Review Article GU J Sci 32(3): 833-851 (2019) DOI: 10.35378/gujs.422000

Gazi University

Journal of Science

http://dergipark.gov.tr/gujs

Virtual Security Functions and Their Placement in Software Defined Networks: A Survey

Sedef DEMIRCI*[1], Mehmet DEMIRCI[1], Seref SAGIROGLU[1]

[1]*Gazi University, Faculty of Engineering, Department of Computer Engineering, 06570, Ankara, Turkey*

Highlights

- We categorize virtual network security functions according to their types.
- We classify the studies on the optimal placement of virtual security functions in SDN.
- We identify important research challenges in the area of virtual security function placement.
- We propose promising future directions for virtual security function placement.

Article Info

Received: 08/05/2018
Accepted: 08/04/2019

Keywords

SDN
NFV
VNF placement
Network security

Abstract

Software Defined Networking (SDN) and Network Functions Virtualization (NFV) are two important technologies gaining prominence thanks to their benefits for improving the flexibility and cost efficiency in networks. These technologies have been utilized extensively for providing new age security solutions in recent years. Through the use of SDN and NFV, network security functions are virtualized and deployed in a hardware-independent manner, thus reducing costs as well as enabling faster innovations and developments. Functions virtualized with NFV such as firewall, deep packet inspection, intrusion detection systems etc. can reside as applications in the SDN architecture. The issue of where to place these functions in the network is an important problem discussed in the literature. When placing these functions, objectives such as efficient use of network resources, energy consumption, cost, network load, delay etc. must be considered for each function, in addition to ensuring that network security requirements are met. This paper provides a critical survey on the placement of virtualized network security functions in software defined networks and identifies open problems in this field. We briefly describe SDN and NFV technologies, touch upon the relationship between them, exemplify and review the most common virtual security functions in SDN. We also examine and compare the studies on the optimal placement of virtual security functions. Finally, we identify several open research challenges in this area and suggest potential future directions to be considered by researchers.

1. INTRODUCTION

Traditional IP networks are complex and difficult to manage in spite of their widespread use. In traditional network architecture, all network operations are performed by the network devices such as switches, routers, middleboxes etc. These devices are vendor-specific, and there may be many network devices produced by different vendors in this type of network architecture. Hence, when there is a need to make an operational change, all devices need to be reconfigured manually according to their vendor-defined low-level commands by the network operator. This situation increases both complexity and the probability of making mistakes. On the other hand, vertical integration, which means that software and hardware are supplied by a single vendor, slows down innovation and complicates making changes. Additionally, as a consequence of this hardware-centric architecture, researchers prefer to use testbeds, emulation and simulation environments to test their proposals instead of studying with real-world network topologies and traffic data [1].

*Corresponding author, e-mail: sedefgunduz@gazi.edu.tr

Software Defined Networking (SDN) is a new technology which emerged with the objective of making networks programmable, thus solving the problems mentioned above and facilitating innovation. The main philosophy of SDN is to decouple the control plane from the data plane. It separates the task of controlling network behavior from the underlying devices and delegates this task to the logically centralized controller software. The network devices in the data plane are only responsible for forwarding packets according to the rules determined by the controller software. Thus, vendor dependency is reduced and hardware-independent development is provided. Furthermore, it becomes easier to analyze and predict network behavior thanks to the logically centralized nature of the controller software [2, 3].

Network Functions Virtualization (NFV) is another new technology which emerged independently from SDN but tries to tackle the same issues of vendor dependence and making network operations programmable. NFV virtualizes network functions such as network address translation, firewall, load balancing etc. which reside in separate middleboxes in traditional networks and implements them as software on commodity servers. Thus, it provides cost efficiency by eliminating both special hardware cost and the energy cost arising from operating a separate device for each function in various parts of the network. Moreover, the time needed to deploy updates is reduced while openness and innovation are facilitated [4].

Even though SDN and NFV are two nascent technologies coming from different sources as explained in Section 2 and can be implemented independently, they complement each other very well and can attain their full potential when they coexist, because both of them follow the basic principles of network agility, cost efficiency and defining network behavior with software. Therefore, when these technologies are used together, network control logic is abstracted from the forwarding mechanism and network functions are implemented as virtualized software [5].

Information security is one of the most significant challenges for SDN, as in all areas of technology. In a network structured with SDN and NFV technologies, security can be provided by virtualized security functions such as intrusion detection/prevention systems (IDS/IPS), deep packet inspection (DPI), firewall etc. [6]. The issue of where to place these functions in a network is a new and important problem faced by network operators. When determining the locations of these functions, security policies of each function, like filtering traffic that meets specific rules, or performing deep inspection on some packets from specific flows, etc. must be considered. However, this is not sufficient for optimality in network operation. Objectives such as efficient use of network resources, energy consumption, cost, network load, delay etc. must also be taken into account in order to find the optimum places for these functions [7]. In this work, we review and evaluate the studies on the placement of virtual security functions in SDN. We identify the gaps in the literature and provide ideas for future work. To the best of our knowledge, this is the first survey study presented in this field. The contributions of the paper are listed below:

- We review the works proposing solutions based on virtual network security functions in the literature and categorize them according to their types.
- We focus on the studies about the optimal placement of virtual security functions on SDN, classify them by their objectives and compare them along many dimensions.
- We identify several research challenges in the area of virtual security function placement and propose promising future directions.

The rest of the paper is organized as follows. We give a brief explanation of SDN and NFV technologies in Section 2. In Section 3, we provide some examples of virtual security functions and discuss their implementations in SDN. We review and compare the studies on optimal placement of virtual security functions in SDN in Section 4. We present some open research challenges and potential future directions in Section 5. Finally, in Section 6, we summarize our points and conclude the paper.

2. BACKGROUND: SDN and NFV

2.1. Software Defined Networking (SDN)

SDN is a networking philosophy with two main tenets: separating the control logic from forwarding devices, and giving the control to a centralized software program, known as the controller [1]. In other words, the control plane and the data plane are decoupled. The software controller is in charge of determining the rules that govern how forwarding is done, and the programmable forwarding devices on the data plane are responsible for routing the traffic according to the rules defined by the controller [8]. SDN architecture also includes a higher layer called the application layer where various application programs reside. Many of these programs provide some kind of network function such as monitoring, load balancing, proxy, firewall, intrusion detection system etc. [2]. Figure 1 illustrates the SDN architecture as presented by the Open Networking Foundation (ONF) [9].

