FENDE: Marketplace-Based Distribution, Execution, and Life Cycle Management of VNFs

Lucas Bondan, Muriel F. Franco, Leonardo Marcuzzo, Giovanni Venancio, Ricardo L. Santos, Ricardo J. Pfitscher, Eder J. Scheid, Burkhard Stiller, Filip De Turck, Elias P. Duarte Jr., Alberto E. Schaeffer-Filho, Carlos R. P. dos Santos, and Lisandro Z. Granville

Abstract

The emergence of NFV has drawn the attention of academia, standardization bodies, and industry, because of the possibility of reducing capital and operational costs while introducing innovation in computer networks. To enable developers to independently publish and distribute VNFs, marketplaces akin to online application stores are essential. Research efforts in several aspects are necessary to enable wider adoption of such online application stores in emerging NFV-based computer networks. This article reviews the historical perspective of networking paradigms and technologies to propose FENDE, a marketplace and ecosystem for the distribution and execution of VNFs and composition of service function chains. Major challenges that must be overcome to promote the adoption of marketplaces in emerging NFV-based networks are investigated and discussed.

Introduction

Computer networking technologies have evolved in many different ways to support the development and adoption of innovative services. Programmable virtual networking (PVN), software-defined networking (SDN), and network functions virtualization (NFV) are examples of disruptive concepts that have been exploited to offer advanced networking environments which foster innovation. Recently, special attention has been given to NFV and its capability to develop, deploy, manage, and integrate virtualized network functions (VNFs). NFV has begun to be widely adopted by both industry and academia, thus becoming fundamental for providing flexible network services.

As NFV adoption grows, the number of available VNFs also increases, leading to the need for proper solutions to offer and distribute these functions to network operators. We advocate that NFV can benefit from a software offering and distribution model, which has proven to be effective for other technologies. More specifically, following the trend initiated by the Google Play Store and Apple App Store, which popularized the business model where third-party developers are able to offer applications to users of mobile devices, marketplace solutions for NFV have been proposed [1, 2]. However, the available NFV marketplaces are designed for specific scenarios and to fulfill specific demands without considering its adoption in different network scenarios, such as multi-vendor VNF acquisition and service function chaining (SFC) composition. Moreover, such solutions usually provide VNFs' source code for download but do not offer adequate management tools nor the NFV infrastructure (NFVI) to execute VNFs. We argue, on the other hand, that the design of NFV marketplaces should consider three fundamental aspects: VNF offering, life cycle management, and infrastructure management.

Based on fundamental aspects of NFV marketplaces, in this article we propose FENDE (https://gt-fende.inf.ufrgs.br), a marketplace and federated ecosystem for the distribution and execution of VNFs [3]. In FENDE, developers are able to offer their VNF solutions, while customers can acquire them and choose whether to use public or private infrastructures to instantiate the acquired VNFs. In addition, FENDE provides infrastructure support for VNF instantiation, so customers can acquire and execute VNFs through a unified interface. FENDE also delivers all VNF life cycle management operations and SFC capabilities, in which customers can compose chains with the acquired VNFs to deliver network services. FENDE is the first NFV ecosystem that provides a marketplace for VNF offering together with VNF and SFC creation and life cycle management, as well as the infrastructure support needed for VNF and SFC instantiation.

Marketplaces: Historical Perspective

Online marketplaces have evolved alongside the emergence of new technologies. The popularization of smartphones, for example, created a market for apps, which led the main mobile vendors to deploy their marketplaces as a way to offer applications to end users. A historical perspective of online marketplaces is depicted in Fig. 1. Each paradigm corresponds to a horizontal line parallel to the main timeline, on which technologies and marketplaces are plotted. These marketplaces are those that have/had market dominance,

The authors review the historical perspective of networking paradigms and technologies to propose FENDE, a marketplace and ecosystem for the distribution and execution of VNFs and composition of service function chains. Major challenges that must be overcome to promote the adoption of marketplaces in emerging NFV-based networks are investigated and discussed.

Lucas Bondan, Ricardo L. Santos, Ricardo J. Pfitscher, Alberto E. Schaeffer-Filho, and Lisandro Z. Granville are with UFRGS; Leonardo Marcuzzo and Carlos R. P. dos Santos are with UFSM; Giovanni Venancio and Elias P. Duarte Jr. are with UFPR; Muriel F. Franco, Eder J. Scheid, and Burkhard Stiller are with UZH; Filip De Turck is with UGent-imec.

Digital Object Identifier: 10.1109/MCOM.2018.1800507

Online marketplaces have evolved alongside the emergence of new technologies. The popularization of smartphones, for example, created a market for apps, which led the main mobile vendors to deploy their marketplaces as a way to offer applications to end-users.

