2016 IEEE 30th International Conference on Advanced Information Networking and Applications

VISION - Interactive and Selective Visualization for Management of NFV-Enabled Networks

Muriel Figueredo Franco, Ricardo Luis dos Santos, Alberto Schaeffer-Filho, Lisandro Zambenedetti Granville
Institute of Informatics – Federal University of Rio Grande do Sul
Porto Alegre, Brazil
{mffranco, rlsantos, alberto, granville}@inf.ufrgs.br

***Abstract*—Network Functions Virtualization (NFV) enhances the flexibility of network service provisioning and reduces the time to services deployment. NFV and SDN promises transform the carrier networks, introducing innovation in the network core. NFV moves packet processing from dedicated hardware middleboxes to Virtualized Network Functions (VNFs), which run on virtual machines hosted on commercial off-the-shelf servers. However, in NFV-enabled networks, the amount of data managed grows in a fast way. Based on this, the network operator must understand and manipulate a lot of information to effectively manage the network. In this paper, we introduce the VISION, a platform to help the network operator to determine the cause of problems based on visualizations techniques. Our platform implements a set of interactive and selective visualizations to assist in the NFV management. Finally, we conducted three cases studies to provide evidences of the feasibility of our platform.**

I. INTRODUCTION

Network Functions Virtualization (NFV) enhances the flexibility of network service provisioning and reduces the time to services deployment [1]. NFV moves packet processing from dedicated hardware middleboxes to network functions, hereinafter referred to as Virtualized Network Functions (VNFs), running on virtual machines hosted on commercial off-the-shelf (COTS) servers. The benefits offered by NFV allows network operators to create innovative services, reducing the capital expenditure (CAPEX) and operational expenditure (OPEX) [2]. Also, by using NFV, the operator can easily scale network services according to the multiple client-organizations (tenants) demands, offering technical feasibility and potential business opportunities while increases the network innovation.

In NFV-enabled networks, one tenant purchases VNFs that implement different services, such as a firewall and load balancer. The operator must chain these VNFs (*i.e.*, service chaining) in order to comply with the tenant's demands. The European Telecommunications Standards Institute (ETSI) defines such service chaining as a forwarding graph of VNFs interconnected by the virtual network infrastructure [3]. This forwarding graph simplifies the representation of the service chaining, allowing the operator realize daily tasks quickly, such as creating and modifying packets flows along VNFs, deployment of new VNFs, and maintenance tasks.

Although the aforementioned service chaining enables the quick deployment and maintenance of a vast number of VNFs, the operator needs to have extensive knowledge of the enormous amount of data available to efficiently manage the NFV environment. In fact, understanding these data is a challenging task itself because of the previously mentioned amount of data involved. Today, there are several visualization solutions to help in the network management [4] and ease the process of a human have insights into a large dataset, such as identifying policy violations [5] and analysis of security issues in large-scale computer networks [6]. However, none visualization solutions consider the NFV features (*e.g.*, forwarding graph and dynamic migration). As a consequence, solutions based on visualization techniques not support directly the daily tasks related to NFV management (*e.g.*, identifying misplacement and misconfigurations in the VNFs).

Visualization solutions provide several benefits for the network operator tackling problems that arise while VNFs are running. Such benefits include a quickly understanding of the information about the environment [7], identification of VNFs behavior patterns [8], and easy maintenance of VNFs. Today, none of the traditional tools to NFV management explore visualization techniques to help in the process of having insights into the cause of VNFs problems. Thus, we advocate that visualization solutions are powerful tools to assist network operator into problem identification that are not obvious in NFV-enabled networks. For instance, a network operator can access the visualization of a history of VNFs migration to identify recurrent problems and have insights about what rules must be created to avoid problems, such as affinities among VNFs and hardwares.

In this paper, we present VISION (**Visu**a**l**izat**i**on **O**n **N**FV), a platform that implements a set of interactive and selective visualization techniques. Such visualizations assist network operators to identify and mitigate the cause of not obvious VNF problems, such as misconfiguration or misplacement. Besides, our platform: *i*) helps in the identification of problems that impact negatively on the VNF performance, *ii*) provides distinct perspectives about the NFV-enabled networks that allow planning and offering a better service for tenants; and *iii*) provides visualizations that allow looking for behavior standards, which determine affinities issues among VNFs. Finally, we conducted three case studies in a Virtualized Network Function as a Service (VNFaaS) scenario to demonstrate evidence of the feasibility of our platform.

The remaining of this paper is organized as follows. In Section II, we present the related work. In Section III, we introduce our platform and describe each of components and modules. In Section IV, we present three case studies to prove the concept of the platform proposed and realize discussion. Finally, in Section V, we conclude the paper and list future work.

