A survey of link flooding attacks in software defined network ecosystems

A Survey of Link Flooding Attacks in Software Defined Network Ecosystems

Raihan ur Rasool[a,*], Hua Wang[a], Usman Ashraf[b], Khandakar Ahmed[a], Zahid Anwar[c,d], Wajid Rafique[e]

[a] *Victoria University, Melbourne, Australia*
[b] *King Faisal University, Riyadh, KSA*
[c] *National University of Sciences and Technology, Islamabad, Pakistan*
[d] *Fontbonne University, Saint Louis, MO, USA*
[e] *Department of Computer Science and Technology, Nanjing University, P. R. China*

Abstract

Link Flooding Attacks (LFA) are a devastating type of stealthy denial of service attack that congests critical network links and can completely isolate the victim's network. In this work, we present a systematic survey of LFA patterns on all the layers of the Software Defined Network (SDN) ecosystem, along with a comparative analysis of mitigation techniques. The paper starts by examining different LFA types, techniques, and behaviors in wired and wireless SDNs. Next, an in-depth analysis of mitigation techniques is presented along with their suitability for each of the SDN variants. Subsequently, the significance of a pattern matching and machine learning-based detection and mitigation approaches as a defense against these attacks is highlighted. The goal is to provide a comprehensive survey to aid the research community in the design of viable solutions to LFA in SDN, that remain effective at different stages of the attack. The paper also contributes by discussing the vulnerabilities of in-band SDNs against LFA when the interface of the data/control plane is attacked by saturating shared strategic links through stealth flows.

Keywords:

[*] Corresponding author

Email addresses: raihan.rasoo@live.vu.edu.au (Raihan ur Rasool), hua.wang@vu.edu.au (Hua Wang), uashraf@kfu.edu.sa (Usman Ashraf), khandakar.ahmed@vu.edu.au (Khandakar Ahmed), zanwar@fontbonne.edu (Zahid Anwar), rafiqwajid@smail.nju.edu.cn (Wajid Rafique)

Preprint submitted to Journal of LaTeX Templates *June 3, 2020*

Link flooding attacks, SDN Attacks, SDN Security, SDWMN, SDMN.

1. Introduction

Link Flooding Attacks (LFA) are classified as one of the most lethal attacks targeting modern-day networks. These attacks can cause a denial of service by choking important links and ultimately bringing the entire network down [1]. Depending on the particular technique and methodology, several LFAs have broadly been described in literature [2] such as crossfire, coremelt, and spamhaus. Among these, the crossfire LFA [3] is harder to detect as it isolates the target by flooding the links around it with low rate legitimate traffic.

SDN is the main driver of enterprise networking and the cloud era, making it a prime target for different attacks. SDN's vulnerability to LFA increases due to the presence of a centralized controller responsible for managing the network. This paper focuses on LFA concerning SDN on all three layers (also called planes) of the SDN architecture i.e. application, data, and control as shown in Fig. 1. An adversary has been shown to be able to manipulate bots to generate attack flows to implement LFA, causing overwhelming damage to the SDN layers.

Application layer: The application layer contains the services and applications that request network functions from the data and control planes. LFA targeting this plane can cause applications to crash, disrupting the normal flow of SDN. Network management applications are a critical component of the application plane and network security is a prime concern. LFA can target the entire SDN ecosystem, and any successful attack can easily destroy network operations and disrupt services [4], [5], [6], [7].

Control Layer: is a central unit in SDN responsible for successful packet delivery from the source to the destination [8, 9, 10]. The SDN controller employs different interfaces to communicate with other layers and network elements including the east, west, north and, southbound APIs [8]. The controller communicates with the infrastructure layer and network devices by utilizing the

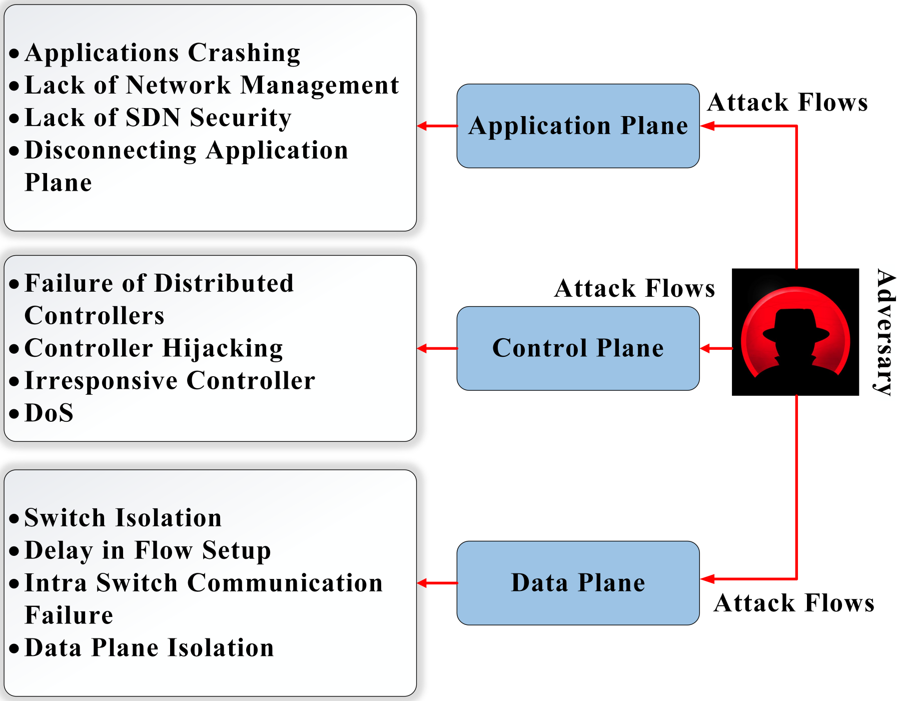

Figure 1: Broad categories of effects of LFA on SDN planes

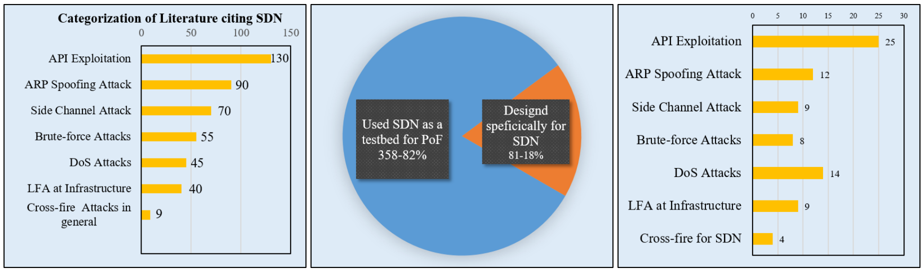

Figure 2: Survey of 439 research papers citing SDN and further narrowing down the literature.

southbound API. The northbound API is used to connect the controller with network applications [11]. Unlike the southbound API, there is no standardized protocol governing the northbound API thereby exposing it to numerous security threats [12, 13, 14]. The east and westbound APIs are utilized for managing multiple controllers that are distributed. Such a configuration is used to manage different portions of a network [15] to avoid a single point of failure or bottleneck [16, 17]. LFA on the control plane can crash and disconnect distributed controllers from other planes resulting in network outages [18]. Additionally, flooding can cause DoS on the control plane, leaving controllers unable to fulfill legitimate requests [19]. Some previous works address DoS [20, 21, 22] and exploit fault-tolerant properties in the controller [23, 24].