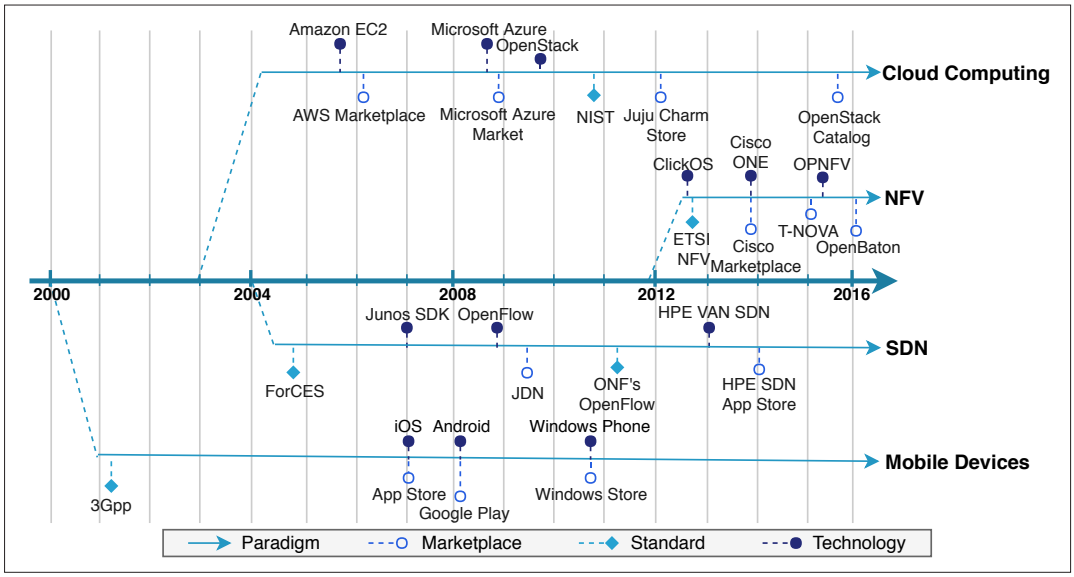

Figure 1. Technologies and marketplace timeline.

regarding the amount of customers and services available, in every technology and paradigm that we considered.

The Third Generation Partnership Project (3GPP) released the 3G Networks specifications in the early 2000s, representing a milestone in mobile networks. Later, in the mid-2000s, the popularization of smartphones boosted the mobile market. Although several mobile platforms were developed (e.g., BlackBerry, Symbian, and Windows Mobile), Apple and Google opened up the market of apps by introducing, in 2007 and 2008, respectively, whole ecosystems composed of operating systems (i.e., iOS and Android) and marketplaces (e.g., Apple's App Store and Google Play) to provide apps to smartphone users. In 2010, Microsoft also adopted this strategy by making available Windows Phone and Windows Store. As a result, the combination of mobile platforms and marketplaces allowed third-party developers to offer a broad range of apps. As an illustration, in 2017, Google Play offered around 3.1 million distinct apps, and its revenue is forecasted to add nearly US$10 billion to the world economy [4].

Although cloud computing emerged at the beginning of the 2000s, its consolidation occurred from 2006 onward, when providers started to offer on-demand services over the cloud. In 2006, Amazon announced an on-demand computing platform called Elastic Compute Cloud (EC2), and in 2008 Microsoft entered into this market by launching the Microsoft Azure cloud computing platform. Later, in 2010, OpenStack was also introduced as an open source cloud enabler. Because of the wide adoption of this paradigm, in 2011, the National Institute of Standards and Technology published the NIST definition of cloud computing [5]. Meanwhile, several companies invested in marketplaces to offer an easy way to distribute applications and services over the cloud. Among them, four marketplaces are worth highlighting: Amazon Web Service (AWS) Marketplace, Microsoft Azure Market, Juju Charm Store, and OpenStack Catalog. AWS Marketplace contains a collection of cloud computing services for EC2. Azure Market provides a collection of integrated cloud services with solutions for data storage, database management, mobile services, and networking. Juju Charm is a project under the auspices of Canonical, which consists of a marketplace to enable applications and services modeling for clouds. Finally, OpenStack Catalog hosts ready-to-use applications that customers can deploy within OpenStack clouds.