1550-445X/16 $31.00 © 2016 IEEE
DOI 10.1109/AINA.2016.26

II. Related Work

Both information visualization and network management areas have received considerable attention from the scientific community. Several visualization solutions support the discovery and analysis of information through visual exploration [4]. These solutions enable human operators to explore and have insights about not obvious network problems, such as conflicting firewall rules [9], network device failures [10], and malicious attacks [11]. Hence, operators can recognize patterns, anomalies, and incompatibility issues in a fast and efficient way. Below, we discuss some of the most important studies in the information visualization area mainly related to the network management.

Liao and Striegel [12] proposed a visual analytic tool based on graph differential anomaly visualization model to find the root causes of anomalies in dynamic graphs. The tool combines both human and computer intelligence for a smarter network operation and management. Several agents collect local context information (*e.g.*, running applications and connections). Such information fed a tool that applies the proposed model and provides different graphs to the network operator. Despite identifying behavior changes of users and applications, which can occur by malicious intention or misconfiguration, this tool neither provides information about conflicting applications and nor considers data from network services.

In another study, Isolani *et. al.* [13] perform an analysis of the control traffic in OpenFlow networks to verify the impact of specific parameters and their influence on the overall resource consumption and network performance. Based on this analysis, the authors proposed a tool that integrates Software-Defined Networking (SDN) monitoring, visualization, and configuration activities. Such tool allows operators to configure and control SDN-related parameters, minimizing the impact, in terms of resource consumption and performance, of the network management. However, this tool only focuses on statistics visualizations and operators customize neither the monitors and nor the generated visualizations.

The Real Status company [14] developed the HyperGlance platform to dynamically aggregates key resource data, relationships, and control functions into a unified, full-scale, interactive visualization for simplified, efficient, cross-platform monitoring and management within a single screen. This platform integrates data from various platforms (e.g., OpenStack, Open Daylight, and Nagios) to generate several visualizations, allowing operators to perform decisions about the whole network. Nevertheless, HyperGlance intends to support NFV environment in the future. Thus, this general solution not covers all issues that concerning the NFV management.

Although the significant attention received by the information visualization and network management areas from companies and academia, none of the solutions explore visualization techniques in NFV-enabled networks. As a consequence, these solutions do not assist human operators to have insights about particular problems, because they ignore important NFV features (*e.g.*, forwarding graph and dynamic migration). To fill this gap, in the next sections, we introduce VISION platform as follows.

III. VISION Platform

In this section, we present VISION (Visualization on NFV), a platform to help the human operator in the problem identification process into NFV-enabled networks. The VISON platform obtains data from the environment and presents it using visualization techniques [15], such as link-node representation, matrix layout, and charts. Our platform provides visual resources that allows network operators to have insights about mitigating the cause of problems in the NFV environment.

VISION considers two main goals to simplify the NFV management and assist in the problem identification process. First, we provide an interactive way to define which data must be visualized and how data are presented. Hence, the network operator can choose the information that will be represented in visualizations. For example, the operator can select statistics to generate a chart that represents only VNFs of the same forwarding graph. Second, our platform collects, organizes, and stores a vast amount of data from distinct sources. VISION communicates with generic NFV platforms (*e.g.*, openMANO) to obtain the available data. Moreover, operators may create and configure their monitors to collect additional statistics in order to enrich VISION database (see Subsection III-B3).

An operator access VISION resources via a Web-based interface that provides details from the collected data, tenants, and configurations. In Figure 1, we show such Web-based interface. This interface contains a dashboard to provide an overview of the NFV environment, such as the amount of VNFs available, resources usage, and migration alerts. Besides, the Web-based interface provides significant information about the network traffic, allowing observe how much traffic is generated by tenants. Also, logs from relevant events (*e.g.*, migrations and failures of VNFs) are available on the dashboard. Such interface was implemented using Django framework 1.8.5 and as database engine PostgreSQL 9.4.5.

In Figure 2, we present VISION conceptual architecture. The main components and modules are, respectively, highlighted with dark gray and white background. As can been seen, a generic NFV environment contains the follows components: *i)* Network Function Virtualization Infrastructure (NFVI), which enables the software to run independently of the hardware, *ii)* Virtualized Network Functions (VNFs) that moves network functions from dedicated hardware devices to software hosted in COTS servers; and *iii)* NFV Management & Orchestration (MANO) [16], which has three components to monitor and configure the NFV environment: Virtualized Infrastructure Manager (VIM), VNF Manager, and Orchestrator.