Data Layer: Initial attacks on SDN focused on attacks of the data plane that exploited a vulnerability in the flow tables of SDN switches whereby fake flows were inserted causing depletion of memory and causing overload [25, 26, 27]. In [28] Sood et al. evaluated the performance of SDN switches that processed incoming packets without interaction with the controller. Since data plane attacks[4] directly target the core SDN hardware, they have the potential to cause severe consequences to the network. Under LFA, these hardware devices can get disconnected making the network services irresponsive [29]. This attack involves flooding the infrastructure layer causing communication delays and performance degradation. Another serious attack on the data plane involves disconnecting switches from each other, which results in loss of flow rules and synchronization issues between coordinating switches [30]. LFA can also interfere with the flow rules installation process in SDN switches which causes a delay or even disconnection of the rule installation service, ultimately slowing the entire network. LFA can isolate the data plane [4, 31] from the SDN bringing the whole network down.

Apart from the above discussion, we have performed an extensive literature survey on LFA and found the following characteristics.

- LFAs can segment off target links and can disconnect networked regions over the Internet.

- Defense against LFA is more difficult than mitigating DDoS attacks [32, 33].

- The attack vector is indirect and never directly targets the end servers to avoid detection by intrusion-detection systems and firewalls [3].

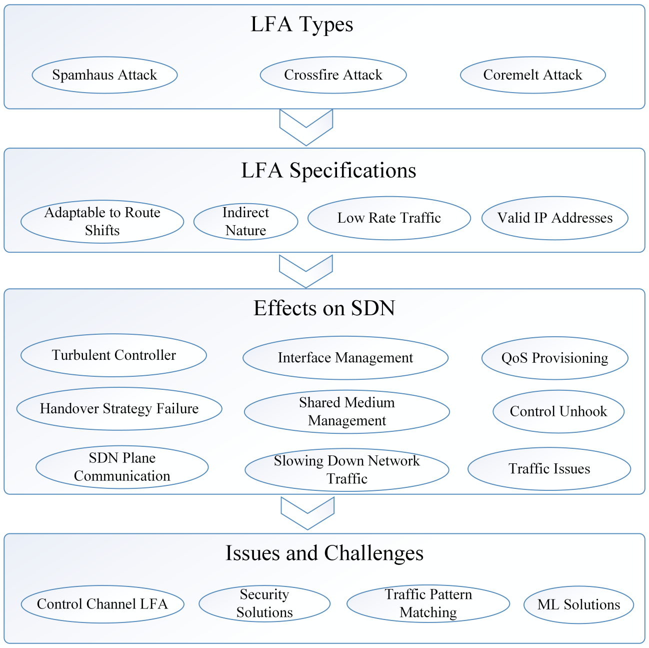

Figure 3: Structure of the paper.

- Attackers use legitimate traffic to flood specific links which cause increased false-positive rate when flow drop techniques are used for mitigation.

Due to the above characteristics, and the stealthy nature of LFA it is important to determine effective mitigation strategies. Several varying techniques [1, 34, 35, 36] have been proposed in the literature that can be broadly categorized into traffic engineering [1, 37, 38], link monitoring [2, 38, 39] and SDN principle-based approaches [32, 40, 41]. In [30] Niyaz et al. have elaborated on the different attacks on SDN, and their effect on web services. Fig. 2 attempts to categorize different attacks in the published research citing SDN. Despite its importance, LFA has received little attention. Fig. 2 highlights that only 18% of these works were actually geared towards SDN, with the majority mainly employing SDN for a proof of concept. The bar chart in Fig. 2 clearly shows that only 9 research papers have so far considered LFA problem in SDN and that too mostly focusing on the data plane only.

Table 1: A comparison of our study with the previous surveys in securing SDNs.

References	Application Plane	Control Plane	Data Plane	Control Channel	REST API	LFA	DDoS	SDN Variants
Security Taxonomies [63]	✓	✓	✓	✓			✓	
Flow-based DDoS[64]	✓	✓	✓	✓				
Control Plane Security[65]		✓					✓	✓
SDN Security[66]	✓	✓	✓	✓			✓	
SDN Architecture Security[67]	✓	✓	✓					
DDoS in SDN[68]	✓	✓	✓	✓			✓	
Stateful SDN[69]	✓	✓	✓	✓				
Our Survey	✓	✓	✓	✓	✓	✓	✓	✓

A growing trend in the adoption of and dependence on the big data and social networks can be observed in all aspects of life [42] and nearly in every field of scientific research that includes education [43], business management [44], health-care [45], aerospace [46] and social works [47]. Preserving privacy [48, 49, 50, 51, 52, 53, 54, 55], ensuring information security [56, 57] as well as safeguarding the internet for effective provisioning of network services [58] is of prime importance in the globally connected world [59]. SDN is a key enabler in IoT network management and Software Defined Internet of Things (SDIoT) infrastructure provisioning [60, 61]. Therefore, it is crucial to safeguard the networks which are mostly software-defined, against vulnerabilities and attacks such as LFAs for seamless provision of the online services [62]. In this connection, an in-depth survey of LFA and defense mechanisms in the SDN ecosystem including wired and wireless environments is needed.

Table 1 discusses a comparison of this study with the already available surveys in the SDN security domain. This table contains surveys that address the security issues at different planes, interfaces and SDN variants. It can be observed that there is a lack of studies available that address all the SDN planes, interfaces, variants and attack categories. A systematic study of LFA is needed to understand the various aspects associated with this critical network security issue and to develop sustainable solutions. The present work is an extension of our preliminary research in the classification of LFA mitigation techniques

Table 2: Comparison with existing Surveys.

Survey Title	**Main Idea**	**Limitation**	**Year**
A Survey and a Layered Taxonomy of SDN [8]	Survey vulnerabilities of SDN applications, platforms, the OpenFlow and the SDN controller.	DoS attacks considered are limited to application-layer attacks and flooding of flow requests to overwhelm the controller.	2014
A Survey of Security in Software Defined Networks [66]	Discusses DoS attacks on the communication channel between the network element and the controller when not operating in the same trust domains.	Does not address the inherent limitations of LFA. Solutions covered focus on the need to establish trust between SDN elements using authorization mechanisms. These are limited.	2016
SDN Architecture, Security and Energy Efficiency: A Survey [71]	Present seven different SDN threat vectors with the first two focused on DoS.	Solutions discussed are focused on programmable SDN to reduce energy consumption by network infrastructure using sleep-awake mechanisms and traffic management while providing network security.	2017

[70]. In this research, we initially present a comprehensive survey of LFA and mitigation techniques in relation to all layers of wired and wireless (mesh) SDN. Subsequently, a comparative analysis of mitigation techniques is presented with pointers to suitability for each of a given SDN type. We categorize and classify LFAs and present the countermeasures from the literature keeping in view the prevention and detection mechanisms suitable for each type. Finally, we highlight some important issues which have been overlooked in the literature and provide directions for future research in this area.

There exist prior surveys providing extensive coverage of research in network attacks involving SDN. Surveys include but are not limited to LFA, topology discovery attacks, malicious switches, compromised controllers, attacks on SDN interfaces, security applications, vulnerabilities of certain SDN platforms, the security of OpenFlow and even the use of SDN as a security solution to traditional attacks. However, all of these surveys are limited in their coverage of vulnerabilities of SDN to LFA and its mitigation. A comparison of noteworthy surveys and their limitations are provided in Table 2.