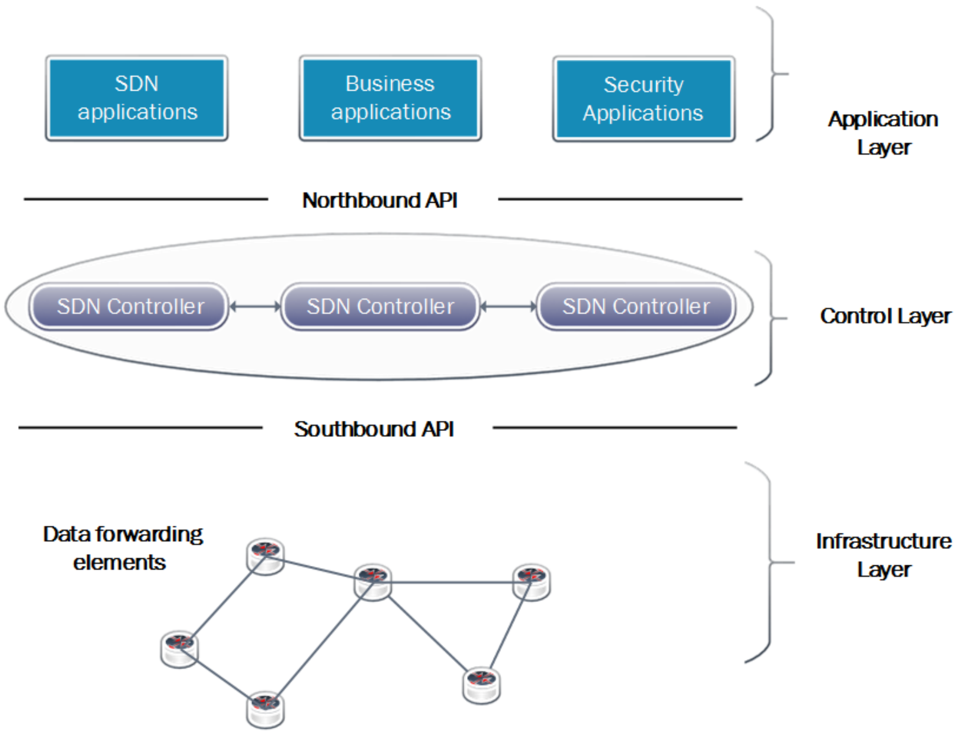

Figure 1. *SDN layers*

Below, we explain the principles of SDN in more detail, describing the properties of each layer and the interfaces between these layers.

2.1.1. Infrastructure layer

Infrastructure layer, also known as the data plane, consists of programmable packet forwarding devices. Unlike traditional network devices, these elements typically do not run distributed routing protocols and cannot make autonomous decisions. Instead, these programmable routers and switches forward packets according to the rules defined by the controller on the control layer [10,11]. These rules are stored in flow tables on a switch and can be modified through an interface called Southbound API. This makes data plane elements simpler and eliminates vertical integration.

2.1.2. Southbound API

The protocol for the communication between the control plane and the data plane is defined by the Southbound API. Currently, OpenFlow [12] is the most common protocol and de facto standard for Southbound API. Since its development at Stanford University in 2008, OpenFlow has been adopted in various domains including data centers, backbone and enterprise networks. OpenFlow messages flow in both directions: Rules and updates go from the controller to the switches, while monitoring information and packets not matching any of the current rules on a switch go in the opposite direction [12,13].

2.1.3. Control layer

This layer hosts the controller, which is the most critical component of a software-defined network. In fact, there may be multiple controllers acting in a coordinated fashion, or organized in a hierarchy. The control layer manages the forwarding behavior and is considered the brain of the network. The controller determines forwarding rules and uses a Southbound API to transmit these rules to the programmable switches in the data plane [14]. There are many SDN controllers (NOX, POX, Floodlight, Beacon, DIFANE, OpenDaylight etc.) in active use with varying goals and priorities [1].

2.1.4. Northbound API

As the complexity of the control logic increases, programming the controller directly becomes more difficult. Higher-level SDN programming languages and compilers are needed to make application development more accessible in this architecture. Northbound API provides the tools necessary for fast development in SDN, including a language with a more familiar syntax and a runtime to translate program code into flow rules to be installed on data plane devices [15].

2.1.5. Application layer

The orchestration and management of a large software-defined network is generally performed by a range of coexisting modules or functions such as routing, monitoring, load balancing, intrusion detection, traffic filtering, deep packet inspection etc. [16]. Each function or application follows certain policies to make changes to the network. It is the responsibility of the Northbound API to ensure harmonious operation in the network by appropriately prioritizing rules and managing conflicts [1].

2.2. Network Function Virtualization (NFV)

Today, network operators mostly rely on advanced purpose-built hardware devices called middleboxes for important networking functionality. The downside of this approach is the difficulty and costliness of adding new services to a network: One needs to select the number and locations of the appliances providing the desired service in accordance with budget restrictions and service-level agreement requirements, make devices operate in a coordinated manner, and sometimes redesign the entire network architecture. Moreover, deploying a dedicated device for each separate network function is neither cost-effective nor energy efficient [17].

Network Functions Virtualization (NFV) is a nascent technology offering a solution to the above mentioned issues by decoupling network functions from dedicated hardware and implementing them as software on standard high-volume servers [18]. Figure 2 shows how the NFV approach works [19]. With NFV, virtualized network functions are easily deployed on existing servers in appropriate locations, making it possible to remove most of the special hardware appliances from the network. Similar to SDN, NFV reduces hardware dependency, cost and energy consumption. Other benefits are accelerated innovation and decreased time-to-market for network services since they are provided as software [4]. Furthermore, services can share physical resources efficiently and remote installation, update, and relocation become possible.

2.3. The Relationship between SDN and NFV

SDN and NFV have many similar goals, including reduced cost, increased manageability, and faster innovation. The SDN approach is based on centralizing and softwarizing the decision making process in the network while taking advantage of the programmability of forwarding devices. On the other hand, NFV focuses on eliminating special-purpose networking hardware as much as possible by virtualizing their functions and deploying these functions on general-purpose high-capacity servers. In other words, SDN provides a programmable network where virtual network functions (VNFs) can shape the forwarding decisions [20].

Although it is possible to deploy SDN and NFV independently, these technologies are highly complementary. In fact, the full potential of both technologies can be realized when they are used together. NFV can utilize traditional virtualization techniques to deploy VNFs without SDN. However, when coupled with SDN, it becomes easier for NFV to reach its goals thanks to increased agility, quicker

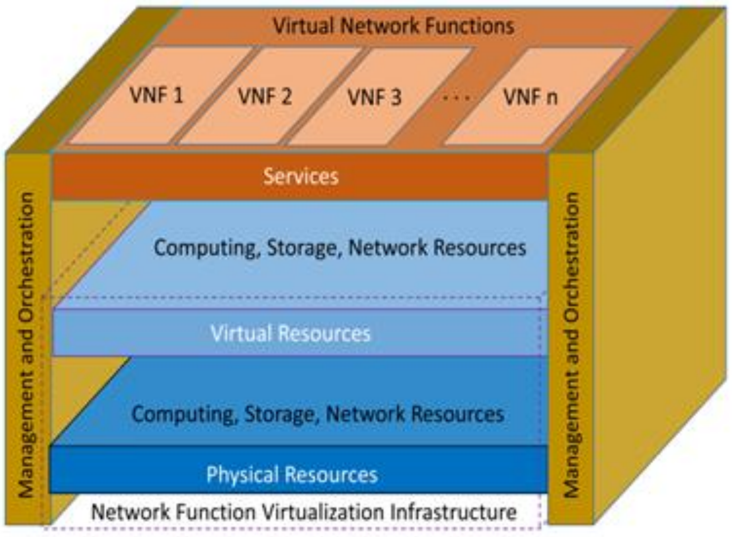
Virtual Network Functions
VNF 1
VNF 2
VNF 3
...
VNF n
Services
Computing, Storage, Network Resources
Virtual Resources
Computing, Storage, Network Resources
Physical Resources
Network Function Virtualization Infrastructure
Management and Orchestration
Management and Orchestration

Figure 2. *How the NFV approach works [19]*

update ability and easier maintenance provided by a programmable network infrastructure under central control [9,17].