In MANO, VIM monitors and controls computing, storage, and networking resources of the NFVI domain. VNF Manager performs several VNFs operations (*e.g.*, deployment and migrations). In addition, this manager stores data about configuration and VNFs events (*e.g.,* alarm to SLAs disruptions). Finally, the Orchestrator realizes the global resource management, validation, and authorization of NFVI resource requests. Besides, such component includes performance measurements and policy management for VNFs. These components have a significant amount of meaningful information. For instance, the orchestrator has a textual representation of forwarding graph (service chaining) and perceives when VNFs are migrated. As

HW-X is satured 4 minutes ago
HW-Y is satured 12 minutes ago
An migration occured 27 minutes ago
An migration occured 43 minutes ago
An migration occured 11:32 AM
HW-Y is satured 11:13 AM
Tenant 42 notification 10:57 AM
HW-X is satured 9:49 AM
An migration occured Yesterday

Fig. 1: VISION Web-based interface

such, the MANO framework information is crucial to populate VISION platform database. The Data Collector component collects this information and a set of specific statistics (*e.g.*, CPU usage and memory diagnostics) available in the custom monitors.

VISION architecture (Figure 2) contains three main components, which are in charge of collecting, organizing and providing visualizations for the network operator. First, the Data Collector gathers data from the NFV environment. Second, the Data Processor manipulates and stores such data in a database. Finally, the Visualization Manager builds the visualizations. All of VISION components are described in the rest of this section.

A. Data Collector

The Data Collector component obtains data (*e.g.*, VNFs events and textual forwarding graph representation) from MANO implementation. For this, the Data Collector sends GETs messages via API REST (Representational State Transfer) to MANO. Besides, this component may collect data from customized monitors placed into the VNFs. The network operator must configure, in VISION platform, the monitors parameters (*e.g.,* correspondent VNF, hardware, and IP address from the monitor) to allow the data collection.

Currently, VISION platform is able to acquire data from openMANO framework [17]. We choose this framework because it is the first open source project that provides a practical implementation of the NFV Orchestration and Management. The openMANO offers a northbound API that allows getting information from the NFV environment, such as resource allocation and VNFs events. The process of collecting data is based on REST requests. The response of each request is a JSON with information from the object. For example, the request *GET /openvim/hosts/*$\{$*host_id*$\}$ returns information about each of interfaces attached on the *host_id*.

Data Collector can get data from another source to enrich VISION database. For this, such component requests data from customized monitors strategically placed in the NFV environment. The network operator can create and configure such monitors to obtain additional statistics from VNFs, such as CPU usage and memory diagnosis. Besides, these monitors can collect specific information (*e.g.,* data about Firewall's rules and packet process errors). Our platform requests and receives data from monitors via a socket connection. For instance, the request *GET* $\{$*monitor_IPaddress*$\}$ returns a JSON with information from the specific monitor, *e.g.*, latency and memory usage.

After collecting the data from the MANO and monitors, the component sends such data to the Data Processor component, which separates, organizes, and classifies the information (see Subsection III-B). Then, the Data Processor stores the information in VISION database, thus completing the process of acquiring and manipulating the data.

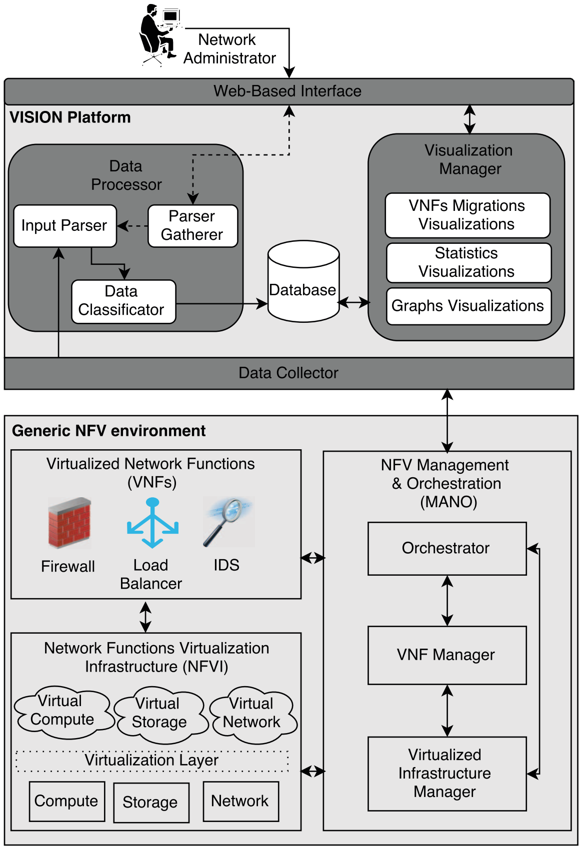


Fig. 2: Conceptual Architecture of VISION

B. *Data Processor*

Data Processor inspects, transforms, and models the data to discover meaningful information. This component manipulates the data to obtain a readable information for others modules. Such manipulations include extracting and organizing the relevant information while removing not usable data. This component is divided in three modules: *i)* Input Parser, *ii)* Data Classificator; and *iii)* Parser Gatherer.