A systematic study of LFA is needed to understand the various aspects associated with this critical network security issue and to develop sustainable solutions. The present work is an extension of our preliminary research in the classification of LFA mitigation techniques [70]. In this research, we initially present a comprehensive survey of LFA and mitigation techniques concerning all layers of wired and wireless (mesh) SDN. Subsequently, a comparative analysis of mitigation techniques is presented with pointers to suitability for each of a given SDN type. We categorize and classify LFAs and present the countermeasures from the literature keeping in view the prevention and detection mechanisms suitable for each type. Finally, we highlight some important issues which have been overlooked in the literature and provide directions for future research in this area. Following are some of the main contributions of this work:

- We provide comprehensive coverage and comparative analysis of the types of LFA targeting SDN networks including deployments in both wired and wireless ecosystems.
- LFA Mitigation solutions are surveyed and ranked based on a novel set of quality metrics proposed in this paper.
- We provide guidelines to follow when implementing mitigation techniques that would allow them to be more proactive and robust.
- We highlight that previous solutions have focused exclusively on either data or control planes and we explore the impact of LFAs when strategic links are choked through the data plane in order to cause a denial of service for the control plane.

An overall structure of the paper has been provided in Fig. 3 and a list of acronyms used in the paper is given in Table 3. The rest of the paper is organized as follows: Section II provides insights into LFA types and threat sources in SDNs. In Section III, details the different SDN types and discusses different layers prone to attacks. Section IV presents link vulnerabilities in SDN

Table 3: List of acronyms.

Symbol	Description	Symbol	Description
LFA	Link Flooding Attack	ASes	Autonomous Systems
SDN	Software Defined Network	FIB	Forwarding Information Base
SDWN	Software Defined Wireless Network	LTE	Long Term Evolution
SDMN	Software Defined Mobile Network	SSID	Service Set Identifier
SDWMN	Software Defined Wireless Mesh Network	OLSR	Optimized Link State Routing
SDLAN	Software Defined Local Area Network	VN	Virtual Networks
IP	Internet Protocol	SVM	Support Vector Machine
ISP	Internet Service Provider	CoDeF	Collaborative Defense
DoS	Denial of Service	VPN	Virtual Private Network
LAN	Local Area Network	DNS	Domain Name System
Wi-Fi	Wireless Fidelity	DHCP	Domain Host Configuration Protocol
IXP	Internet Exchange Point	WMN	Wireless Mesh Network
API	Application Programming Interface	MTD	Moving Target Defense

highlighting the most LFA prone components of SDN. Section V exhaustively reviews LFA mitigation techniques from the literature and provides a classification of the defense mechanisms based on a certain criterion. Section VI focuses on future directions and current issues, while Section VII concludes the paper.

2. Link Flooding Attacks

This section details different LFAs including the crossfire [3], spamhaus [72], and the coremelt attacks [73]. Furthermore, we present a comparison of these attacks according to certain parameters that are critical in nature for all LFA types to show why crossfire attacks are the most lethal as compared to all other attacks. During recent years, attacks on the network infrastructure have been increased tremendously [74, 75, 76].

LFA is a category of DDoS attacks [33, 77], which has the potential to disconnect network connections of a target area. DDoS attacks are often carried out by a huge number of superfluous requests to a target resource or a machine that causes disruptions in the legitimate service delivery. DDoS attackers use spoofed IP addresses, which makes it harder to detect the actual source of an attack. Due to extensive research in securing current networks, DDoS adversaries have evolved a novel attack strategy, which involves stealth flows to attack the current networks. This DDoS strategy known as LFA is challenging to detect and mitigate as the adversaries exploit legitimate flows, whereas the attack rate never exceeds the allocated data rate. Hence, the attackers are able to cautiously disconnect the victim from the rest of the network.

In LFA the attacker analyzes the network to identify the connecting links to the target area. A use case example of LFA is given in Fig. 5, where bots are sending flood traffic on the target link, which is connected to the target server. Using this setup the adversary can potentially disconnect the target server from the network without directly attacking the target server. We explain different types of LFA in the following subsections.

2.1. Coremelt Attack

In this attack, a set of compromised systems is used by the attacker to send packets to each other in order to flood a specific link in a network [73] as shown in Fig. 4. While compromised systems exchange packets with each other, the adversary can evade defence mechanisms based on flow filtering, because the attackers use protocol conforming traffic [78]. The attacker can effectively shut down a backbone when the attack hosts are dispersed across different networks.

The adversaries use the following steps to launch an attack.

1. Identify a target link in the network.

2. Distinguish the set of attacking host pairs that can generate enough traffic to overload the desired link.

3. Exchange packets among the attack hosts to congest the target link.

Because of less resource requirement and simplified attack implementation, the coremelt attack can be manipulated easily to attack current networks including SDN.

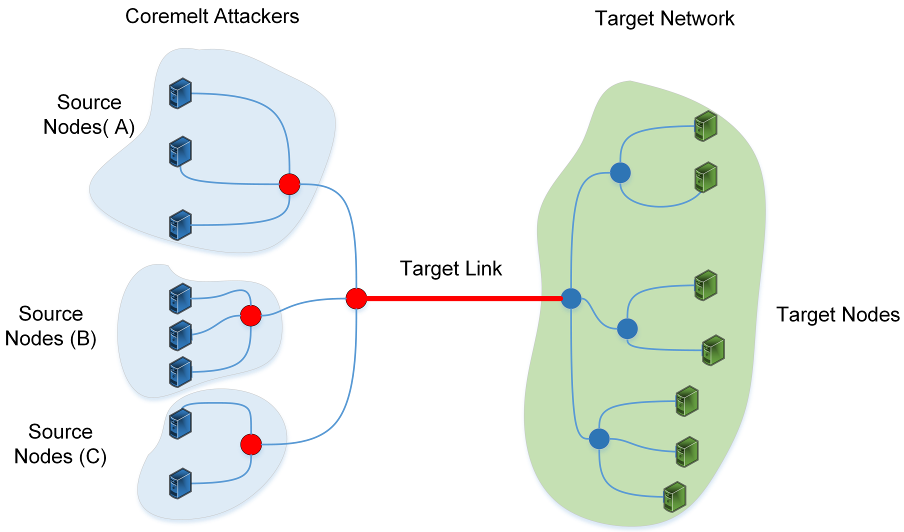

Figure 4: The setup of a coremelt attack, pairs of computers are carefully selected to attack the target link. The Figure at [73] has been redrawn and expanded.

2.2. Crossfire Attacks

The crossfire attack is more sophisticated and stealthy as compared to the coremelt attack, because it is adaptable to route shifts and avoids triggering alarms by changing the target links after a specified time interval set by the
190 attacker. In contrast to the coremelt attack, the efficiency does not depend upon the geographic distribution of the bots. ISP collaboration is an essential requirement for an effective defense using modern security tools. Therefore, all these factors need to be a consideration when developing a mitigation solution.

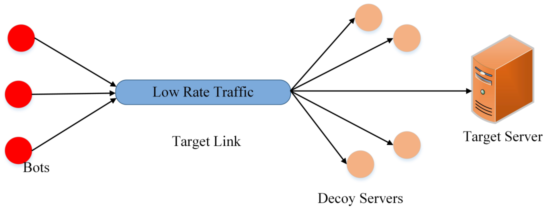

Figure 5: The Setup of a Crossfire Attack demonstrating the target link between bots and decoy servers [3].