3. VIRTUAL SECURITY FUNCTIONS IN SDN AND THEIR CLASSIFICATION

In traditional networks, security functions such as firewalls, intrusion detection/prevention systems, and deep packet inspection (DPI) etc. are implemented on vendor-specific appliances or middleboxes located at fixed points [6]. On the other hand, in software defined networks intertwined with the NFV technology, these functions are virtualized and placed in appropriate virtual machines to be executed [21,22]. According to Hu and Ahn [6], there are three types of virtualized network security functions which are attack detection, attack prevention, and attack capturing functions. We propose a different classification with four types of functions: Attack detection, prevention, deception, and mitigation. Attack deception differs from prevention in that its goal is to confuse or mislead the attacker instead of just blocking the attack. In this way, defenders are able to collect valuable information about attacker behavior and make attackers waste their own resources. Attack mitigation is also different because it tries to limit the impact of attacks through a combination of various functions (including those from the other classes) when complete prevention cannot be achieved. For these reasons, we believe that these two categories should be separated from the others. The rest of this section provides an overview of the recent studies in each of the above mentioned categories.

3.1. Attack Detection Functions

3.1.1. Intrusion detection system (IDS)

Intrusion Detection Systems are the passive security functions that monitor the network, detect the malicious activities and policy violations and report them to system administrators [23]. IDSs on traditional networks use signature-based, statistical, and stateful protocol analysis methods. They utilize system records, services and node messages; hence detecting attacks to virtual machines is very hard and costly

[24]. Also, performance is adversely affected as the network grows and complexity increases [25]. On the other hand, in SDN, collecting statistical data, making sense of it and routing the traffic is easier than the traditional networks due to its software based architecture. Using these advantages, Van Adrichem et al. [26] developed an IDS named as OpenNetMon working on POX controller. OpenNetMon detects the intrusions by evaluating the throughput, packet loss and delay metrics. The central view of the SDN controller enables access to a large amount of data to be analyzed and interpreted. Tang et al. [27] employ deep learning as the basis for an IDS which works in this manner at the control layer.

3.1.2. Malware scanner

Malware scanners protect the local network from the malicious software on the internet such as viruses, worms, trojans etc. Ceron et al. [28] proposed a malware analysis architecture by utilizing the flexibility of SDN. The inspection module which is implemented on top of the SDN controller analyzes network flows by looking for pre-defined patterns. If malicious traffic is detected, the containment module prevents the malware from communicating with other system elements, and the configuration manager modifies the network topology dynamically according to the changing traffic characteristics. Experimental results show that the proposed system has the ability to analyze advanced malwares dynamically, and detect more malware events than traditional solutions.

3.1.3. Distributed denial of service (DDoS) detector

In DDoS attacks, a target computer is overwhelmed by sending a high volume of fake requests, thus it cannot process real requests and remains out of service. There are a variety of solutions to detect DDoS in traditional networks; but most of these solutions require analyzing large numbers of packets; so their accuracy level decreases while response time increases [29,30].

However, in SDN, DDoS attacks can be detected more effectively with better response time since numerous switches can be directly controlled by the controller simultaneously. In this regard, Braga et al. proposed a DDoS detector function on NOX controller, and they easily add and remove the switches which are responsible for DDoS detection from the network. Also, they can easily include new attack types to the system [30].

3.1.4. Deep packet inspection (DPI)

DPI is an advanced method for analyzing flows and user activities in detail and real-time. DPI engines examine the payload portion of a packet in terms of traffic type, protocol compliance and malware content in addition to the packet headers. In case of a suspicious event or an attack threat, DPI reroutes the packet to a different destination or reports it to another security tool [31]. Thus, it aims to protect the system from attacks, improve the performance, reduce bandwidth costs, control the congestion, and enhance the quality of service [32].

In traditional networks, DPI engines which are implemented on hardware middleboxes are placed in specific locations on the network. However, with SDN and NFV, DPI tools are virtualized and dynamically deployed [7,31]. In this regard, in [33], the authors implemented a DPI engine on an SDN controller as a function and they show that performance of the network can be improved by up to 67%.

3.2. Attack Prevention Functions

3.2.1. Firewall

Firewall is the first level of access control placed between the local network and the internet. It protects the local network from untrusted devices outside. In traditional networks, due to the fixed firewall position, internal traffic cannot be seen and audited. With SDN, all internal traffic can be filtered because the firewall is implemented as a software function independent of physical location. Packet filtering is only one aspect of firewall design in SDN. Compliance with firewall rules and dynamic flow rules is also verified since network conditions, configurations and flow rules can change dynamically. In addition, firewall placement architecture (central or distributed) and the exact places for this function should be determined carefully [34,35].

There are two types of firewalls in traditional networks: stateless and stateful. In stateless firewalls, packets are filtered by checking only IP address and port number, while stateful ones track the connection status, protocols, and source and destination ports. In other words, stateful firewalls accept packets from only the flows known and established before [36]. However, firewalls in a software defined network apply stateless filtering because OpenFlow offers very limited information about the connection to the controller. Therefore, making SDN firewalls completely stateful is an important problem that should be addressed [34].

There are a certain number of studies proposing firewall systems for SDN [12,34,37]. In this regard, Hu et al. developed a firewall solution called FlowGuard. It can resolve policy violations automatically and in real-time when networks conditions changed [34]. Deng et. al [18] claimed that the traditional firewall approach cannot be used in SDN effectively, and they proposed the VNGuard framework for managing virtual firewalls with NFV. VNGuard defines a policy language for virtual firewalls to produce high-level policies. In addition, it involves a module for finding the optimum places for the policies defined by the functions.

3.2.2. Intrusion prevention system (IPS)

IDSs only detect and report attacks, that is, they do not proactively prevent the attacks in early stages or even before they start. Therefore, intrusion prevention systems are needed in order to act quickly against suspicious network activities. Traditional IPSs are implemented on IDSs because they primarily need the attacks to be identified. However, this architecture is not extensible. Although many are open source software tools, different coding styles, development environments and interfaces make it difficult to deploy these systems. In addition, producing dynamic solutions to changing network states is quite difficult with the IPSs on traditional networks [24].

In SDN, on the other hand, IPSs detect and prevent intrusions in a more agile, dynamic manner, and with lower cost. In the literature, the number of studies developing IPS for SDN is fewer than other security functions. Zhang et al. [38] proposed an SDN based IPS and a load balancing function. Experimental results show that the proposed IPS model can detect attacks in shorter time and reduce latency with load balancing.

3.2.3. Anti-spoofing

Spoofing in different network protocols is a widespread technique utilized by adversaries to combat attack detection and mitigation. Proposals against IP spoofing generally involve the use of cryptographic primitives such as message authentication codes and hash functions, as well as filtering tables maintained by SDN controller applications [39,40]. Similarly, SDN application modules have been developed to prevent ARP spoofing by performing validation in real time using dynamic MAC-IP association lists [41], or replacing the header fields of ARP request packets with safe dummy values to protect switches from cache poisoning [42].

3.3. Attack Deception Functions

3.3.1. Honeypot

Honeypot is software with deliberately placed vulnerabilities to monitor penetrations and intrusions, learn about attack methods and detect new types of attacks. In SDN, the systems requiring dynamic learning and routing like honeypots attain better results than in traditional networks since the infrastructure is controlled by software [43]. In this regard, Shin et al. proposed a framework (Fresco) in which there are several security modules as well as honeypot. Fresco sends traffic to the honeypot when it detects a malicious connection request. Thus, the attackers think they connected to the real target and cannot penetrate the original system [44]. Another honeypot-based system called HoneyMix [45] utilizes the programmability in SDN to exercise fine-grained control on network traffic and keep attackers occupied on the honeynet as long as possible to prevent them from threatening valuable targets.