1) Input Parser: The Input Parser module is the first phase of Data Processor. It main function takes the input data and builds a data structure to giving a structural representation of the data, such as organize the data in tables where the columns and lines are representing information from VNFs. Moreover, this module checking for incorrect syntax (*e.g.*, information outside the accepted standards) and identifying failures in the process of acquiring data.

2) Data Classificator: This module tags and classifies the data based on shared characteristics (*e.g.*, create a data structure that represents the VNFs hosted by the same hardware). Such characteristics define groups of VNFs to ease information access. The classification of data allows other VISION components to access quickly and filter the information in order to create customized visualizations, such as shows only statistics from VNFs contracted by a specific tenant. After this step, the data is stored in the database by Data Classificator and available to be requested by VISION platform.

3) Parser Gatherer: This module allows the operator to upload a new parser, in order to assure the manipulation of data from customized sources and formats. Such customized parser must be a Python script, which creates new rules about how the Input Parser will deal with data received from the Data Collector (*e.g.*, defining a new rule to discard the first five columns). This module takes the input data and manipulates according to rules predefined, ensuring that different data will be recognized. Moreover, the operator can configure and use a new parser for each one of the monitors. For example, considering that the Data Collector sends data from two distinct monitors, which contains the information divided, respectively, by commas and semicolon. The Input Parser must separate both data correctly, for this, the operator uses differents implementations of the parser to deal with such data.

C. *Visualization Manager*

The Visualization Manager builds visualizations based on the information available in VISION database. This component explores visualization techniques, such as forwarding graphs and statistics charts visualizations, to provide an interactive and selective view of the NFV environment. Such visualizations help the network operator to identify the cause of problems. In addition, this component receives and processes the parameters to customize the visualization modules, such as which information will be presented and how the visualization will occur.

This component is divided in three modules, which allows the human operator having insights about not obvious problems: *i)* Statistics Visualization, *ii)* VNFs Migrations Visualization; and *iii)* Graphs Visualization. Below, we describe each one of the visualization modules individually.

1) Statistics Visualization: The statistics about VNFs are important to alert the network operator when a problem is occurring. For example, the network operator may analyze events messages that suggest a VNF saturated. With this information, the operator can conduct a more in-depth analysis to have insights about the cause of the problem that impacts on the VNF performance. The Statistics Visualization module provides intuitive visualizations to improve the understanding of VNFs behaviors. The operator can select the available statistics from VNFs (*e.g.*, latency, propagation delay, and CPU usage). Moreover, any specific statistics collected by customized monitors can be presented in a useful visualization. In Figure 3, we show an example of this visualization. In this example, we present a line chart that display a physical host CPU usage along the time. Such visualization helps the operator to understand some behaviors, such as the latency oscillation along a day and what is the VNF behavior when is processing a lot of packets.

An essential feature of Statistics Visualization module is the comparison of distinct statistics from several VNFs (*e.g.*, difference in the latency of VNF-A and VNF-B along the monitoring time). This module provides real-time information for each monitored VNF, such as a number of packets processed and the resources usage. Moreover, the network operator can use the Statistics module to improve others

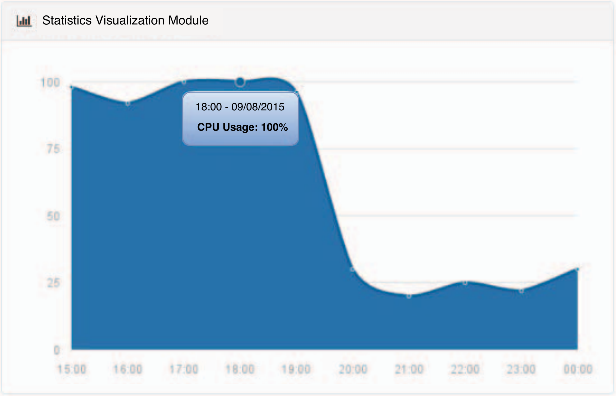


Fig. 3: Example of Statistics Visualization module

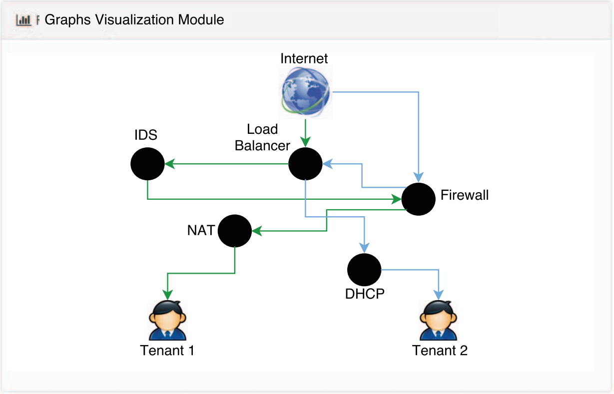


Fig. 4: Example of Graphs Visualization module

visualization modules. For instance, VISION platform can provide a visualization of the VNF migration history, enriching it with novel statistics (see Subsection IV-B). Thus, when the operator clicks in determined VNF, the Statistics module is invoked to provide information about the selected VNF (*e.g.*, build a line chart that shows the CPU usage history).