The crossfire attack is carried out using bots as shown in Fig. 5. It can be observed that the bots send traffic to decoy servers which block the traffic to the target. The attack rate and the number of bot-decoy pairs are carefully selected to effectively congest the link. Sending low bit-rate traffic to decoys chokes the links connecting the target server with the rest of the network. Detecting crossfire LFA is complicated because the target server never receives flood traffic directly [3]. Similarly, bots use legitimate IP addresses, hence, it is difficult to identify attack sources [3, 79, 80]. The adversary starts by constructing the network profile by sending traceroute packets to identify target server and critical links to attack. It then selects the decoy servers and calculates a sufficient number of bot-decoy pairs to launch the flooding operation. Finally, it sends low rate flows from bots to decoys, so all paths to the target experience flood traffic, as a result, the legitimate traffic is blocked. Crossfire attack has the potential to attack modern SDN in a variety of ways by leveraging its indirect attack strategy. One such scenario can be control channel LFA.

2.3. Spamhaus Attack

A major Internet-scale LFA attack was performed on the spamhaus server, an organization that provides spam filtering services to subscribers. In this particular case, the attack was launched on a number of important links to vital Internet exchange points (IXPs) in Europe and Asia to deny specific cloud services. The intensity of the attack was so severe that it constitutes an attack type in LFAs. The attack started by directly targeting specific servers and with time, the attack evolved by flooding network links on multiple IXPs [72].

The attack setup is shown in Fig. 6. Initially, attackers used open resolvers to send service requests to the spamhaus server, which was unable to service this massive amount of requests and became unresponsive. Later, spamhaus used the services of CloudFare [81] to cater to the enormous attack traffic which provided the spamhaus server the ability to fulfill requests. When attackers were unable to bring spamhaus down, they attacked the regional exchange links in Europe and Asia that allowed CloudFlare and a large internet service provider to pass traffic to each other. This evolved strategy was successful in bringing the spamhaus services down.

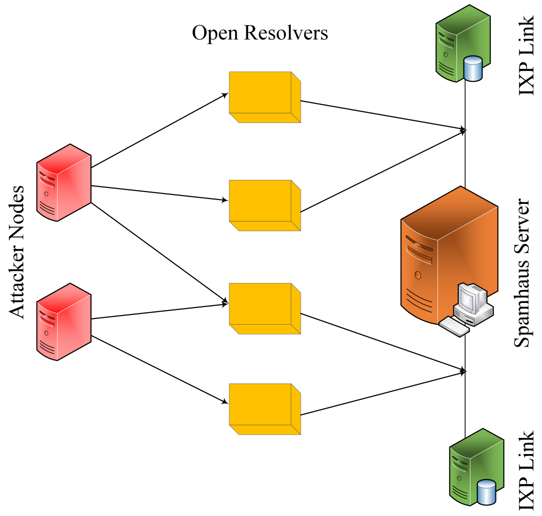

Figure 6: Open resolvers sending service requests to IXPs to flood links leading to spamhaus server.

2.4. Taxonomy of Attacks in SDNs

In Fig. 7, a comparison of the three attacks according to specific attack parameters has been carried out. A taxonomy of attacks has been previously conducted which outlines the attacks on SDN planes; however, it lacks in discussing the potential impact of LFAs on SDN planes and interfaces [82]. The coremelt attack uses a pair of bots which coordinate with each other to accomplish the attack, while crossfire attacks use bots and public servers' collaboration for attack implementation.

In the crossfire attack, the targets are network links around a certain geographical region which belong to several ASes and Internet Service Providers ISPs. Bots are independently distributed around the target server in crossfire attack while in coremelt attack, bots are placed on each side of the target link which needs to be flooded i.e. they are not distributed independently in the coremelt attack. Flows in both crossfire and coremelt attack are legitimate.

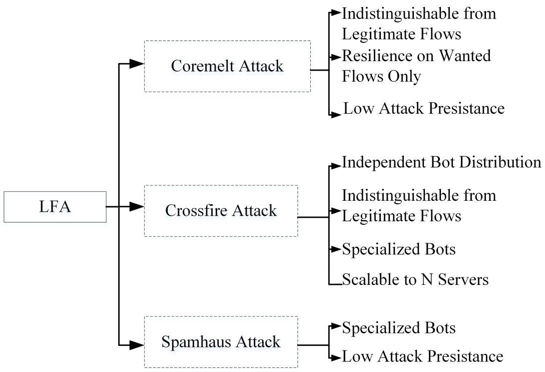

Figure 7: LFA taxonomy.

Spamhaus used open resolvers which sent illegitimate flood traffic to the links. Attack persistence is high compared to crossfire and coremelt attacks.

Fig. 7 illustrates that the crossfire attacks are the most lethal attacks because of their low detection rate. The crossfire attack has the ability to disable network links by flooding them with attack traffic. Multiple factors contribute to the severity of this attack.

- First, it uses legitimate IP traffic, rather than using spoofed IP addresses which makes it hard to filter.
- Furthermore, it sends legitimate packets to publicly accessible servers (decoys). So these packets keep on flowing without any interruption, flooding the link between these servers.
- Finally, it transmits low bandwidth flows from each bot individually and these legitimate flows then cumulatively flood certain links in the network without being detected.

Table 4, contains different types of attacks on SDN planes and interfaces along with their possible solutions. As it can be observed that the impact of LFA on the control plane to the data plane interface has not yet been adequately explored in the literature. In this regard, a viable solution against LFA on control plane to data plane is necessary.

Table 4: Attack Types on SDN Planes/ Interfaces And Solutions.

Target Plane/Interface	**Attack Type**	**Current Solutions**
Data Plane	DoS	CONA [83], SDN-Guard [21], FloodDefender [84]
	Anomalous Switches	SDNsec[85], OFGUARD [86], FortNox [87]
	Anomalous hosts	OFGuard[86], FlowVisor [88]
Control Plane	DoS	Lightweight DDoS[89], LineSwitch[90], SDNShield [10]
	Compromised controller	HyperFlow [91], Fleet [92], OrchSec[93], FleXam[94]
	Malicious modules	Avant Guard [95]
	Controller disconnection	DRS [96]
Application Plane	SDN security	PremOFF [97], SDNRootkits [98], OFX [99]
	Information leakage	Proactive Strategic and Randomization [99]
Control Plane-Data Plane	Malicious rules	Veriflow [100], HSCS Architecture [101]
	LFA	No prior research conducted
Control Plane-Application Plane	DoS	Multi Controller Architecture for SDN [102]

As discussed in Section 1, attacks on SDNs can be broadly classified into five different areas based on the target: data plane, control plane, application plane, control-data plane and control-application planes. Table 2 categorizes the different solutions based on the target type.

The first class of solutions focus on data-plane attacks [83, 21, 84, 85, 86, 87, 88] involving DoS [88, 21, 83], anomalous switches [85, 86, 87] and anomalous hosts [86, 88]. In [83], the data plane is flooded using a resource-exhaustive storm of requests by bots, while [18] explored attacks involving flooding the server with a large number of TCP SYN packets with different IP source addresses to emulate DDoS. In [84], authors considered attacks in which massive table-miss traffic was sent mixed with normal traffic to the OpenFlow switch to overwhelm victim switches. In [86], switches generate a large number of fake packets that trigger table-miss and send a lot of packet-in messages to controller, thus overloading the memory of the network devices as well as overwhelming the control plane bandwidth. In [87] the attack considered is mostly about leveraging flow rule conflicts to sabotage the SDN.