3.3.2. Moving target defense

Adversaries rely on information gathered about a network before planning and executing their attacks. Moving target defense (MTD) is a strategy for reducing the success of this reconnaissance effort by obscuring the structure of a network or deliberately deceving the attacker with false information. For instance, IP address mutation methods have been proposed for concealing the actual IP addresses of hosts and making only frequently changed virtual IPs visible [46,47]. In SDN, such techniques can easily be deployed as a controller module working in concert with DNS resolvers and programmable switches. Other methods make use of virtual topologies to provide users with inaccurate views of the underlying network [48–50]. Comprehensive deception solutions should integrate various approaches like resource hiding, address mutation, honeypots, fake network component creation etc. through dynamic configuration of many network services including NAT, DHCP, ARP, and DNS.

3.4. Attack Mitigation Functions

Mitigation systems are commonly proposed as a layer of defense against DDoS attacks attempting to disable the targeted network. LFADefender [51] is such a system offering flexible and cost-efficient defense mechanisms against link flooding attacks (LFA) in SDN. These concentrated attacks are use numerous low-rate flows to congest a small number of critical links. Hence, they prove costly for the attacked network as they are difficult to detect and stop. LFADefender employs continuous link monitoring to pinpoint the target links by examining flow density and congestion levels. If it detects congestion on a link, it then mitigates the problem by rerouting a portion of traffic from that link to multiple other paths. After this step, it conducts further analysis to identify bots and blocks them completely.

Many LFAs such as crossfire attacks [52] start with a reconnaissance phase to assess the criticality of links using tools such as traceroute. Researchers have proposed a solution [53] against such attacks based on link profiling and traffic diversion in SDN. This mitigation solution first analyzes the concentration of traceroute attempts to find out which links are likely to be targeted by the attacker, and then diverts ICMP packets to other routes randomly in order to obfuscate the attacker's view.

4. PLACEMENT OF VIRTUAL SECURITY FUNCTIONS IN SDN

One of the promising application areas for NFV is providing network security. In an NFV based network, security is provided by deploying virtual security functions such as intrusion detection/prevention systems (IDS/IPS), deep packet inspection (DPI), firewall etc. However, one of the important new challenges faced by network operators is where to deploy these virtualized functions. To illustrate, assume a scenario in which a DPI and a firewall need to be deployed in order to provide secure communication. If we put the DPI before the firewall, all traffic flows will be examined deeply even though some of them would have to be filtered out by the firewall, meaning DPI resources are being wasted. On the other hand, if we place the firewall before DPI, then unallowed flows will be eliminated and only allowed flows will be inspected deeply. Furthermore, if these two functions are placed on two different servers, energy cost will be higher than the case when they are consolidated on a single server. However, the latter choice may lead to uneven load and reduced reliability. Hence, optimizing the placement of virtual security functions for various objectives is an important research issue [54–56]. From the operational point of view, these functions must be placed with regard to possibly conflicting objectives such as cost, energy consumption, load balancing, coverage, delay, congestion, and traffic management as well as meeting network security requirements [7,57,58]. Therefore, there is a need for virtual function placement solutions that simultaneously respond to the operational requirements of the network and do not compromise the security policies [21,59,60].

Generally, studies on virtual security function placement focus on finding the optimal solution in terms of cost efficiency. In the literature, researchers define cost in different ways. More clearly, some studies consider the number of deployed functions and license fees to optimize cost, while others try to minimize the used network resources such as additional used bandwidth, number of links, CPU usage, etc. In this regard, we classify the parameters used to define cost in two categories: CAPEX (Capital expenditures) and OPEX (Operating expenses). CAPEX includes infrastructure costs such as purchased hardware, software license fees, installations etc. On the other hand, OPEX comprises network operational costs such as network planning, provisioning, reconfiguration and the usage of network resources etc. [61].

In this section, the studies on the placement of virtual network security functions in software defined networks are summarized and categorized. In this regard, we classify existing solutions based on the cost parameters defined by researchers as shown in Figure 3.

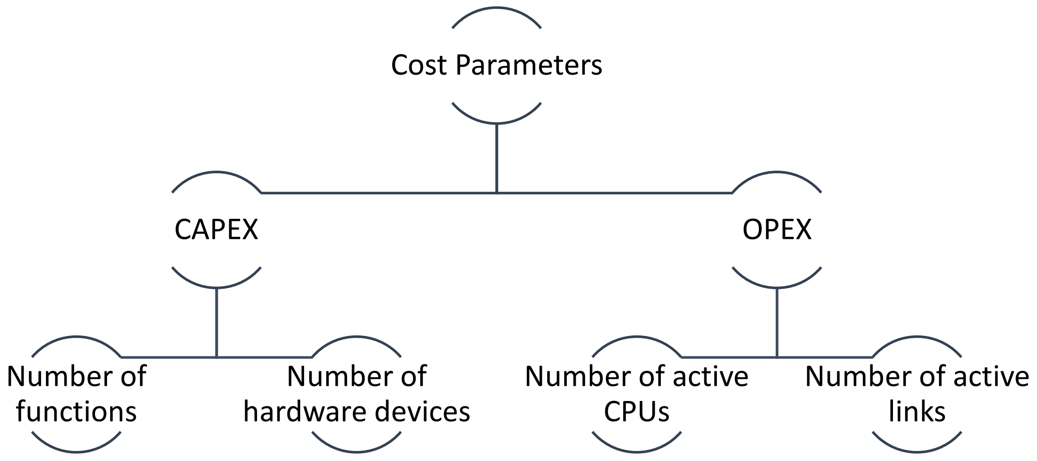

Figure 3. *Parameters used to minimize cost*

4.1. Minimizing CAPEX

The NFV approach reduces CAPEX by eliminating the need for single-purpose hardware appliances in networks. CAPEX can be further reduced by minimizing the number of virtual functions, and thus the money spent for purchasing and deploying them on servers. Murukan et al. [62] studied the placement of multiple virtual security functions with the objective of minimizing the number of activated functions. Their optimization approach, which is based on genetic algorithm, allows specifying ordering constraints for functions. However, the authors do not take into account the specific needs, resource consumption parameters, and security requirements of different types of functions.

4.2. Minimizing OPEX

One of the most important factors affecting the network operating cost is energy consumption of the servers. To address this issue, an energy-aware virtual security function placement model is presented in [63]. This study aims to place virtual security functions at optimum locations while minimizing server energy consumption. For this purpose, an ILP model is proposed subject to strict security constraints which can be defined as “every traffic flow must pass through all security functions once”. The model is tested on two different network topologies, and it has been shown that energy consumption of the servers can be reduced significantly while providing security at the desired level.

When placing multiple virtual security functions in SDN, rules of these functions should not conflict with each other. Therefore, sequential order of them should also be taken into consideration. Shameli-Sendi et al. [64] studied placing virtual security functions in specific order while minimizing latency and computing costs. They specified a security defense zone to place all functions and formulize the problem with integer linear programming. It is aimed to route traffic over less node in a short time. They tested their algorithm in OpenStack on OpenDaylight controller. Although the optimal placement cannot be found in reasonable

time for large problem sizes, proposed algorithm produces better results than the current placement algorithm of OpenStack.
Doriguzzi-Curin et al. [65] consider the specific security and QoS requirements of each user application, and they aim to minimize total bandwidth used. Differently from other studies, they take security constraints into account instead of focusing on only cost optimization constraints. An important advantage of this model is that it solves the placement problem for dynamic network scenarios where the service requests change over time. The evaluation results show that the proposed method can reduce the average end-to-end latency by 2-3 times when the nodes in the network are partially busy.