In computer networks, SDN is a paradigm characterized by decoupling the network control (control plane) from the forwarding functions (data plane) [6]. One of the first relevant attempts to standardize SDN was ForCES, which defined an architectural framework and associated protocols for the communication between control and forwarding elements. In 2007, Juniper released the Junos service development kit (SDK) to allow the development of applications in the Junos OS, and since 2009 Juniper maintains the Juniper Developer Network (JDN) and Juniper Professional Services Marketplace (JPSM) to foster a community of network application developers. At the end of 2008, OpenFlow established itself as the most important SDN implementation. Shortly after the first specification of OpenFlow, the Open Network Foundation (ONF) became responsible for the standardization efforts. In turn, Hewlett-Packard Enterprise (HPE) released the HPE VAN SDN controller at the beginning of 2013. Right after, in 2014, HPE introduced a marketplace for SDN applications (HPE SDN App Store), allowing network end users to deploy services by aligning the network with business needs.

With respect to the virtualization of network functions, between 2012–2013, the European Telecommunications Standards Institute (ETSI) published the NFV architectural framework [7], running in virtual machines (VMs) hosted on commercial off-the-shelf servers. From 2012 onward, several solutions were proposed to accelerate the adoption of NFV, such as open platforms for NFV (e.g., ClickOS in 2012 and OPNFV in 2014) and frameworks that simplify the development

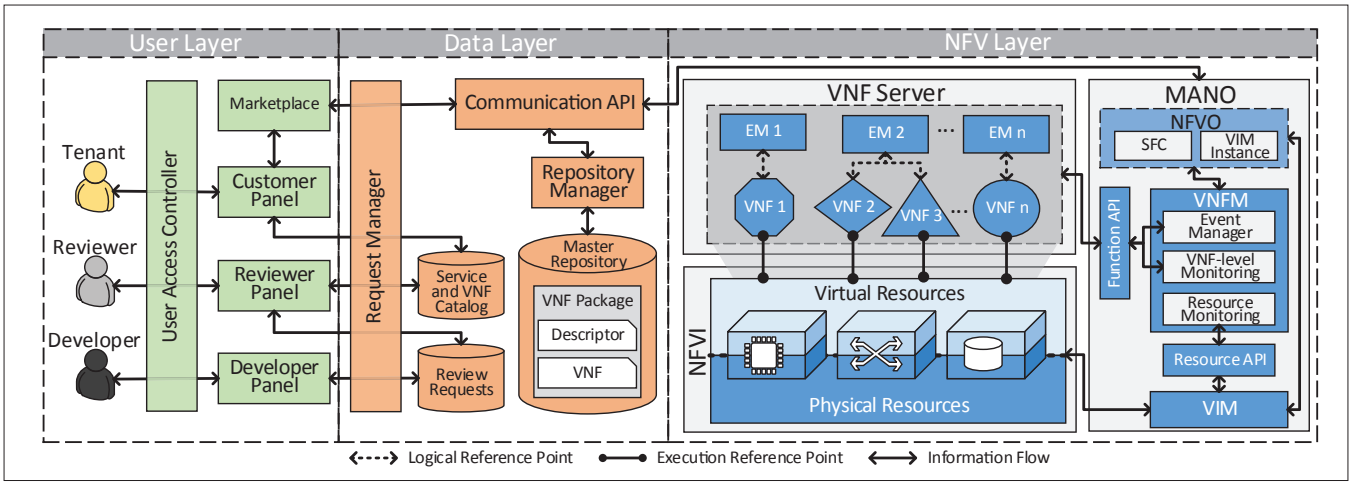

Figure 2. FENDE architecture.

of VNFs (e.g., Cisco Open Network Environment [ONE] in 2013). Also, important players have spent efforts to facilitate the distribution of VNFs: Cisco Marketplace (released in 2014) offers applications and hardware solutions from Cisco itself as well as from partner companies, whereas the T-NOVA project (started in 2015) proposes a marketplace that enables network end users to purchase and deploy VNFs according to their demands. In 2016, the Open Baton project made available a marketplace for downloading VNFs compatible with the Open Baton NFV Orchestrator and VNF Managers.

By analyzing both NFV characteristics and general marketplace features, we identified four main requirements for the development of NFV marketplaces: offering, execution, accounting, and management. Today, NFV marketplaces partially cover these requirements. Cisco Marketplace, T-NOVA, and Open Baton offer VNFs and services as well as tools for their configuration. However, only the first offers physical resources for their execution. Cisco and T-NOVA provide a business model where network end users can purchase and deploy VNFs according to their demands. Such a business model, however, does not consider third-party developers offering their solutions in the marketplace. Moreover, there is no possibility of integration with institutions belonging to federated infrastructures, widely disseminated infrastructures/testbeds, or even the usage of private infrastructures for VNF execution. We capitalize on these previous efforts to investigate and present all functionalities needed in a full marketplace ecosystem for VNF offering, execution, accounting, and management.