The second class of solutions focus on the control-plane attacks [89, 90, 10, 91, 92, 93, 94, 95, 96] specifically. In [10], an abnormally high packet_in arrival rate of a switch is considered to be an attack flow since it overwhelms the network. In [89], the attack traffic is considered to be persistent and synchronous while most flows of normal traffic are short-lived and non-synchronous. In [90], the attack is considered to be focused on saturating the control plane in order to achieve buffer saturation. In [92], the attack is considered to be due to a group of colluding malicious administrators whose goal is to reduce network availability by deliberately misconfiguring controller policies to cause undesired flow rules to be pushed to switches, thus saturating switch tables. In [93], a DNS amplification attack is considered in which attacking hosts spoof the source IP address of the victim, and Open DNS resolvers are configured to send large DNS responses to the victim. In [95], the solution focuses on attacks exploiting the lack of scalability between the data and control planes which enables an external entity to craft an inbound stream of flow requests to overwhelm communication between the two planes through a control plane saturation attack.

The third class of solutions focuses on the application plane attacks [97, 98, 99, 99]. In [98], a northbound channel overflow attack is used in which applications send excessive amounts of configuration messages to the SDN controller as well as a southbound attack in which excessive events are generated on the forwarding plane (e.g. switch migration, switch reboots) to overwhelm the resources. In [99], a TCP SYN flooding attack is considered where an attacking application floods the control plane by sending TCP SYN packets with random sources, destinations, and ports.

The fourth class of solutions focuses on attacks targeting the interface of control and data planes [100, 101]. In [100][80], the solution considers a generic category of attacks which are based on faults in the network state due to loops, sub-optimal routing, black holes and access control violations which result in services becoming unavailable. In [101], the solution focuses on attacks that leverage malicious API calls to overwhelm the network.

The fifth class of solutions [102] investigates on attacks that target the interface of the control and application planes. In [102], the attack comprises of several hosts simultaneously flooding UDP packets to other hosts using such that the top of rack (ToR) switches connected with these hosts generate excessive flow requests to the controller in order to flood the controller.

We have discussed broad categories of defense techniques on different SDN interfaces and planes. Moreover, we have elaborated on types of different LFAs and demonstrated how crossfire LFA is lethal as compared to other LFAs. In the next sections, we further discuss this attack and its implementation on all the SDN variants: SDMN, SDWN, and SDWMN, and SDLAN. We explain the effect of crossfire attack on different planes of SDN and at the end, the defense mechanism against crossfire LFA is explained.

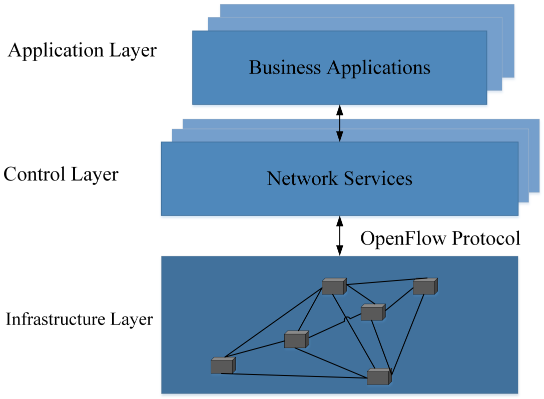

Figure 8: SDN architecture comprising of three layers [103].

3. Vulnerabilities of SDN under LFA

This section, explains the vulnerabilities of all of the variants of LFA including SDWN, SDMN, SDLAN, and SDWMN.

3.1. SDN preliminaries in an LFA Context

We start with the SDN architecture where a network behavior is controlled centrally by an API. SDN decouples the packet forwarding control structure

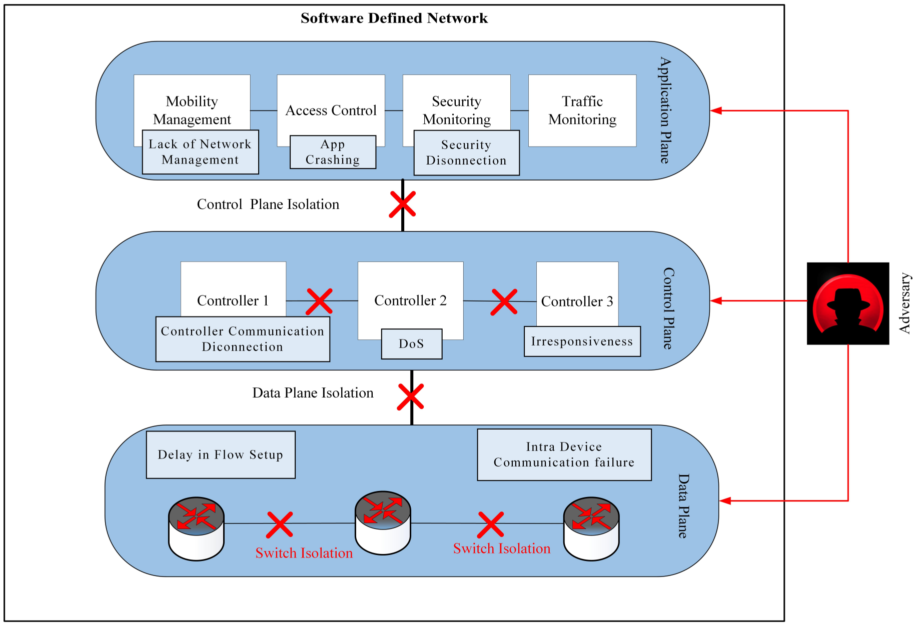

Figure 9: Effect of LFA on all SDN planes can be observed, LFA can disrupt inter-plane and intra-plane communication.

from the switching hardware [16, 104, 20, 105]. Network traffic is forwarded without changing any setting of individual switches or routers by using this approach. This network architecture consists of three planes: data, control, and the application plane as shown in Fig. 8. The data plane also commonly known as the infrastructure layer contains forwarding devices that are responsible for packets forwarding. Similarly, the control plane is responsible for managing the data plane devices in the same way the application plane contains applications to enable different SDN services [106, 107].

LFA can severely affect the SDN architecture by attacking the link between the infrastructure layer and the control layer. It can disconnect the entire infrastructure layer from the SDN. The logic to control the network infrastructure resides on the control plane of SDN. This is an area where we see a large level of commercialization whereby network vendors are offering customized solutions for the SDN controllers and frameworks. The business logic primarily resides here and is responsible for controlling network infrastructure and fetching and maintaining different types of network information such as state, topology, and statistics. As the SDN controller provides efficient network management, therefore, it must incorporate the ability to control real-world networks including firewall, switching, DHCP, security, layer2 VPN, DNS, routing, and clustering. In parallel to these three layers, there are two interfaces for communication between the different layers, i.e. northbound and southbound interfaces. Generally, the northbound interface provides the capability to communicate with the upper layer known as the application layer of SDN using REST API. Communication with the network infrastructure is controlled by southbound API using OpenFlow [104], Netconf [108], and Ovsdb [109] etc. The application layer is a relatively unexplored domain to develop as many innovative applications are possible by exploiting all the network information such as statistics, state and topology etc. Custom applications can be developed to optimize the network functionality on the application layer. These applications include network security, monitoring, configuration, troubleshooting, and automation. SDN has revolutionized the way traditional networks are managed since it dramatically reduces network operating costs by using inexpensive switches that can perform automated network functions. Different network configurations can be tested without disrupting the actual network. Due to the centralization of the FIB, optimal routes can be calculated for seamless traffic flow in the network. SDN can filter the packets as they enter the network. Data plane switches can act as firewalls and are able to redirect traffic flows to security controls at higher-level layers. In contrast to these advantages, centralized control of SDN makes it vulnerable to multiple attack types intended on all layers of SDN, like flow table overflow attacks [25, 110, 111, 112, 113, 71, 66, 114, 115, 116, 117] control channel flooding attacks, link fabrication attacks [118] and controller fail-over attacks. SDN implementation is also a very challenging task, as it requires completely changing network infrastructure.

