

$$\text{Minimize} \sum_{m=0}^{N_{MME}} \sum_{s=0}^{N_{SGW}} dMS_{ms} + \sum_{m=0}^{N_{MME}} \sum_{h=0}^{N_{HSS}} dMH_{mh} + \sum_{s=0}^{N_{SGW}} \sum_{p=0}^{N_{PGW}} dSP_{sp}$$

where:

dMS_{ms} = communication delay between $VNFC_m$ of type MME[1] and $VNFC_s$ of type SGW.[2]

dMH_{mh} = communication delay between $VNFC_m$ of type MME[1] and $VNFC_h$ of type HSS.[3]

dSP_{sp} = communication delay between $VNFC_s$ of type SGW[2] and VNFCp of type PGW.[4]

N_{MME} = total number of VNFC instances of type MME.

N_{HSS} = total number of VNFC instances of type HSS.

N_{SGW} = total number of VNFC instances of type SGW.

N_{PGW} = total number of VNFC instances of type PGW.

Virtualized evolved packet core (vEPC) is used as a use case in the simulation. vEPC has been introduced by 3GPP as a simplified all-Internet-protocol (IP) core network architecture [15]. vEPC is developed to unleash the full potential of radio access technologies. It combines the leading IP infrastructure and mobility to enable mobile broadband services and applications. Table 2 summarizes the input data of the model. Given the available computing processing power and the computational complexity of the MILP model, the dataset is defined to generate the simulation results within a reasonable time. The delay tolerances between entities are based on data center network latency measurements as defined in [16].

The MILP model is implemented using the IBM ILOG CPLEX optimization studio, and the greedy algorithm is implemented using Java. A virtual machine with 12 vCPU cores and 64 GB of mem-

[1] MME is the mobile management entity in the EPC.

[2] SGW is the serving gateway in the EPC.

[3] HSS is the home subscriber server in the EPC.

[4] PWG is the packet gateway in the EPC.

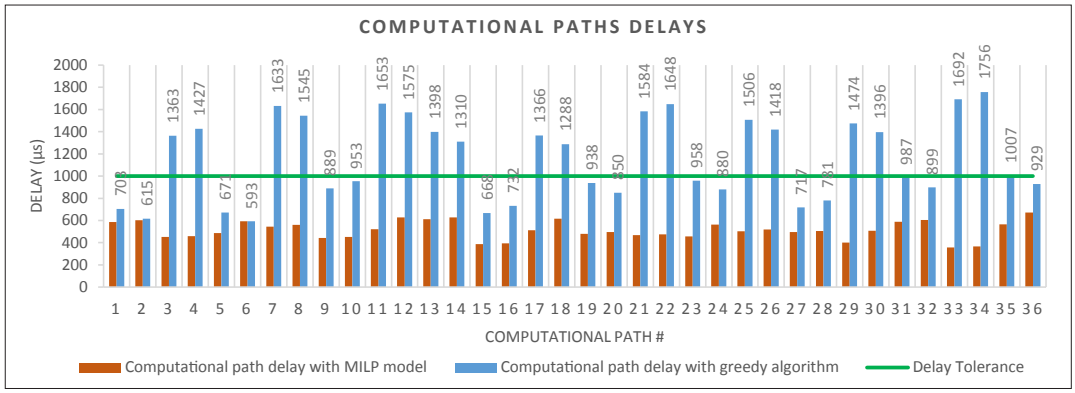

FIGURE 3. Computational paths delays.

ory is used to run the simulation environment. We have compared the NFV-aware scheduler with a greedy algorithm. The corresponding results are shown in Fig. 3 and Table 3.

The MILP model generates the optimal placements that satisfy all the aforementioned constraints while minimizing network delays. These placements maximize the number of available computational paths that represent the VNFC service chains. This objective is achieved by placing the VNFC instances on the hosts with minimum connection delays, which provides valid connections for the computational paths. Increasing the number of computational paths can be quantified by the number of participating members in a functional group of a VNFC instance. All the functional group members should share the same VNFC instance type and reside in the same orbital area. The higher the number of participating members in a functional group, the better it becomes. Table 3 shows the count of the functional groups' members that is generated from the MILP model and the greedy algorithm placements. The MILP model achieves higher functional group member counts compared to the greedy algorithm. The NFV-provided service can achieve better performance and availability using the MILP model placement algorithm than the greedy algorithm. From the perspective of performance, the MILP model allows the functional group to offload traffic between higher VNFC members; however, this is not the case with the greedy algorithm. From the perspective of availability, the MILP model provides better availability to the functional group compared to the greedy algorithm because the MILP model has higher member counts; these members act as redundant components that can take over the workload upon a failure of a VNFC instance.

In addition to the increase in the count of the VNFCs functional group members using the proposed MILP model, the results show that the computational paths' delays are minimized compared to the greedy algorithm as depicted in Fig. 3. Minimizing the VNFC computational paths' delays is paramount for the VNFC management entities. The difference between the delay tolerance and the computational paths' delays allows the management entities to apply various policies on the systems. These policies vary according to the intentions of the network service providers, such as green or advanced security-based analysis policies.

VNFC instances	MILP model functional group members count	Greedy algorithm functional group members count
MME #1	3	2
MME #2	3	1
MME #3	3	2
HSS #1	2	1
HSS #2	2	1
SGW #1	2	1
SGW #2	2	0
PGW #1	3	1
PGW #2	3	2
PGW #3	3	2

TABLE 3. Functional group member.