FENDE: Marketplace and Federated Ecosystem for VNFs

FENDE was designed considering three users: developers, reviewers, and customers. Each interacts with FENDE through dedicated access and management panels that provide all operations needed for marketplace operations. Also, FENDE provides infrastructure support to execute and monitor virtualized functions. Thus, FENDE places itself as the first NFV marketplace solution that provides all functionalities needed for customers to acquire and execute VNFs and SFCs, thus combining marketplace, management, and infrastructure capabilities in one solution.

FENDE Architecture

FENDE is based on the NFV architectural framework defined by ETSI, as illustrated in Fig. 2. The FENDE architecture is divided into three layers, each layer with specific modules for different operational levels.

User Layer: The user layer contains the elements responsible for the interaction between different users with the platform. Developers fill a registration form for VNFs they want to offer, containing information regarding VNF characteristics (e.g., source code and virtualization requirements). Then reviewers analyze registration requests sent by developers before VNFs become available in the marketplace. Once approved, customers can access the marketplace and select the VNFs they want. Acquired VNFs are available in the customer's library, where instances of each VNF can be created. In addition, customers are also able to perform life cycle management operations as well as create SFCs with VNFs acquired.

Data Layer: The data layer handles all information regarding VNFs, SFCs, and FENDE's users. As such, three main databases are designed: Review Requests, containing all developers' requests for VNF registration in the marketplace; Service and VNF Catalog, containing all VNFs and SFCs available for acquisition by customers in the marketplace; and Master Repository, where all VNF descriptors and information on running instances are stored. Once VNFs are accepted and/or instantiated, a series of events must occur in the platform so that other modules can use the information synchronously. To do so, three modules in this layer integrate the user layer with the NFV layer.

Request Manager: Controls the repository of submissions in the Review Requests database and also performs the migration to the catalog when a VNF is accepted.

Communication API: Provides communication between the user and NFV layers. Its main functions are: 1) requesting the creation or update of VNFs' repositories and 2) requesting VNFs' descriptors for instantiation. All modules that need

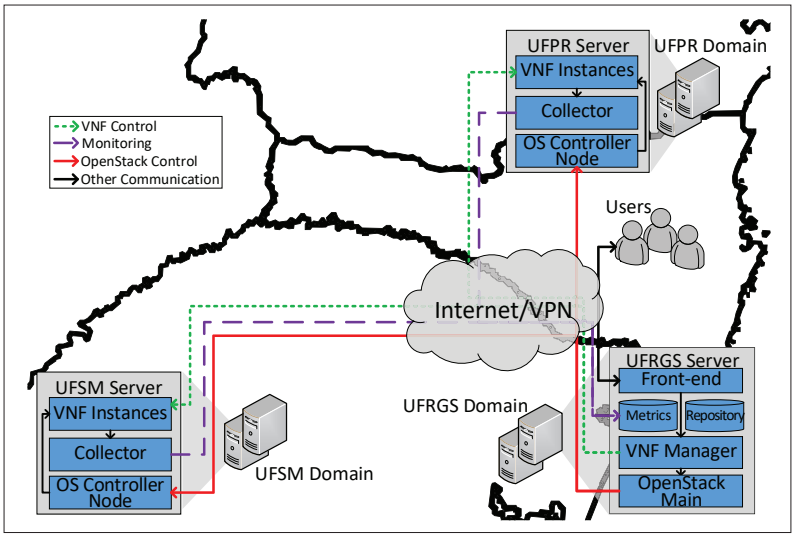

Figure 3. FENDE platform deployment scenario.

information belonging to the Repository Manager can forward the request through the Communication application programming interface (API).

Repository Manager: Creates, maintains, and manages VNFs' descriptors available in the Catalog. For example, when a repository is accepted, the Repository Manager clones and maintains a local version of that repository.

NFV Layer: This layer brings together the main NFV elements proposed by ETSI, divided into three sublayers.

NFV Management and Orchestration (MANO): Designed to handle operations related to services' and functions' life cycle management, as well as resource sharing among virtual elements. It has three main components.

VNF Manager (VNFM): Responsible for VNF life cycle management operations, such as instantiating, removing and updating VNFs, as well as creating SFCs. To enable both hardware and software-level VNF management, FENDE has three main modules:

- Events Manager, responsible for receiving requests from the user layer and performing VNF life cycle management, activating the two other modules accordingly
- Resource Monitoring, which monitors metrics related to physical resources assigned for each VNF such as CPU, memory, and storage
- VNF-Level Monitoring, which monitors the function of each VNF, collecting metrics related to the function usage, such as number of processed packets and operations latency

Virtualized Infrastructure Manager (VIM): Controls all resources available in the NFVI. FENDE supports different VIMs, using its communication API to abstract technology-specific commands. Thus, FENDE supports the composition of heterogeneous infrastructures, such as local interconnected infrastructures based on the CloudStack and OpenStack platforms;

NFV Orchestrator: Brings intelligence to service provisioning and composition processes, directly interacting with VNFMs for managing VNF operation life cycle. Likewise, NFVI virtual and physical resource sharing orchestration among different virtualized elements is performed by NFVOs through VIMs.