The network service providers must alter the entire network as well as retain skilled staff and employ state-of-the-art management and diagnostic tools for SDN implementation. The consequence of LFA is shown in Fig. 9 where an adversary is sending LFA traffic on all SDN planes. The application plane contains the applications responsible for managing the entire SDN including access control, traffic and security monitoring and mobility management [119] [120]. Although the difficulty of launching LFA on the application plane is very high because it requires authentication, but still, it has a disastrous effect on the network if it gains the privileges and passes the authentication. The impact of LFA on SDN application plane is severe because it enables the applications to run smoothly, the whole network suffers if it comes under attack. LFA on application plane results in complete or partial disruption of these management modules from the whole network. Every module in the application plane performs some specific and specialized operation that is critical for the seamless operation of SDN. Flood traffic in the application plane results in network security applications to crash and hence it is a security vulnerability in SDN. LFA can affect the traffic running from the application plane to the control plane, in severe conditions the link can get disconnected. The application plane holds crucial network management policies like requesting network functions from other planes and building a high level view of the network by requesting information from the controller. LFA has the potential to create severe network management issues by attacking this plane. In the control plane, there are distributed controllers which coordinate with each other to accomplish the network control functionality. The purpose is to set up a backup mechanism, so that a secondary controller can take charge in case of the primary controller's failure. LFA can disrupt communication between the controllers that work synchronously to provide network control functionality. As the controller is the critical component of SDN, it can be attacked for DoS [121], which creates a hindrance to successful network operation. Network is centrally managed in SDN, so the communication in the control plane must be managed with care because any attack on the controller can disrupt business activity. The data plane is the main working element in SDN as it forwards the packets to its destinations. The data plane hardware components (switches, routers etc.) forward packets according to the rules provided by the control plane. On the arrival of a new packet, the switch looks for the forwarding rule for this packet and if a switch is unable to find a flow rule for a packet it queries the coordinating switches for forwarding rules. LFA can disconnect intra-switch communication by flooding the link between switches, which results in a rules updation problem and disrupts the communication between coordinating switches. The data plane can be attacked in a variety of ways using LFA, which incurs devastating effects on SDN. For example, when a flow rule is added according to the packet destination, LFA can isolate the forwarding devices, which affects the rule set up in data plane switches.

3.2. LFA in Wireless Networks

The OpenFlow protocol was initially developed for campus networks, but with further development, it has now been extended for several modes of wireless, equipping them with fine control, greater flexibility and ease of use [122, 123]. During recent years SDWN has gained tremendous popularity in terms of research [124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134]. This section presents different variants of SDWN and highlights how the LFA problem differs in the different network architecture contexts and the distinguishing characteristics of LFA mitigation solutions.

In [135], the performance enhancement of SDN is discussed in different wireless networks. Fig. 10 shows the basic structure of a SDWN. The versatile nature of devices can be from different vendors and operated by multiple operators. Managing the interoperability of the versatile nature of wireless devices is a challenging task. In the same manner, providing consistent security and seamless communication for the users of the wireless networks who hops between different networks being managed by different operators is very complex. SDN can be a straightforward solution for wireless networks because of its ability to hide complexities.

Legacy wireless networks rely on distributed schemes for network operation and provisioning, SDWN offers centralized controller for all the network devices which make it vulnerable to various LFA. In SDWN, the core characteristic of SDN, i.e. separation of control and data planes are utilized while extending it with another characteristic of a clear separation of radio access and service definition. These two principles are combined to provide robust and scalable virtualization of wireless networks [136]. Radio access is a shared resource that is responsible for delivering seamless access to mobile devices over an air medium [137, 138]. The consequences of LFA increases because of its shared nature. SDWN have unique characteristics due to the presence of wireless infrastructure, but at the same time, it also introduces new challenges of LFA for the SDN framework. The unstable nature of the wireless medium is a great security threat to SDWN and risks can be exploited by the adversaries. Due to the heterogeneous nature of wireless technology, some special problems are incorporated in SDWN like QoS provisioning, interference management, and reliable handover. In traditional networks, forwarding devices are separated so LFA to these devices can only be achieved in a cooperative manner, which makes it difficult to accomplish. SDWN relies on the concept of centralized control hence LFA can be implemented easily as compared to traditional networks. Due to the physical separation of the control and data planes, SDWN is more prone to LFA in many ways as compared to traditional networks.

LFA on forwarding devices can have devastating effects on the network because the forwarding devices play a crucial role in the network. Similarly, LFA on the control plane communication may result in slowing down the traffic over the link. LFA can attack controllers which is one of the most lethal attacks on SDWNs. A faulty controller can harm the communication of the entire network and detecting the cause of failure can be a difficult task. Lack of trust mechanisms between the controller and management applications can also prove to be lethal because a malicious application can mislead the communication flows. The controller only provides abstraction and only issues configuration commands. A user's mobility in the SDWNs poses a threat in itself because users switch between multiple networks using different technologies; it is very difficult to track anomalous activities. The negotiation process provided by multiple platforms complicates the handshaking between networks, which raises privacy issues. SDWN must observe the basic network characteristics like authenticity, confidentiality, integrity, consistency, fast responsiveness and adaptation for the smooth operation of the underlying network.

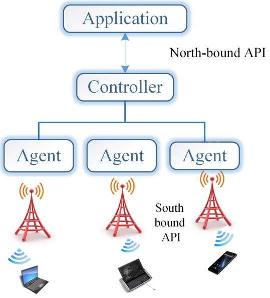

Figure 10: The controller communicates with the devices using hardware specific agents. Extended and redrawn the figure available at [139].

3.2.1. Software Defined Mobile Networks (SDMN)

SDMN form an integral part of the wireless world, and therefore warrant exploration of LFA attacks and solutions in their context. During the past years, research on SDMNs has received increased attention because of its wide adoption as a replacement for traditional mobile networks [140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 144]. The control channel is the lucrative target of the adversaries because it handles all the traffic-governing decisions [151]. In [144], Liyanage et al. have suggested for a secure channel for SDMNs based on host-identity protocol. With the rapid adoption of smartphone technologies, mobile networks have gained immense importance during the past few years. As mobile networks converge due to revolutions in wireless technologies like LTE. WiMAX and Wi-Fi are also being widely integrated into the current network infrastructure. SDMN concept is a proposed extension of SDN architecture to incorporate mobile network specific features in the software defined networking paradigm. The SDMN architecture comprises of a flow-forwarding model with a logically centralized controller. SDMN can address several limitations of mobile networks including heterogeneity, complexity, and consistency in the network efficiently. The centralized controller architecture can effectively manage spectrum usage and spectrum sharing strategies in the underlying mobile network. SDMN offers various benefits like centralized control segmentation management, network management, network control and on-demand provisioning. However, SDMN is also vulnerable to new security threats that were harder to implement on traditional mobile networks. The main security threat in SDMN is the IP level security threats, LFA, DoS and TCP reset attacks.