2) VNFs Migrations Visualization: VNFs can be dynamically migrated, and it brings additional complexity for the network operator know where a given VNF is hosted (*i.e.*, VNF placement). Besides, migrations can have several impacts on the NFV environment, such as affecting VNFs performance when affinities issues are not respected and difficulting for the operator knows where a VNF is hosted. We argue that the analysis of the historical behavior provide a better understanding of the present. Based on this, we store, in the database, information about all of migration events. With such information, we provide a visualization for assisting the operator in identifying the exact time when a VNF presented a misbehavior, textite.g., what VNFs were sharing the same physical resources when a problem occurred.

VNFs Migrations Visualization module provides a full overview of migrations. In this module, the operator selects VNFs to be analyzed, and VISION shows in a clear perspective the hardwares that hosted these VNFs. Also, this module interacts with the Statistics Visualization module to enrich the visualization with charts and statistics about each VNF (*e.g.*, display in a chart the latency of VNFs along the time). Moreover, the network operator can customize these visualizations to obtain useful information, such as highlight VNFs and compare statistics among them. Thus, several issues can be explored with this visualization. For example, if two or more VNFs are not running fine when sharing the same physical hardware, the operator identifies it and may create rules for anti-affinity among these VNFs, preventing that unwanted behaviors arise in the network.

3) Graphs Visualization: Graph theory studies structures to model the relationship among nodes [18]. Based on this, we implemented a module to provide a way for network operators explore the relations between VNFs and tenants. Such module helps humans operators to identify some problems. For example, the network operator can understand the demands of the NFV-enabled network and plans how the VNF instances will be shared among tenants. Besides, this module provides a meaningful view about the forwarding graphs, facilitating the problem identification, such as when a VNF placement incorrect impacts negatively on the performance of another VNF (see Subsection IV-A). In Figure 4, we present a visualization built by VISION that represents two tenants and their respective forwarding graphs. In this visualization, the blue lines represent the service chaining of Tenant 1 and green lines the chaining of Tenant 2.

The Graphs Visualization module provides others visualizations that help in identification process of the distinct problems, such as the consumption resources profile of each tenant and visualization for the relationship between tenants and VNFs. Moreover, such module has several optional settings to improve the human perception and the flexibility of visualizations. For example, the operator can configure attributes, such as color and size of nodes to represent different information (*e.g.*, the level of CPU usage and packets processed). Thus, the human operator can suppose that a VNF node, represented by a big circle, received a lot of packets to be processed, and VNFs, with a red color node, the CPU usage saturated.

IV. CASE STUDIES

We analyze different scenarios to evaluate the technical feasibility of VISION platform. All of this case studies considers a VNFaaS (VNF as a Service) scenario, which contains VNFs from service provider's application and several costumers of the service (tenants). In this section, we detail three case studies focus on distinct problems that can occur in the NFV environment. First, we explore the graphs and the statistical visualizations modules to provide a way for determining VNFs bottlenecks. Second, we investigate affinities among VNFs, using the migration and statistical module. Finally, we present a visualization to understanding the necessities of each tenant. As a result of these analyzes, we observe the effectiveness and the feasibility of VISION.

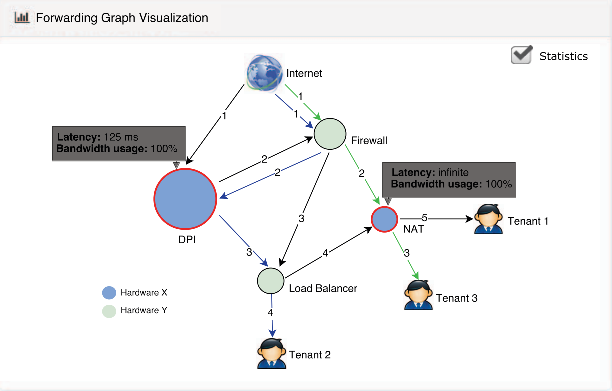
A. Case Study #1

Let's suppose a scenario with four VNFs (Deep Packet Inspection – DPI, Firewall, Load Balancer, and Network Address

Translation – NAT) and three distinct tenants sharing these VNFs, but with different forwarding graphs. In this scenario, the network operator receives several complaints from the tenant 1 about the performance of the contracted service. In a preliminary analysis, the operator does not have sufficient information to identify the cause of the problem, although the NAT performance presents abnormal behavior (saturated network bandwidth). Thus, a more in-depth analysis is required to determine the cause of the problem.