VNF Server: Supports the execution and control of VNFs' operation locally. In virtualized environments, resources in the underlying infrastructure must be abstracted, for example, through hypervisor-based or container-based virtualization. In addition, element managers (EMs) are designed to handle technology-specific information, such as fault, configuration, accounting, performance, and security (FCAPS) parameters. EMs must be co-located with VNFs and retrieve information regarding VNFs' execution, sending such information to the VNFM to control VNFs' operation.

NFV Infrastructure: Corresponds to physical and virtual resources available for VNF deployment, that is, computing, memory, storage, and networking. FENDE supports multiple network domains to compose the NFVI, connecting them through VPNs, so communication among VNFs is possible and instances run on the same subnet, allowing the use of SFCs spanning over multiple domains. Although no elasticity mechanism is currently implemented, FENDE supports the definition of placement optimization mechanisms, which can be used for automated horizontal and vertical scaling [8].

FENDE Prototype

The FENDE prototype provides a web interface to enable management and interaction with different users. Each user interface provides resources that enable a set of operations within the ecosystem. Developers must submit a valid Git repository with the VNF source code to be evaluated by reviewers. The current review process is manual, with reviewers analyzing if submitted VNFs perform as described by developers, and checking for the absence of malicious code. However, autonomic revision mechanisms (e.g., bots as used in Google Play) are an interesting research topic to be further explored. Once approved, the Git repository is cloned to the local marketplace repositories, and customers are then able to acquire, instantiate, and manage VNFs and create their own SFCs. In the NFV layer, OpenStack and Tacker were used for management at the hardware level. For software-level management, Click-On-OSv [9] is used. In addition, a VNFM submodule was developed to consume both APIs and to fully manage VNFs' life cycle.

Federated Testbed

In our testbed, the Brazilian National Research Network (RNP) is the marketplace regulator, while the FENDE project is in charge of maintaining the marketplace, that is, responsible for its operation and for the VNF review process. Developers are from both industry and academia (Brazilian universities), and can register their own VNF solution and acquire VNFs for instantiation. On the NFVI layer, cloud infrastructure providers such as Amazon EC2 and Windows Azure could be registered along with the infrastructure provided by FENDE.

FENDE was deployed in three network domains across two different Brazilian states, as depicted in Fig. 3: the UFRGS and UFSM domains in the state of Rio Grande do Sul, and the UFPR

domain in the state of Paraná. The marketplace and VNF management front-end are instantiated at the UFRGS domain, while the other domains are used as NFVI, hosting VNFs, and SFCs. The prototype supports operations performed by each actor, such as VNF submission, VNF review process, and all operations regarding VNF life cycle management by customers, including statistics retrieval from VNF operation.

Experimental Results

Since VNFs can be instantiated, configured, and dynamically scaled, the execution time of these operations is a relevant metric to analyze, as delays in such operations can negatively impact a VNF's execution. This way, the execution time of each VNF management operation and respective sub-operations that constitute them was evaluated, showing the cost of FENDE's VNF management calls. This metric is important to quantify the overhead of FENDE in terms of VNF management. These results are shown in Fig. 4.

The *Create* operation is composed of four sub-operations, with the *polling* sub-operation consuming around 90 percent of the total execution time. This occurs because, when sending a request to VM creation (sub-operation *vm_create*), the *polling* sub-operation should periodically check and wait for the infrastructure to finish the instantiation process. Then the function can be configured in the next sub-operation (*vnf_init*). The *Update Function* operation needs to upload a new network function (VNF software) and wait for the VNF to restart. The *Update* operation, after updating the VM description, waits for it to restart, while the *Delete* operation sends commands for the VM to be removed.

Research Challenges and Directions

In the context of NFV, network marketplaces must deal with specific challenges. While typical marketplaces (e.g., Google Play and Apple App Store) are concerned mostly with publishing and deployment, NFV marketplaces need to address aspects such as placement, auditing, and VNF life cycle management. Although FENDE is concerned to some extent with these aspects, it can benefit from improvements in several areas of computer science that can be applied to NFV marketplaces. Based on our experience in developing the FENDE ecosystem in a national backbone network, we identified fundamental challenges, which are detailed next.

Business Model

Business models are critical for the wide adoption of network marketplaces. Nowadays, two major methods are used for application acquisition: *fixed-price* and *pay-as-you-go*. In the former, customers must pay a predefined price for each VNF and can use them without restrictions (e.g., time or size). Google Play, JDN, and Cisco Market are examples of marketplaces that use this method of offering/acquisition. The latter considers different aspects to define the value to be paid (pre/post) to use the application. For example, a developer can offer a multicast-based application and charge the customers based on the number of concurrent flows. AWS and Microsoft Azure, for example, employ this method for specific cases.