SDMN is prone to LFA because of their wireless nature and they can be more lethal as compared to wired medium LFAs. In SDMNs LFA can exhaust the resources in forwarding devices and controllers. Unlike SDN powered wired networks where data and control plane are on dedicated hardware, wireless medium is shared by both data and control plane. Therefore, LFA on wireless medium results in disconnecting both these planes. The forwarding devices in SDMNs communicate each other for forwarding rules sharing and updating flow tables, attack on forwarding devices results in responsiveness of the whole forwarding plane. More specifically, unlike wired networks, in which dedicated wire-lines can be used separately for the control and data planes, the same wireless communication links are typically used for both the data and control planes. This poses unique challenges as any LFA based attack will impact both the control and data planes simultaneously. In particular, launching jamming attacks combined with LFA on strategic links in the network can overwhelm the network. However, given the rather extensive coverage of the underlying wireless infrastructure in SDMNs, multiple solutions could be adopted e.g. using multiple backup nodes, load distribution across multiple forwarding devices or multi-channel availability between nodes.

3.2.2. Software Defined Wireless Mesh Networks (SDWMN)

Wireless mesh networks are one of the most important components in providing multi-hop connectivity between users and Internet gateways. They have been widely used in public internet access systems and intelligent transportation systems [152, 153, 154, 155]. The wireless community aims at providing high-speed with high-throughput connectivity for the network users by exploiting the available technologies including fourth and fifth generation mobile cellular systems, IEEE802.11 based networks, IEEE802.16 WiMAX broadband wireless networks as shown in Fig. 11. Wireless mesh networks are playing a pivotal role in providing multi-hop connectivity between end users and Internet gateways [156]. SDN enhances the performance of the SDWMN in several ways. Access points association is enhanced by using SDN, as in [157] Sood et al. propose access point association minimize location changing problem for new users in IEEE802.11 networks. A typical architecture of the wireless mesh network (WMN) consists of nodes connecting in a multi hop manner using a wireless medium. These network nodes can be either static mobile devices or static wireless routers. WMN is the main contributor in providing users with multi-hop connections and internet gateways. WMNs can be challenging to manage because of the diversity of the involved devices and dynamic network provisioning. These networks may consist of different network devices and have diverse communication capabilities. In [158] the authors use SSID for separating data networks from the control plane in a physical network. Yang et al. [155] proposed an in-band approach to developing OpenFlow SDWMN that can maintain the traffic load on the internet. Detti et al. [159] proposed the deployment of in-band style by using Optimized Link State Routing OLSR for data and control packets.

In contrast to mobile networks, mesh networks have multiple redundant links which provide resilience against link failures. Moreover, they are meant to cover a smaller geographical area. In this context, the natural structure of mesh networks provides some inherent protection against LFA in the sense that typically strategic resources in the network will have multi-link reachability, offering fortification against LFA attacks. However, at the same time, there are some open issues and challenges that need to be addressed.

A major weakness in SDWMN in the context of LFA is that the wireless medium is shared by both control and data traffic. Being a scarce resource, radio spectrum must be utilized efficiently [160]. The policy and rules for guiding and

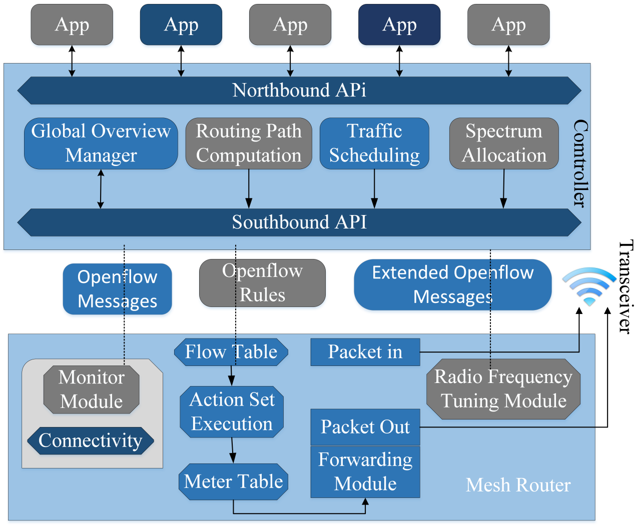

Figure 11: The architecture of SDWMN. (Modified and redrawn the figure available at [160].)

routing of data traffic is another big contribution to control traffic. An SDWMN must provide higher priority to controlling traffic as compared to the data traffic. As rules are generated to serve the traffic transmission, the control traffic can only be transmitted when control traffic is passed successfully, because control traffic is more important than the data traffic. In this regard, high priority must be given to the control traffic. LFA creates more complications in the SDWMN operation. If data plane traffic is waiting for control traffic and LFA occurs, the control traffic will not be able to reach the hardware resulting in blockage of the communication. LFA on the wireless medium also results in disconnecting both planes, which brings the whole network down. Thus, sharing the same wireless links for both control and data traffic poses vulnerability to LFA.

Another important difference between SDMN and SDWMN is that the long-wavelength cellular wireless technologies have typically longer range and provide extended wireless coverage, whereas in the case of mesh networks, the wireless range is typically limited and highly dependent on node distribution. This poses the problem that network segments with the sparse deployment of mesh nodes can be targeted by LFA since fewer wireless links will be available as a fall-back option.

In terms of solution, in addition to the LFA detection and mitigation framework developed in the paper, multi-channel and multi-radio capability of mesh networks can be leveraged to increase the link redundancy for strategic nodes in the network. Also, a higher node density can also be ensured around these strategic assets to increase the "degree of meshing"of wireless links. From a tactical perspective, perhaps the controllers in SDWMN would need to query mesh switch nodes more frequently with some triggering mechanisms in case mesh switches near strategic nodes are not accessible.

3.2.3. Software Defined Wireless Local Area Networks (SDWLAN)

The controller in WLAN can be attacked using LFA, disconnecting the communication between the controller and the wireless receivers which results in disconnecting the whole communication process. The controller in WLAN must be able to track the movement of the subscribers and this needs to be handled carefully to provide seamless communication to the users [161, 162, 163]. LFA can severely affect the handover strategy by flooding strategic links. These complexities in WLANs make it more vulnerable to LFAs.

In a study by Min et al. [164] applied SDWLAN on a university campus and published the simulation steps for its installation. So SDWLAN is now widely being adopted by many organizations because of its easy management, the flexibility of adaptation and centrally controlled nature. The operational complexity and the cost for network holders grow with the size of WLAN. The SDN architecture can also be applied in a WLAN setting to reduce operative costs and efficient network control [135]. SDWLAN has some advantages in contrast to SDWMN in the sense that they typically have a dedicated high speed wired backbone which provides more reliable communication. Generally, SDWLAN appears to have more resilience against LFA attacks in contrast to SDWMN since they have significantly higher bandwidth compared to the bandwidth-limited mesh networks. However, they are vulnerable to LFA in the sense that in contrast to a rather stringent admission control and scarcity of bandwidth in SDWMN networks, the high bandwidth offered by SWLAN can also be abused by adversaries to launch comprehensive attacks. However, a key difference is that processing power is significantly higher compared to SDWMN and therefore LFA detection and mitigation can be more efficiently achieved in SDLAN as compared to SDWMN.