Another way to deal with the aforementioned problem is to analyze one by one all VNFs that compose the service chaining of tenant 1. However, this imposes several challenges (*e.g.*, verify a lot of statistics), increasing the time to identify the cause of the problem. Moreover, the operator must access the information about each forwarding graph, in order to understand how the VNFs are shared among tenants. In this way, the traditional management without visualization is not efficient to address the problem, because is not trivial for humans have insights into textual information about a lot of statistics.

Our platform improves the capacity of operator have insights about VNFs misbehavior that provoke problems in another VNFs. For this, the operator may interact with the Visualization Manager to access the global view about the forwarding graphs. In this visualization, each node represents a VNF and has the size defined by the number of packets that are processed. In addition, the nodes are represented with a red color border when some observed statistic is saturated (*i.e.*, network bandwidth usage higher than 90%). In addition, the color of edges represents the forwarding graphs of each tenant (black lines represents the tenant 1 service chaining, blue lines the tenant 2, and green lines the tenant 3). Also, the color of nodes represents the hardware where a VNF is hosted (blue nodes are hosted in one hardware and green nodes are in another) In the Web-based interface, the network operator defines saturation thresholds and chooses the statistics that will compound the visualization. Using this visualization, the operator can identify VNFs with abnormal behavior, analyzing individual statistics (*e.g.,* latency and network bandwidth) to find bottlenecks.



Forwarding Graph Visualization

☑ Statistics

Latency: 125 ms
Bandwidth usage: 100%

Latency: infinite
Bandwidth usage: 100%

Fig. 5: Forwarding graphs and statistics visualizations

In Figure 5, we present the visualization that helps the operator to identify the cause of the problem. The numbers in the edges represent the sequence of the service chaining. As can be observed, the NAT and DPI are saturated. But, it is not sufficient to understand the problem. However, we can note that the DPI is receiving more packets from the Internet that can process. Based on this visualization, we can see that the hardware that host both DPI and NAT are with the bandwidth saturated. As a consequence, the NAT cannot receive and deliver the packets to the next hop, because of the congestion caused by this saturation. Thus, we can conclude that the forwarding graph of the tenant 1 and 3 are crashed due to incorrect placement of VNFs.

The network operator must take decisions about how the DPI is running. For example, the Firewall perform fewer in-deep analyzes than DPI. Therefore, the operator changes the forwarding graph, putting the firewall to process packets prior to the DPI. In another solution, the network operator can maintain the original forwarding graph, performing changes directly in the DPI, such as create affinity rules to host the DPI in hardwares with more resources and capacities to sustain the high demands.

B. *Case Study #2*

The placement of VNFs focuses on finding a physical host that has the resources available to run properly the service. However, the placement of a set of VNFs in relation to each other is not considered, even though affinity and anti-affinity rules are recognized [19]. An affinity rule expresses that VNFs must be hosted on the same hardware, whereas anti-affinity requires that VNFs are instantiated on distinct physical machines. If these rules are not well defined, VNFs with anti-affinity can be placed on the same host. As a consequence, one VNF will present misbehavior and impact negatively in the service provided.

The identification of affinities among VNFs is a very complex task. Such complexity increases when the operator does not have the appropriate resources. For example, if operators try to understand the affinities issues just using traditional management tools. Thus, they will only access the available events logs and will spend a lot of time to analyze textual information about migrations of each VNF. Nevertheless, the identification of affinities issues will be not evident.

Let's suppose the following scenario: the network operator, based on previous reports, observes that a virtual instance of Firewall is presenting sudden changes in its performance along the day. The first analysis indicates that it caused several impacts in the service provided, such as increasing the propagation delay and spoiling the packet processing. Thus, the operator adopts VISION platform to try identifying and mitigating this problem. Initially, the visualizations provide information to conclude that VNF performance worsens when migrations events occur. Based on this, the network operator must analyze VNFs migrations to understand the problem.

In Figure 6, we present a migration history visualization and statistics from VNFs. In this visualization, the operator can select what statistics data will be available about VNFs, showing them in a line chart to understand the behavior of VNFs along the time. Meanwhile, another visualization provides a view about what physical structure hosts these VNFs. Thus, the operator uses both visualizations to understand the nature

Fig. 6: Visualization for VNFs migration history

of problems in the NFV network. In this visualization, a red dotted line can be moved to define what is the threshold for the observed statistic. With this, the operator has a more clear view when the statistic value is high (*e.g.*, when the latency is high). Besides, we can highlight the interval where the values are apparently fine.