4.3. Minimizing CAPEX and OPEX Together

Bouet et al. [31] stated that deploying DPI functions is costly in terms of license fees. It is needed to deploy DPI engines cost effectively to be able to meet network security constraints. For this purpose, the authors proposed a genetic algorithm based approach that minimizes the number of engines and the network load at the same time. Experiments are conducted with different traffic types and the results show that global cost can be reduced up to 58% with this multi-objective optimization approach. However, this approach is not scalable for larger networks. Therefore, the authors solved the same problem with integer linear programming [7] and reduced the complexity in [31] with their graph based greedy algorithm. Experiments conducted with real traffic data showed that the proposed heuristic is 20-25 times faster than the integer linear program and it is a scalable solution applicable in large networks.

Jarraya et al. [21] proposed a framework named OCDO (Ordered Cloud Defense Optimization) for the same problem as in [62] and [31]. OCDO determines the proper order for security functions by calculating their priorities. The main goal is to perform these tasks with minimum cost in a scalable manner. The authors formulated the problem with integer linear programming as well, but unlike others they used divide-and-conquer technique in order to solve the problem in a reasonable time. Experimental results show that OCDO can reduce the cost at reasonable levels even for very large data centers and it is a scalable framework for optimal placement of security functions in a short time.

In a study by Shameli-Sendi et al. [66], a Security Defense Patterns Aware Placement (SDPAP) approach is proposed, and security constraints are taken into account as well as cost optimization constraints. The main difference of this study is the consideration of network security defense patterns (NSDP), which are identified by network security experts, in determining different placement solutions. Thus, the authors are able to prevent incorrect or inefficient placement of security functions, such as deployment of an IDS at the core network while the VPNs are located next to the edges. They capture the needs of each customer's application requiring different levels of security, place the security functions at optimal locations according to those needs, and route the traffic through the appropriate security functions. In this way, they claim to provide a more resilient approach to attacks compared with the existing solutions in the literature.

Table 1 provides a summary of the studies discussed above. As we see, there are some gaps in the literature such as the lack of dynamic flow handling strategies and solutions tailored to the needs of diverse application domains. The next section elaborates on future research challenges in this area.

5. RESEARCH CHALLENGES and FUTURE DIRECTIONS

In Section 4, the studies on optimal placement of virtualized security functions in SDN are presented and they are summarized in Table 1. In this section, open research challenges are identified by analyzing the previous section, and illustrated in Figure 4. In addition, several future research directions are provided for virtual security function placement problem.

5.1. Challenge 1: Different Types of Security Functions

Studies in the literature are generally tackling the placement of virtual DPIs and firewalls [7,18,31,64]. However, the security functions that must be deployed on the network are not limited to firewalls and

Table 1. *Studies on placement of virtual network security functions in SDN*

Authors	**Problem**	**Cost Parameters**		**Flow Handling Strategy**		**Optimization Metrics**	**Optimization Method**		**Technologies, Tools, and Data**	**Application Domain**
		CAPEX	OPEX	Static	Dynamic		Integer Linear Programming	Heuristic		
Bouet et al. [7]	DPI placement	✓	✓	✓		DPI number, additional used bandwidth	✓	✓	GLPK [67], Java Universal Network/Graph Framework [68] , GEANT [69]	General
Jarraya et al. [21]	Security functions chaining and placement	✓	✓	✓		Number of functions and links	✓		OpenStack [70], OpenDaylight [71]	Data Center
Bouet et al. [31]	DPI placement	✓	✓	✓		DPI number, additional used bandwidth		✓	JGAP [72]	General
Murukan et al. [62]	Security functions placement	✓		✓		DPI number, additional used bandwidth		✓	OpenStack [70], OpenDaylight [71]	Data Center
Demirci et al. [63]	Security functions placement		✓	✓		Number of activated servers	✓		CPLEX [73], Internet2 [74]	General
Shameli-Sendi et al. [64]	Sequential firewall placement		✓	✓		Time required by the traffic to traverse the links, and to be proceeded by the functions	✓		OpenStack [70], OpenDaylight [71], GLPK [67]	Data Center
Doriguzzi-Corin et al. [65]	Security functions chaining and placement		✓		✓	Total used bandwidth	✓	✓	Gurobi [75], Python [76], GARR [77]	Data Center, Backbone
Shameli-Sendi et al. [66]	Security functions chaining and placement	✓	✓	✓		Number of functions and used bandwidth by all flows	✓	✓	OpenStack [70], OpenDaylight [71], GLPK [67]	Data Center

DPIs, if there is a need for ensuring integrity, confidentiality, and availability of network communications. For example, IDSs are required to detect suspicious network activity, or IPSs must be deployed for blocking unwanted activity in early stages [78]. There appears to be a lack of studies addressing the placement of virtual IDSs, IPSs, DDoS detectors, malware scanners, honeypots etc. on SDN-enabled networks. In this regard, placing new security function types optimally by determining their requirements, constraints and missions entirely would be a good line of future research.

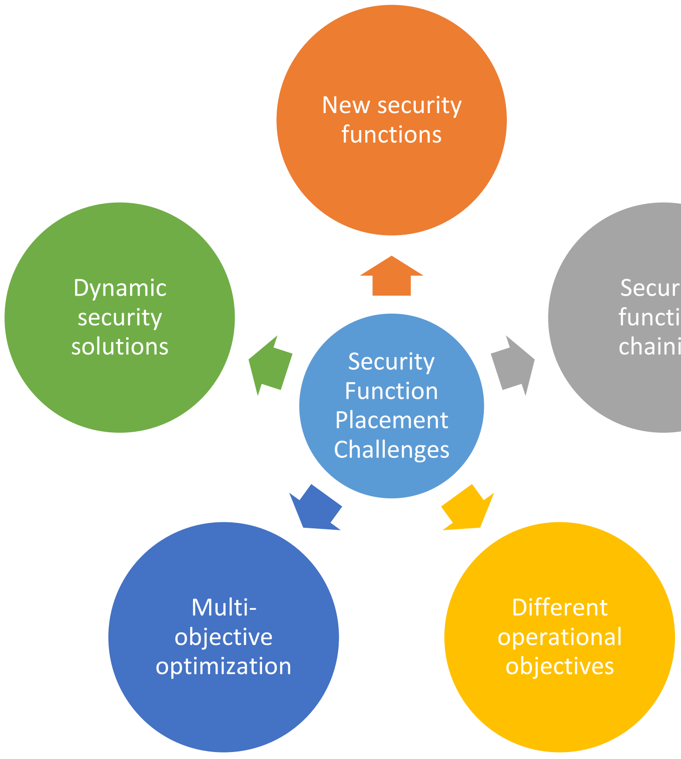
New security functions
Dynamic security solutions
Security function chaining
Security Function Placement Challenges
Multi-objective optimization
Different operational objectives

Figure 4. *Security function placement challenges*

5.2. Challenge 2: Security Function Chaining

The ordered placement of security functions and subsequently routing traffic flows through those functions is known as service function chaining [79,80]. Considering the deployment order of these functions must be required in order to provide security effectively and efficiently. For example, assume a scenario in which all traffic flows must be inspected by the IDS. In such a case, if we place a set of virtual VPNs before the IDS, the traffic flows passing through the VPNs could not be fully analyzed by the IDS since they are encrypted. Therefore, considering placement of only one security function is inadequate and the placement approaches must be extended for several virtual security functions. However, the studies focusing on more than one security function evaluated them in a general framework. They did not dwell on scopes of these functions [21,64–66]. Therefore, determining the types and requirements of security functions to be placed, analyzing the interactions between them, and taking the placement order of those functions into account is a promising direction for SDN and network security applications. The placement process must take into account the implications of coexisting network security functions, including but not limited to the dependencies and contradictions that may emerge.