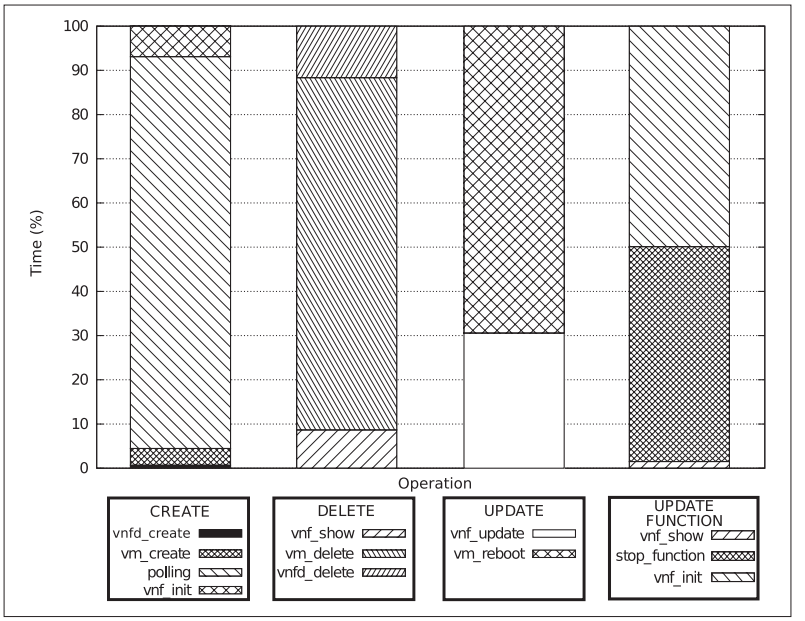

Figure 4. FENDE platform deployment scenario.

In our view, future NVF marketplaces can support the previous cited business models and other innovative methods according to business requirements. We can cite two other interesting methods for network service acquisition: *auction-based* and *custom-built*. An auction-based method can benefit both third-party developers and customers because of its capacity to enable the competition to provide the best product regarding cost and performance. In the custom-built method, there is a negotiation between third-party developers and customers to develop a VNF for specific needs. This negotiation considers the final price, requirements, service level agreements (SLAs), deadlines, and desired features of the VNF to be developed.

These business models have advantages and disadvantages. Fixed-price is the simplest, but does not support any additional customization. Pay-as-you-go and auction-based models are able to adapt to customers' demands, but may be complex to deploy (e.g., need to define trustful monitoring for accounting and a reliable auction system). Finally, the custom-built method provides freedom for the customers to order customized VNFs, but implies challenges to guarantee that customers will describe requirements correctly.

Auditing

Network end users should be able to verify if the deployed VNFs are providing the advertised functionalities. Therefore, network marketplaces must apply auditing mechanisms to gather information about the execution of VNFs. For example, an end user may deploy a network service for distributed denial of service (DDoS) prevention. Upon request, the marketplace must provide reports to the end user showing that the contracted VNF is preventing DDoS attacks according to the previously defined SLAs.

Auditing reports must consider not only if a VNF meets the established SLA, but also how it affects the environment in which it runs. Thus, research efforts should be devoted to designing mechanisms that combine monitoring information (e.g., traffic pattern and resource usage) and

We expect that VNFs will be developed and published by distinct third-party developers and that different environments will deploy these VNFs. For these reasons, the marketplace must employ security mechanisms to prevent the environment from becoming a target of malicious attacks.

diagnosis models (e.g., classification and machine learning approaches) to generate comprehensive reports. Although a comprehensive auditing approach is an open research challenge, there are current efforts that deal with specific parts of it. In the NFV context, Bless and Flittner [10] propose an auditing mechanism to allow customers to verify if the resources allocated by service providers are in accordance with the established SLAs.

VNF Recommendation

As the NFV market grows, the number of VNFs developed is also expected to increase proportionally. Reports indicate that by 2024 the NFV market will be valued at US$70 billion [11]. Although we cannot precisely estimate how many VNFs will be available, we can consider the number of middleboxes present in current network infrastructures as a baseline [12]. In such a direction, an open research challenge is to provide means to distinguish (or compose) the available VNFs to meet specific requirements. For example, security-related VNFs can offer security capabilities at distinct levels, such as inspection firewalls for L3 packets and intrusion prevention systems that detect malicious traffic patterns. The challenge is how to define which VNFs must be selected by network end users to meet their target requirements.