The network operator has identified that the blue VNF (Firewall) is working fine when shares the same physical host that the red VNF (Load Balancer). It can be viewed in Figure 6 analyzing the times 07:00 to 08:00, 09:00 to 10:00, and 13:00 to 14:00. Based on this, the operator has an insight about the problem: the affinity among the Firewall and the Load balancer are not respected. Thus, in order to mitigate the problem, the operator must create one affinity rule to host both VNFs on the same physical hardware.

C. Case Study #3

The network operator manages the VNFs to provide a great service for the tenants. For this, the operator must have a view about the distribution of VNFs and the consumption profile of each tenant to support all performance requirements. The available information is fundamental to identify the specific demands and foresee the impacts of maintenance activities on the service. The current solutions for NFV management do not provide features that help the operator manage this information to have insights, which contribute to planning the network management. For example, the maintenance of determined VNF can have a significant impact on a high priority tenant (based on politics of priorities). Thus, the operator repairs VNFs when there is a minor traffic of such tenant, avoiding several consequences in service contracted. Visualization allows the operator to analyze the demands, and, thus, offering specific services for each tenant, such as instantiating a dedicated Firewall for one tenant or increasing the resources available for a VNF.

In an NFV environment, there are four shared VNFs (NAT, DHCP, Firewall, and Video Transcoding) among three tenants (Tenant 1, 2, and 3). The tenant 1 contracted the Firewall, DHCP, and NAT services, whereas the tenant 2 uses only the Firewall and DHCP. The VNFs correspondent to DHCP, NAT, and a particular Video Transcoding supply the tenant 3 demands (Figure 7). In this environment, the network operator receives a performance service complaint from a high priority tenant (tenant 1). Based on this, the operator must reassure that the hight priority tenant demands will be satisfied, considering that one or more VNFs are shared. For this, the information about the network and specifics demands of each tenant 1 must be analyzed to reorganize the distribution of VNFs. An example of these information are: *i)* what VNFs are being shared by two or more tenants, *ii)* which tenant use more determined VNF; and *iii)* what is the performance requisites for each tenant.

In order to provide ways for operator understanding the information from the network, we use a technique known as the forced-direct graph. This technique defines forces to attract or repulse pair of nodes. Thus, we can represent the proximity between VNFs and tenants based on the pre-defined statistic (*e.g.,* the VNFs are near to the tenants that send more packets to process). Using VISION interface, the operator can choose another statistic to create the forced-direct graph, such as physical resources usage and geographic localization. This visualization presents an overview of the different relationships among nodes, assisting in some NFV management issues, *e.g.,* the operator can perceive when the geographic location of the physical hosts is impacting in the VNFs performance.

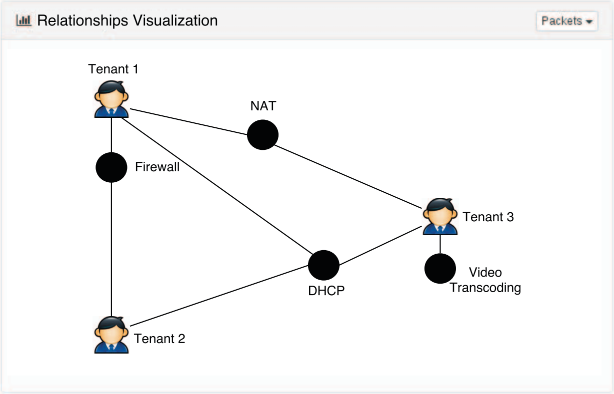


Fig. 7: Forced-direct graph based on the consume of each tenant

Figure 7 shows the relationships between VNFs and tenants (*i.e.,* which VNFs is available to each tenant). The number of packets processed defines the distance among two nodes. For example, this visualization suggests that an instance of Firewall, although shared by both tenant 1 and 2, is more requested (in terms of packets processed) for the tenant 1. Hence, this visualization provides a better view of the utilization profiles, helping the network operators to provide services that will attend the tenants' demands. Thus, the operator can offers for the tenant 1 a dedicated Firewall that prioritize it high demands.

V. CONCLUSION AND FUTURE WORK

In this paper, we introduced VISION (**Vis**ual**i**zation **O**n **N**FV), a platform that implements a set of interactive and selective visualization techniques. Such visualizations help network operators to identify and mitigate the cause of not obvious VNF problems, such as misconfiguration or misplacement. To sum up, in this paper we: *i)* implement an approach to collect and organize data from NFV environment, *ii)* provide visualization solutions to simplify the identification of the cause of some VNFs problems; and *iii)* help the operator to have insights about several issues related to NFV-enabled networks management.