5.3. Challenge 3: Different Operational Objectives

Generally, the main objective of virtual security function placement solutions is finding the optimal solution from the cost point of view. Additionally, they aim to provide good solutions in terms of performance, delay, and scalability [7,18,21,31,62,64–66]. Unlike others, an energy-aware virtual security function placement model is addressed in [63], which reduces the energy consumption significantly while providing security at the desired level. However, we have not come across any study on optimal placement of virtual security functions in a software defined network optimizing resilience, fault tolerance, and robustness etc. Therefore, there is a need for new models optimizing mentioned objectives for real world problems.

5.4. Challenge 4: Multi-Objective Optimization

As mentioned before, virtual security function placement solutions generally aim to optimize capital and operational expenditures. However, this problem is a multi-objective optimization problem by its nature, because many potential optimization objectives in networking are in conflict with each other, such as maximizing energy efficiency and fault tolerance, or minimizing delay and operational costs [81].

In the literature, there are only a few studies aiming to optimize for conflicting objectives [31,62]. For instance, Bouet et al. [7,31] focused on minimizing the number of virtual DPI engines and network load at the same time. Therefore, after pinpointing new optimization objectives, future efforts can concentrate on optimizing them simultaneously by developing new multi-objective optimization algorithms.

5.5. Challenge 5: Dynamic Security Scenarios and Solutions

In order to achieve network security objectives, there is a need for dynamic, real-time placement solutions [82–84] since changing deployment locations of the functions, or adding a new security function may be of vital importance in case of an attack or crash. Dynamic security solutions could be provisioned in two ways: in one type of solutions, function placement does not change, instead paths traversed by flows are recalculated [85]. However, this approach may be inefficient when the set of flows change drastically over time. In another type of solutions, locations of function are redetermined depending on the new network conditions and requirements. This would be a more resilient solution but may lead to a more complex problem in terms of scalability [22].

As shown in Table 1, there is only one study that proposes a dynamic solution [65], and it dynamically invokes the related function chains according to the changing traffic characteristics. Other studies [7,21,31,62,64–66] did not provide any solution on how the placement should be changed dynamically in case of a new security threat, attack, failure, or natural disaster etc. Providing uninterrupted service in the face of such problems is a critical issue. In this regard, future research efforts must focus on developing models, algorithms, and solutions for dynamic planning and placement of functions after identifying the cases when placement should be changed.

6. CONCLUSIONS

Software defined networking has emerged as a promising new technology with significant implications for the future of the Internet. The influence of SDN is growing at a fast pace, and SDN-based solutions are being used in many domains ranging from data centers to campus networks. Improving security has been recognized a primary goal, as the fine-grained control power and flexibility offered by SDN have been beneficial in developing new security applications.

In this survey we provide an overview of virtualized network security functions in SDN, with a focus on the optimal placement of these functions. We give a background for SDN and NFV concepts. We address network security functions in SDN by comparing what happens in traditional networks against what is happening in SDN. We review the existing studies on the optimal placement of virtual network security functions by comparing their cost parameters, optimization metrics, methods, technologies, tools and the data used for evaluation.

There are many open research issues in this field. Studying the deployment of different security function types is one of these issues as the variety of security functions considered should be increased. Additionally, the interactions and dependencies among them should be taken into account when tackling the placement problem, which is also known as security function chaining. Moreover, objectives such as energy efficiency and resilience should be incorporated into the solutions, possibly with a multi-objective optimization approach. It should also be noted that dynamic and agile solutions are necessary for responding to changing network conditions, security requirements and priorities.

CONFLICTS OF INTEREST

No conflict of interest was declared by the authors.

REFERENCES

[1] Kreutz, D., Ramos, F.M.V., Verissimo, P.E., Rothenberg, C.E., Azodolmolky, S., Uhlig, S. "Software-defined networking: A comprehensive survey", Proceedings of the IEEE, 103: 14–76, (2015).

[2] Feamster, N., Rexford, J., Zegura, E. "The Road to SDN: An Intellectual History of Programmable Networks", ACM SIGCOMM Computer Communication Review, 44:87–98, (2014).

[3] Nunes, B.A.A., Mendonca, M., Nguyen, X.N., Obraczka, K. , Turletti, T." A survey of software-defined networking: Past, present, and future of programmable networks", IEEE Communications Surveys & Tutorials, 16 :1617–1634, (2014).

[4] Han, B., Gopalakrishnan, V., Ji, L., Lee, S. "Network function virtualization: Challenges and opportunities for innovations", IEEE Communications Magazine, 53:90–97, (2015).

[5] Internet: ETSI - NFV, https://www.etsi.org/technologies-clusters/technologies/nfv (accessed January 18, 2019).

[6] Hu, H., Ahn, G.-J. "Virtualizing and Utilizing Network Security Functions for Securing Software Defined Infrastructure", NSF Workshop on Software Defined Infrastructures and Software Defined Exchanges, Washington, D.C., USA, 70, (2016).

[7] Bouet, M., Leguay, J., Combe, T., Conan, V. "Cost-based placement of vDPI functions in NFV infrastructures", International Journal of Network Management, 25:490–506, (2015).

[8] Internet: Software-Defined Networking (SDN) Definition - Open Networking Foundation, https://www.opennetworking.org/sdn-definition/ (accessed January 18, 2019).

[9] Open Networking Foundation "SDN Architecture", ONF TR-502, June, (2014).

[10] Jarraya, Y., Madi, T., Debbabi, M. "A survey and a layered taxonomy of software-defined networking", IEEE Communications Surveys & Tutorials, 16:1955–1980, (2014).

[11] Kim, H., Feamster, N. "Improving network management with software defined networking", IEEE Communications Magazine, 51:114–119, (2013).

[12] Mckeown, N., Anderson, T., Peterson, L., Rexford, J., Shenker, S., Louis, S. " OpenFlow: enabling innovation in campus networks", ACM SIGCOMM Computer Communication Review, 38(2): 69-74, (2008).

[13] Sezer, S., Scott-Hayward, S., Chouhan, P., Fraser, B., Lake, D., Finnegan, J., Viljoen, N., Miller, M., Rao, N. "Are we ready for SDN? Implementation challenges for software-defined networks", IEEE Communications Magazine, 51:36–43, (2013).

[14] Hu, F., Hao, Q., Bao, K. "A survey on software-defined network and OpenFlow: From concept to implementation", IEEE Communications Surveys & Tutorials, 16:2181–2206, (2014).

[15] Raza, S., Lenrow, D., "Northbound Interfaces" Open Networking Foundation North Bound Interface Working Group (NBI-WG) Charter White Paper, 1-8, (2013).

[16] Ahmad, I., Namal, S., Ylianttila, M., Gurtov, A. "Security in Software Defined Networks: A Survey", IEEE Communications Surveys & Tutorials, 17:2317–2346, (2015).