Clustering techniques can be applied to address the recommendation of applications and products. Similarly, VNFs could be grouped into clusters in a multi-dimensional plane considering distinct levels of, for example, security and performance requirements. This could help identify VNFs that provide a high level of security but a low level of performance; and VNFs that provide a low level of security but a high level of performance. However, a reliable recommendation mechanism for VNFs must address several challenges regarding: classification mechanism, number of VNFs in each cluster, order of VNFs through which the flows will pass, classification accuracy, and affinity and anti-affinity relations among VNFs.

In our ongoing research, we are currently addressing these aforementioned recommendation challenges. In a recent study [13], we proposed an intent refinement process that clusters VNFs according to user-defined contexts. In another study [14], we introduced a mechanism to compute the affinity score for each pair of VNFs in a service function chain. During the development of these works, we identified challenges to the recommendation of VNFs due to the limited knowledge about the behavior of VNFs and customers. Besides, the current solutions are not able to deal with all issues related to validation, verification, and performance analysis of the recommended VNFs. Thus, research efforts are still necessary to fully integrate these aspects into network marketplaces.

Placement

Finding the best placement of VNFs over the substrate infrastructure is difficult because each network end user may have different priorities and goals. Also, some VNFs may require a specific location for execution. For example, while firewalls are better placed in the network edge (i.e., close to the external link), an IP media transcoder should stay close to content servers. We observe that several efforts are focusing on optimizing the placement task in NFV-enabled networks. Placement mechanisms must take into account predefined criteria, and provide automated and manual mechanisms to define optimal locations for VNFs. The placement criteria can include minimal network delay, energy saving, deployment cost, and resource utilization.

The placement problem in software-based networks has been widely investigated for years. For instance, Moens and Turck [8] aimed to optimally place VNFs and network services according to established policies in the context of NFV. However, network marketplaces for NFV must also enable the easy and flexible placement of VNFs regarding both distinct technologies and concurrent or conflicting placement criteria. Further, marketplaces must be able to deal with custom infrastructures provided by end users.

Security

We expect that VNFs will be developed and published by distinct third-party developers and that different environments will deploy these VNFs. For these reasons, the marketplace must employ security mechanisms to prevent the environment from becoming a target of malicious attacks. For instance, if a network end user acquires a VNF for energy saving in his/her network, the marketplace needs to ensure the integrity of the VNF, and also provide a secure communication channel to deploy and send management commands to the VNF. This would prevent malicious users from interfering with the communication (e.g., man in the middle attacks to steal sensitive data) or sending commands to perform undesired actions (e.g., installing malicious software or stopping services).

Malicious users could also develop VNFs to be the source of attacks against third-party environments. In view of this, marketplaces should employ tools to guarantee the integrity of VNFs and SFCs [15]. Much can be learned from the two most successful mobile marketplaces: Google's Play Store and Apple's App Store. Apple developers need to go through a rigorous enrollment process and adhere to a stringent review process in order to publish their apps. Despite being less restrictive with submissions, Google imposes security mechanisms for submitted apps, automatically scanning them for potentially malicious code before acceptance. This way, strict contracts and autonomic VNF check mechanisms would play an important role to guarantee both marketplace and customer safety.

Conclusion

As NFV becomes more popular, a sharp increase in the number of VNFs available in the market is expected. As such, NFV marketplaces can provide the environment where VNF developers and customers can negotiate solutions. In this article we introduce FENDE, an NFV architecture for marketplace-based distribution and execution of VNFs. FENDE provides a VNF marketplace together with all life cycle management functionalities needed to instantiate and control VNFs' operation, as well as the composition of SFCs. In addition, FENDE provides the infrastructure for VNF and SFC instantiation, placing itself as the first end-to-end NFV marketplace ecosystem.

Research challenges regarding the adoption of NFV marketplaces are also investigated. We find that issues regarding auditing, recommendation, and placement still require considerable research efforts. Also, significant research efforts are needed to integrate the distinct technologies to provide flexible and useful NFV ecosystems. As future research, security implications of the VNFs published in marketplaces must be investigated. For example, mechanisms to keep the integrity of NFV elements throughout their lifetimes are needed to avoid security breaches or service unavailability. Moreover, auditing mechanisms can help point out responsibilities when something goes wrong in any part of the system.

References

[1] G. Xilouris *et al.*, "T-NOVA: A Marketplace for Virtualized Network Functions," *Proc. Euro. Conf. Networks and Commun.*, 2014, pp. 1–5.

[2] OpenBaton; https://openbaton.github.io/; accessed 15 June, 2018.

[3] L. Bondan *et al.*, "FENDE: Marketplace and Federated Ecosystem for the Distribution and Execution of VNFs," *Proc. ACM SIGCOMM — Posters and Demos*, 2018, pp. 135–37.