Case studies reveal that VISION platform provides useful visualizations that simplify the understanding of the information and, as a consequence, the network operator can have insights into the cause of problems. In the Case Study #1, VISION provides a complete forwarding graph visualization, which allows the network operator to perceive incorrect VNF placement and VNFs overloaded. Otherwise, the operator needs to identify the correspondent forwarding graph and exhaustively analyze each VNF to understand the problem. In the Case Study #2, we transform migrations logs in a meaningful visualization that helps the operator to identify affinity/anti-affinity issues, based on the physical hardware that hosts a VNFs. Finally, in the Case Study #3 are presented helpful graphs visualizations to understand the demands of each tenant, allowing operators offer services to supply their demands.

As future work, we plan to extend VISION with novel visualizations to support another VNFs problems (*e.g.*, conflicting policies, incorrect service chaining, and Service Level Agreement issues). In addition, we plan to investigate about human-centric evaluation techniques to quantify the benefits of VISION platform. Also, we intend to extend our visualizations to support other issues, such as affinities among VNFs running parameters and security issues. Finally, we aim to support another implementation of NFV (*e.g.*, ClickOS and openNFV).

ACKNOWLEDGEMENTS

This work is supported by ProSeG - Information Security, Protection and Resilience in Smart Grids, a research project funded by MCTI/CNPq/CT-ENERG 33/2013.

REFERENCES

[1] ETSI NFV ISG, "Network Functions Virtualisation," Jun 2012, White Paper.

[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, pp. 18–26, 2014.

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

[4] V. T. Guimaraes, C. M. D. S. Freitas, R. Sadre, L. M. R. Tarouco, and L. Z. Granville, "A Survey on Information Visualization for Network and Service Management," *IEEE Communications Surveys and Tutorials*, 2015.

[5] W. Xu, M. Shehab, and G.-J. Ahn, "Visualization Based Policy Analysis: Case Study in SELinux," in *Proceedings of the 13th ACM Symposium on Access Control Models and Technologies*, ser. SACMAT '08. New York, NY, USA: ACM, 2008, pp. 165–174. [Online]. Available: http://doi.acm.org/10.1145/1377836.1377863

[6] F. Fischer, J. Fuchs, F. Mansmann, and D. Keim, "BANKSAFE: A visual situational awareness tool for large-scale computer neteorks," *Conference on Visual Analystics Science and Technology (VAST)*, pp. 14–19, 2012.

[7] N. Tateishi, M. Thara, N. Tahara, and H. Seshake, "Method for visualizing information from large-scale carrier networks," *Asia-Pacific Network Operations and Management Symposium (APNOMS)*, pp. 1–6, 2013.

[8] E. Glatz, S. Mavromatidis, B. Ager, and X. Dimitropoulos, "Visualizing big network traffic data using frequent pattern mining and hypergraphs," *ACM Computing*, vol. 96, pp. 27–38, 2014.

[9] F. Mansmann, T. Gobel, and W. Cheswick, "Visual analysis of complex firewall configurations," *Symposium on Visualization for Cyber Security (VizSec)*, pp. 1–8, 2012.

[10] Y. Kumata and A. Koyama, "Mesh Net Viewer: A Visualization System for Wireless Mesh Networks," *International Conference on Broadband and Wireless Computing, Communication and Applications (BWCCA)*, pp. 80–87, 2013.

[11] T. Nunnally, P. Chi, K. Abdullah, A. Uluagac, J. Copeland, and R. Beyah, "P3D: A parallel 3D coordinate visualization for advanced network scans," *International Conference on Communications (ICC)*, pp. 2052–2057, 2013.

[12] Q. Liao and A. Striegel, "Intelligent Network Management Using Graph Differential Anomaly Visualization," *Network Operations and Management Symposium (NOMS)*, pp. 1008–1014, 2012.

[13] P. H. Isolani, J. A. Wickboldt, C. B. Both, J. Rochol, and L. Z. Granville, "Interactive Monitoring, Visualization, and Configuration of OpenFlow-Based SDN," *International Symposium on Integrated Network Management (IM)*, pp. 207–215, 2015.

[14] Real Status, "Hyperglance - A visibly diffent approach to cross-domain, hybrid IT management," http://www.real-status.com/, 2015.

[15] Y. Zhu, "Introducing Google Chart Tools and Google Maps API in Data Visualization Courses," *IEEE Computer Graphics and Applications*, pp. 6–9, 2012.

[16] ETSI GS NFV-MAN, "Network Functions Virtualisation (NFV); Management and Orchestration ," Dez 2014, White Paper.

[17] Telefonica Research and Development, "OpenMANO framework," http://www.tid.es/long-term-innovation/network-innovation/telefonica-nfv-reference-lab/openmano, 2015.

[18] R. Tamassia, *Handbook of Graph Drawing and Visualization*, 1st ed. CRC Press, 9 2014.

[19] S. Oechsner and A. Ripke, "Flexible Support of VNF Placement Functions in OpenStack," *IEEE Conference on Network Softwarization (NetSoft)*, pp. 1–6, 2015.