Conclusion

NFV is the technology revolutionizing the ICT industry by implementing network functions as software based applications running on COTS hardware. It adopts the IT virtualization platform benefits and innovations. The industry and academic researchers are exploiting virtualization technology to simplify and enhance the NFV platforms in order to pave the way for wider adoption by the ICT industry. To unleash all the advantages of NFV, various challenges should be overcome. Therefore, the leading ICT service providers, equipment vendors, and academic researchers should be aware of NFV's challenges and explore new approaches to overcome them.

This article discussed the possibility of adopting microservices architecture in NFV to enable hyper-scaling services. To this end, various challenges were identified and discussed. Anticipated solutions for these issues were provided as well. The article introduced a detailed VNFC placement challenge study and proposed an NFV-aware scheduler design. The proposed scheduler was evaluated in terms of an MILP model to show the potential advantages of optimized VNFC placement in a virtualized environment.

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] ETSI, "Mobile-Edge Computing — Introductory Technical White Paper," Solutions Reference Architecture, 2014.

[3] Intel, "Evolved Packet Core (EPC) for Communications Service Providers," Issue 1, white paper, 2016.

[4] Ericsson, "Ericsson Mobility Report," https://www.ericsson.com/assets/local/mobility-report/documents/2017 /ericsson-mobility-report-june-2017.pdf, Report, 2017.

[5] Netflix, "First NetflixOSS Meetup," http://techblog.netflix.com/2013/02/first-netflixoss-meetup.html , Feb. 2017.

[6] Twitter, "The infrastructure behind Twitter: efficiency and optimization," https://blog.twitter.com/2016/the-infrastructure-behind-twitter-efficiency-and-optimization, February 2017.

[7] Heavy Reading, "Service Chaining in Carrier Networks," White Paper, 2015.

[8] Digital Ocean, "One-Click Install Apps," https://www.digitalocean.com /community/tags/one-click-install-apps?type=tutorials, Feb. 2017.

[9] T. Yarygina and A. H. Bagge, "Overcoming Security Challenges in Microservice Architectures," *Proc. IEEE Symposium on Service-Oriented System Engineering (SOSE)*, Bamberg, 2018, pp. 11–20.

[10] Cisco and NetApp, "Converged vs. Hyper-converged Infrastructures: Understanding the impact on your organization," white Paper, 2016.

[11] NUTANIX, "The Secret to Google's Datacenter and Why it Matters to Every Enterprise," https://www.nutanix.com/2014/02/20/the-secret-to-googles-datacenter-and-why-it-matters-to-every-enterprise/, Feb. 2017.

[12] M. Jammal *et al.*, "Mitigating the Risk of Cloud Services Downtime Using Live Migration and High Availability-Aware Placement," *Proc. IEEE Int'l.Conf. Cloud Computing Technology and Science (CloudCom)*, Dec. 2016, pp. 578-583.

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

[14] CNBC, "Apple: services back after major outage," http://www.cnbc.com/ 20 15/03/11/some-apple-services-suffering-outages.html, February 2017.

[15] ETSI, "Digital cellular telecommunications system; Universal Mobile Telecommunications System (UMTS); LTE; Network architecture," 3GPP TS 23.002 version 11.6.0, Release 11, 2013.

[16] C. Guo *et al.*, "Pingmesh: A Large-Scale System for Data Center Network Latency Measurement and Analysis," *Proc. ACM Conf. Special Interest Group on Data Communication*, Aug. 2015, pp. 139-152.

Biographies

Hassan Hawilo (hhawilo@uwo.ca) received his B.E. degree in communication and electronics engineering in 2012 from Beirut Arab University, Lebanon. In 2015, he received his M.E.Sc. in computer and software engineering from Western University, Canada. He is currently working toward the Ph.D. degree in computer networks and cloud computing virtualization technologies at Western University, Canada. His research interests include cloud computing, virtualization technologies, software defined network, network function virtualization, distributed systems, and highly available software.

Manar Jammal (mjammal@uwo.ca) is a research associate at Western University. She received her B.Sc. in electrical and computer engineering in 2011 from the Lebanese University, Beirut, Lebanon. In 2012, she received her M.E.Sc. in electrical and electronics engineering from the Ecole Doctorale des Sciences et de Technologie, Beirut, Lebanon in cooperation with the University of Technology of Compiegne, France. In 2017, she received her Ph.D. degree in high availability of cloud applications from the Western University, London, Canada. Her research interests include cloud computing, virtualization, high availability, simulators, machine learning, software defined network, and virtual machine migrations. She is the Past-Chair of IEEE Women In Engineering, London, ON and Chair of IEEE Canada Women In Engineering.

Abdallah Shami received the B.E. degree in electrical and computer engineering from the 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, New York, NY in September 2002. In September 2002, he joined the Department of Electrical Engineering at Lakehead University, Thunder Bay, ON, Canada as an assistant professor. Since July 2004, he has been with Western University, Canada where he is currently a professor in the Department of Electrical and Computer Engineering. His current research interests are in the area of network optimization, cloud computing, and wireless networks. He is an editor of *IEEE Communications Surveys and Tutorials* and has served on the editorial board of *IEEE Communications Letters* (2008–2013). He has chaired key symposia for IEEE GLOBECOM, IEEE ICC, IEEE ICNC, and ICCIT. He is an IEEE Distinguished Lecturer and Senior Member of IEEE and was the elected chair of the IEEE London Section and chair of IEEE Communications Society Technical Committee on Communications Software.