[4] S. T. S. Portal, "App Stores — Statistics & Facts," 2017; https://www.statista.com/topics/1729/app-stores/, accessed 15 June, 2018.

[5] P. Mell and T. Grance, "The NIST Definition of Cloud Computing," White Paper, NIST, 2011; https://csrc.nist.gov/publications/detail/sp/800-145/final, accessed 15 June, 2018.

[6] J. A. Wickboldt *et al.*, "Software-Defined Networking: Management Requirements and Challenges," *IEEE Commun. Mag.*, vol. 53, no. 1, Jan. 2015, pp. 278–85.

[7] M. Chiosi *et al.*, "Network Functions Virtualisation (NFV)," ETSI NFV ISG, White Paper 1, 2012; https://portal.etsi.org/NFV/NFV White Paper.pdf, accessed 15 June, 2018.

[8] H. Moens and F. De Turck, "VNF-P : A Model for Efficient Placement of Virtualized Network Functions," *Proc. Int'l. Conf. Network and Service Management*, Nov 2014, pp. 418–23.

[9] L. da Cruz Marcuzzo *et al.*, "Click-on-OSv: A Platform for Running Click-Based Middleboxes," *Proc. IFIP/IEEE Symp. Integrated Network and Service Management*, May 2017, pp. 885–86.

[10] R. Bless and M. Flittner, "Towards Corporate Confidentiality Preserving Auditing Mechanisms for Clouds," *Proc. IEEE Int'l. Conf. Cloud Networking*, Oct. 2014, pp. 381–87.

[11] "Network Function Virtualization (NFV) Market to See 42% Growth to 2024"; https://markets.businessinsider.com/news/stocks/networkfunction-virtualization-nfv-market-to-see-42-growth-to-2024- 1027473777, accessed 15 Nov. 2018.

[12] J. Sherry *et al.*, "Making Middleboxes Someone Else's Problem: Network Processing as a Cloud Service," *Proc. ACM SIGCOMM Conf. Applications, Technologies, Architectures, and Protocols for Computer Commun.*, 2012, pp. 13–24.

[13] E. J. Scheid *et al.*, "INSpIRE: Integrated NFV-Based Intent Refinement Environment," *Proc. IFIP/IEEE Symp. Integrated Network and Service Management*, May 2017, pp. 186–94.

[14] A. S. Jacobs *et al.*, "Affinity Measurement for NFV-Enabled Networks: A Criteria-Based Approach," *Proc. IFIP/IEEE Symp. Integrated Network and Service Management*, May 2017, pp. 125–33.

[15] L. Bondan *et al.*, "Anomaly Detection Framework for SFC Integrity in NFV Environments," *Proc. IEEE Conf. Network Softwarization*, July 2017, pp. 1–5.

Biographies

Lucas Bondan is a Ph.D. student at UFRGS, Brazil. His research interests include NFV, network management and orchestration, and SFC.

Muriel F. Franco is a Ph.D. student at UZH, Switzerland. His research interests include NFV, network management, information visualization, and blockchain.

Leonardo Marcuzzo is an M.Sc. student at UFSM, Brazil. His research interests include NFV and operating systems.

Giovanni Venancio is a Ph.D. student at UFPR, Brazil. His research interests include NFV and fault-tolerant distributed systems.

Ricardo L. dos Santos is a Ph.D. student at UFRGS, Brazil. His research interests include network programmability and network virtualization.

Ricardo J. Pfitscher is a Ph.D. student at UFRGS, Brazil. His research interests include network virtualization, VNF monitoring, and network management.

Eder J. Scheid is a Ph.D. student at UZH, Switzerland. His research interests include NFV, PBNM, and blockchain.

Burkhard Stiller is a full professor at UZH, Switzerland. His research interests include Internet services, decentralized systems, and network and service management.

Filip De Turck is a full professor at the University of Gent, Belgium. His research interests include network and service management, IoT, and multimedia delivery systems.

Elias P. Duarte, Jr. is a full professor at UFPR, Brazil. His research interests include network management, distributed systems, and algorithms.

Alberto E. Schaeffer-Filho is an associate professor at UFRGS, Brazil. His areas of expertise are network/service management, network resilience, and programmable networks.

Carlos R. P. dos Santos is an adjunct professor at UFSM, Brazil. His research interests include network virtualization, network programmability, and QoS management.

Lisandro Z. Granville is an associate professor at UFRGS, Brazil. His research interests include network management, Intent-based networking, and network programmability.

We find that issues regarding auditing, recommendation, and placement still require considerable research efforts. Also, significant research efforts are needed to integrate the distinct technologies to provide flexible and useful NFV ecosystems.