[17] Chiosi, M., Clarke, D., Willis, P., Reid, A., Feger, J., Bugenhagen, M., Khan, W., Fargano, M., Cui, C., Deng, H., Telekom, D., Michel, U. "NFV", White-Paper 2, Citeseer., (2012).

[18] NFV/SDN combination framework for provisioning and managing virtual firewalls", IEEE Conference on Network Function Virtualization and Software Defined Network, San Francisco, CA, USA, 107–114, (2016).

[19] Mijumbi, R., Serrat, J., Gorricho, J.L., Bouten, N., De Turck, F., Boutaba, R. "Network function virtualization: State-of-the-art and research challenges", IEEE Communications Surveys & Tutorials, 18:236–262, (2016).

[20] Matias, J., Garay, J., Toledo, N., Unzilla, J., Jacob, E. "Toward an SDN-enabled NFV architecture", IEEE Communications Magazine, 53:187–193, (2015).

[21] Jarraya, Y., Shameli-Sendi, A., Pourzandi, M., Cheriet, M. "Multistage OCDO: Scalable Security Provisioning Optimization in SDN-Based Cloud", IEEE 8th International Conference on Cloud Computing, New York City, NY, USA 572–579, (2015).

[22] Krishnaswamy, D., Kothari, R., Gabale, V. "Latency and policy aware hierarchical partitioning for NFV systems",IEEE Conference on Network Function Virtualization and Software Defined Network, San Francisco, CA, USA, 205–211, (2016).

[23] Internet: "What is an Intrusion Detection System?", http://techgenix.com/intrusion_detection_systems_ids_part_i__network_intrusions_attack_symptoms_ids_tasks_and_ids_architecture/ (accessed January 18, 2019).

[24] Xiong, Z. "An SDN-based IPS Development Framework in Cloud Networking Environment", (2014).

[25] Ballard, J.R., Rae, I., Akella, A. "Extensible and scalable network monitoring using opensafe", Internet Network. Management Workshop/Workshop on Research and Enterprise Networking, San Jose, CA, USA, (2008).

[26] Van Adrichem, N.L.M., Doerr, C., Kuipers, F.A., "OpenNetMon: Network monitoring in OpenFlow software-defined networks", IEEE Network Operations and Management Symposium, Krakow, Poland, 1–8, (2014).

[27] Tang, T.A., Mhamdi, L., McLernon, D., Zaidi, S.A.R., Ghogho, M. "Deep Recurrent Neural Network for Intrusion Detection in SDN-based Networks", 4th IEEE Conference on Network Softwarization and Workshops, Montreal, QC, Canada, 202-206, (2018).

[28] Ceron, J.M., Margi, C.B., Granville, L.Z., "MARS: "An SDN-based malware analysis solution", IEEE Symposium on Computers and Communication, Messina, Italy, 525–530, (2016).

[29] Carl, G., Kesidis, G., Brooks, R.R., Rai, S. "Denial-of-service attack-detection techniques", IEEE Internet Computing, 10:82–89, (2006).

[30] Braga, R., Mota, E., Passito, A. "Lightweight DDoS flooding attack detection using NOX/OpenFlow", IEEE 35th Conference on Local Computer Networks, Denver, CO, USA 408–415, (2010).

[31] Bouet, M., Leguay, J., Conan, V. "Cost-based placement of virtualized deep packet inspection functions in SDN", IEEE Military Communications Conference, San Diego, California, USA, 992–997, (2013).

[32] Internet: "The Role of DPI in an SDN World",http://niwiit.org/wp-content/uploads/lteasia13/6841-Heavy_Reading-Qosmos_DPI-SDN-WP_Dec-2012.pdf (accessed January 18, 2019).

[33] Bremler-barr, A., Hay, D., Koral, Y. "Deep Packet Inspection as a Service", 10th ACM International on Conference on emerging Networking Experiments and Technologies, Sydney, Australia, 271–282, (2014).

[34] Hu, H., Han, W., Ahn, G.-J., Zhao, Z. "FlowGuard: Building Robust Firewalls for Software-Defined Networks", Third workshop on Hot topics in software defined networking, Chicago, Illinois, USA, 97–102. (2014).

[35] Suh, M., Park, S.H., Lee, B., Yang, S. "Building firewall over the software-defined network controller", IEEE International Conference on Advanced Communication Technology, Pyeongchang, South Korea, 744–748, (2014).

[36] François, J., Dolberg, L., Festor, O., Engel, T. "Network Security through Software Defined Networking : a Survey" (2014).

[37] Hu, H., Ahn, G., Han, W., Zhao, Z. "Towards a Reliable SDN Firewall", 2-4, (2014).

[38] Zhang , L., Shou, G., Hu, Y., Guo, Z. "Deployment of Intrusion Prevention System based on Software Defined Networking", IEEE International Conference on Communication Technology, Guilin, China, 26–31, (2013).

[39] Kwon, J., Seo, D., Kwon, M., Lee, H., Perrig, A., Kim, H. "An incrementally deployable anti-spoofing mechanism for software-defined networks", Computer Communications, 64:1–20, (2015).

[40] Yao, G., Bi, J., Feng, T., Xiao, P., Zhou, D. "Performing software defined route-based IP spoofing filtering with SEFA", 23th International Conference on Computer Communication and Networks, Shanghai, China, 1–8, (2014).

[41] Cox, J.H., Clark, R.J., Owen, H.L., "Leveraging SDN for ARP security", IEEE SOUTHEASTCON, Norfolk VA, 1–8, (2016).

[42] Alharbi, T., Durando, D., Pakzad, F., Portmann, M. "Securing ARP in Software Defined Networks", IEEE 41th Conference on Local Computer Networks, Dubai, UAE, 523–526, (2016).

[43] Ali, S.T., Sivaraman, V., Radford, A., Jha, S. "A Survey of Securing Networks Using Software Defined Networking", IEEE Transctions on Reliability, 64:1086–1097, (2015).

[44] Shin, S., Porras, P., Yegneswaran, V., Fong, M., Gu, G., Tyson, M., Texas, A., Station, C., Park, M. "Fresco: Modular composable security services for software-defined networks", 20th Annual Network & Distributed System Security Symposium, San Diego, CA, USA, 1-16, (2013).

[45] Han, W., Zhao, Z., Doupé, A., Ahn, G.-J., "HoneyMix: Toward SDN-based Intelligent Honeynet", ACM International Workshop on Security in Software Defined Networks & Network Function Virtualization, New Orleans, Louisiana, USA, 1–6, (2016).

[46] Jafarian, J., Al-Shaer, E., Duan, Q., "OpenFlow Random Host Mutation: Transparent Moving Target Defense using Software Defined Networking" ACM Workshop on Hot topics in software defined networks, Helsinki, Finland, 127–132, (2012)

[47] Jafarian, J.H., Al-Shaer, E., Duan, Q. "An Effective Address Mutation Approach for Disrupting Reconnaissance Attacks", IEEE Transactions on Information Forensics and Security, 10:2562–2577, (2015).

[48] Achleitner, S., La Porta, T., McDaniel, P., Sugrim, S., Krishnamurthy, S. V., Chadha, R. "Cyber Deception: Virtual Networks to Defend Insider Reconnaissance", 8th ACM CCS international workshop on managing insider security threats, Hofburg Palace, Vienna, Austria, 57–68, (2016).

[49] Chiang, C.Y.J., Gottlieb, Y.M., Sugrim, S.J., Chadha, R., Serban, C., Poylisher, A., Marvel, L.M., Santos, J. "ACyDS: An adaptive cyber deception system", IEEE Military Communications Conference, Baltimore Maryland, USA, 800–805, (2016).

[50] Robertson, S., Alexander, S., Micallef, J., Pucci, J., Tanis, J., Macera, A. "CINDAM: Customized information networks for deception and attack mitigation", IEEE International Conference on Self-Adaptive and Self-Organizing Systems Workshops 114–119, Cambridge, MA, USA, (2015).

[51] Wang, J., Wen, R., Li, J., Yan, F., Zhao, B., Yu, F. "Detecting and Mitigating Target Link-Flooding Attacks Using SDN", IEEE Transactions on Dependable and Secure Computing, 5971:1–13, (2018).

[52] Kang, M.S., Lee, S.B, Gligor, V.D. "The crossfire attack", IEEE Symposium on Security and Privacy, San Francisco, CA, 127–141, (2013).

[53] Aydeger, A., Saputro, N., Akkaya, K., Rahman, M. "Mitigating Crossfire Attacks Using SDN-Based Moving Target Defense", IEEE 41st Conference on Local Computer Networks, 627–630,Dubai, UAE, (2016).

[54] Ma, W., Jonathan, B., Pan, Z., Pan, D., Pissinou, N. "SDN-Based Traffic Aware Placement of NFV Middleboxes", IEEE Transactions on Network and Service Management, 14:528–542, (2017).

[55] Li, X., Qian, C. "The virtual network function placement problem", IEEE International Conference on Computer Communications Workshops, Hong Kong, China, 69–70, (2015).

[56] Li, X., Qian, C. "A survey of network function placement", 13th IEEE Annual Consumer Communications & Networking Conference, Las Vegas, NV, USA, 948–953, (2016).

[57] Bari, M.F., Chowdhury, S.R., Ahmed, R., Boutaba, R. "On orchestrating virtual network functions", 11th International Conference on Network and Service Management, Barcelona, Spain, 50–56, (2015).

[58] Addis, B., Belabed, D., Bouet, M., Secci, S. "Virtual network functions placement and routing optimization", IEEE 4th International Conference on Cloud Networking, Niagara Falls, Canada, 171–177, (2015).

[59] Luizelli, M.C., Bays, L.R., Buriol, L.S., Barcellos, M.P., Gaspary, L.P. "Piecing together the NFV provisioning puzzle: Efficient placement and chaining of virtual network functions", IFIP/IEEE

International Symposium on Integrated Network Management, Ottawa, ON, Canada, 98–106, (2015).

[60] Rawat, D.B., Reddy, S.R. "Software Defined Networking Architecture, Security and Energy Efficiency: A Survey", IEEE Communications Surveys & Tutorials, 19:325–346, (2017).

[61] Machuca, C.M. "Expenditures study for network operators", International Conference on Transparent Optical Networks, Nottingham, UK, 18–22, (2006).

[62] Murukan, P., Jamaluddine, D. "A Cost-based Placement Algorithm for Multiple Virtual Security Appliances in Cloud using SDN: MO-UFLP (Multi-Ordered Uncapacitated Facility Location Problem)", http://arxiv.org/abs/1602.08155, 1–14, (2016).

[63] Demirci, S., Demirci, M., Sagiroglu, S. "Optimal Placement of Virtual Security Functions to Minimize Energy Consumption", International Symposium on Networks, Computers, and Communications, Rome, Italy, 1–6, (2018).

[64] Shameli-Sendi, A., Jarraya, Y., Fekih-Ahmed, M., Pourzandi, M., Talhi, C., Cheriet, M. "Optimal placement of sequentially ordered virtual security appliances in the cloud", IFIP/IEEE International Symposium on Integrated Network Management, Ottawa, ON, Canada, 818–821, (2015).

[65] Doriguzzi-Corin, R., Scott-Hayward, S., Siracusa, D., Savi, M., Salvadori, E. "Dynamic and Application-Aware Provisioning of Chained Virtual Security Network Functions", https://arxiv.org/pdf/1901.01704.pdf, (2019).

[66] Shameli Sendi, A., Jarraya, Y., Pourzandi, M., Cheriet, M. "Efficient Provisioning of Security Service Function Chaining Using Network Security Defense Patterns", IEEE Transactions on Services Computing, 1–1, (2016).

[67] Internet: GLPK - GNU Project - Free Software Foundation (FSF), https://www.gnu.org/software/glpk/ (accessed January 18, 2019).

[68] Internet: JUNG - Java Universal Network/Graph Framework, http://jung.sourceforge.net/ (accessed January 31, 2019).

[69] Internet: GÉANT Network, https://geant3.archive.geant.org/Network/pages/home.aspx (accessed January 18, 2019).

[70] Internet: Build the future of Open Infrastructure., https://www.openstack.org/ (accessed January 18, 2019).

[71] Internet: OpenDaylight, https://www.opendaylight.org/ (accessed January 31, 2019).

[72] Internet: JGAP download | SourceForge.net, https://sourceforge.net/projects/jgap/ (accessed January 31, 2019).

[73] Internet: CPLEX Optimizer | IBM, https://www.ibm.com/analytics/cplex-optimizer (accessed January 18, 2019).

[74] Internet: Internet2, https://www.internet2.edu/ (accessed January 18, 2019).

[75] Internet: Gurobi Optimization - The State-of-the-Art Mathematical Programming Solver, http://www.gurobi.com/ (accessed January 18, 2019).

[76] Internet: Welcome to Python.org, https://www.python.org/ (accessed January 31, 2019).

[77] Internet: The GARR Network, https://www.thegarrnetwork.org/ (accessed January 31, 2019).

[78] Internet: Interface to Network Security Functions (i2nsf), https://datatracker.ietf.org/wg/i2nsf/about/ (accessed January 31, 2019).

[79] Farrel, A. "Recent Developments in Service Function Chaining (SFC) and Network Slicing in Backhaul and Metro Networks in Support of 5G", I 20th International Conference on Transparent Optical Networks, Bucharest, Romania, 1–4, (2018).

[80] Lee, W., Choi, Y., Kim, N., "Study on Virtual Service Chain for Secure Software-Defined Networking", Advanced Science and Technology Letters, 29:177–180, (2013).

[81] Chantre, H.D., da Fonseca, N.L.S. "Multi-Objective Optimization for Edge Device Placement and Reliable Broadcasting in 5G NFV-Based Small Cell Networks", IEEE Journal on Selected Areas in Communications, 36:2304–2317, (2018).

[82] Wang, X., Wu, C., Le, F., Liu, A., Li, Z., Lau, F. "Online VNF scaling in datacenters", IEEE 9th International Conference on Cloud Computing, San Francisco, CA, USA, 140–147, (2017).

[83] Alleg, A., Kouah, R., Moussaoui, S., Ahmed, T. "Virtual Network Functions Placement and Chaining for real-time applications," IEEE 22th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks, Lund, Sweden, 1-6, (2017).

[84] Cho, D., Taheri, J., Zomaya, A.Y., Bouvry, P. "Real-Time Virtual Network Function (VNF) Migration toward Low Network Latency in Cloud Environments", IEEE 10th International Conference on Cloud Computing, Honolulu, CA, USA, 798–801, (2017).

[85] Kim, S., Han, Y., Park, S. "An energy-Aware service function chaining & reconfiguration algorithm in NFV", IEEE International Workshop on Foundations and Applications of Self Systems, Augsburg, Germany, 54–59, (2016).