4. Link Vulnerabilities in SDN

In this section, we explain the weaknesses of SDN where LFA can be critical. We show how the control plane and data plane interfacing is critical and can be attacked. Earlier work on SDN security, mitigates attacks including flow table overflow [95, 165, 166] and bandwidth consumption attacks [116, 167, 86] which causes DoS [167, 168, 169] while others [170] check SDN for its deployment. In [90] authors propose a framework for mitigation control plane saturation attacks. In [95] a solution is proposed for the mitigation of control plane saturation attacks in which polling strategy is devised from data plane to control plane for checking of the control flows that come across for news communications. In [171] authors propose a packet migration and proactive flow rule-based solution to mitigate control plane saturation attacks. In [172] authors have identified vulnerabilities of SDN where an attacker can gain access to SDN and hence the network infrastructure. A study has been conducted in [70] which provides an architecture for a solution against LFA, however no experimental evidence and evaluation has been given.

Based on the above discussion, it is pertinent to say that previous work has focused on providing security specifically for the data plane or the control plane, while ignoring the vulnerable strategic links capable of driving the communication of the whole network. In case of LFA, the link between controllers to data plane can become a bottleneck for the successful operation of the network. The disconnection of this link causes the information not to pass between the two planes, resulting in failure of the network. All the previous work that has been done in the field of LFA is done on the data plane or control plane as we have discussed in the literature. Therefore, the literature findings on this issue are insufficient. So underlying work has great importance in mitigating attacks on the interface of the control plane and data plane. This work also contributes to overcoming the weakness of SDN to be vulnerable to LFAs. Most of the IT companies are transferring their infrastructure to the cloud [173]. According to Gartner [174] survey Amazon, Microsoft, and Google are providing 98% of the global computing facilities, and all of these are virtualized. This shows that more and more companies are transferring their infrastructure from hardware systems to software [175]. SDN is becoming more and more prevalent and replacing traditional networking schemes. Many threat vectors have been identified that can affect the SDN architecture [176, 177, 178, 179, 180, 181].

SDN-based networks provide easy adaptation to changing business requirements. Since SDN is a relatively new concept so, the threats it may experience are also novel. The new types of threats in SDN are likely to tend to flood the link between control planes to the data plane. If this happens in a network, the link from the control plane to data plane can be chocked and the controller becomes irresponsive and will no more be administrating the traffic. Since the controller is the brain in SDN, it installs rules in the switches which are in the data plane. The rule installation procedure cannot be performed and the entire network becomes irresponsive. This in turn, stops the services provided over the network. Since the network connection will be unavailable, network services on the cloud will also be affected and may cause the cloud infrastructure to become unavailable. Based on the above discussion, we can summarize the following weaknesses of SDN:

- The control traffic is the most critical information for SDN because it performs all the crucial tasks of flow rules installation, network configuration, and optimum path calculation. The prime concern of the attacker is to disconnect the controller path links to make it ineffective.

- Data plane is the main working unit which is dependent on the controller communication. In SDNs, where it derives all the information through the hardware devices, any attack on data plane devices can result in service disruption throughout the network.

- The link between control planes to the data plane is very critical because all the control information passes through this link, as this is a sole link and any congestion can cause a delay in information.
- The adversaries can take target this link by employing the hard-to-detect LFA and congest the link in a way that it becomes unresponsive resulting in disconnection of the whole network.
- The data to control plane link has the tendency to be flooded by a bunch of traffic so there must be a link-scanning mechanism that can check whether the congestion is because of the legitimate traffic or it has caused by some adversary.
- Since data plane to control plane link does not have any flood detection and mitigation mechanism, there is a strong need to implement a flood detection and mitigation strategy. This link is very crucial to install new flow rules for every new incoming flow, so it becomes a lucrative target for adversaries, which can easily affect the complete network by attacking this link.
- LFA disruption in control plane to data plane communication can have serious consequences in all types of networks including wireless. Interruption in linkages between the control plane to the network devices can ultimately bring the whole network down. So all the SDN variants are equally in a position to be attacked by crossfire LFA. In Fig. 12 the link between control plane to data plane is shown to be disconnected due to an LFA, resulting in communication failure of the control plane with network devices.

LFA has a potential to can cut off specific areas of the underlying networks from others, therefore, we suggest to device a solution for LFA mitigation on

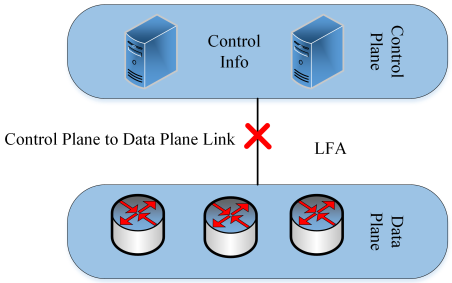

Figure 12: LFA can disconnect the control channel.

control plane to data plane linkage. LFA can have a devastating effect on SDN which can bring individual planes down, disrupt the communication between the planes and ultimately bring down the whole network. But the sensitive nature of the control plane to data plane link can make this a single point of failure for the whole network. So the network community should take this issue for successful implementation and up gradation of SDN architecture. The above study shows that there are profound vulnerabilities between the control plane to the data plane communication path. There is also a lack of literature available that addresses this weakness that shows the chief importance of our study.

5. LFA Mitigation Techniques

In this Section, we perform an in-depth analysis of the available LFA mitigation techniques.We compare existing techniques and categorize LFA mitigation according to the type of mitigation technique used.

5.1. Categories of LFA mitigation Techniques

We categories LFA mitigation techniques into three types based on the underlying method to alleviate LFA.

- Traffic engineering principles based mitigation techniques
- SDN principles-based approaches
- Link obfuscation based techniques

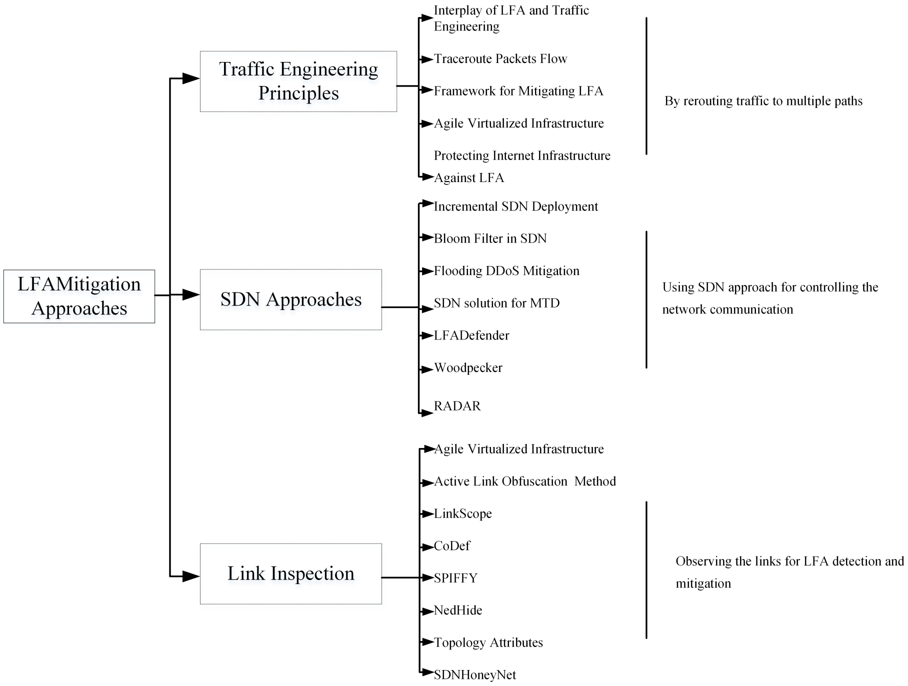
LFAMitigation Approaches
Traffic Engineering Principles
Interplay of LFA and Traffic Engineering
Traceroute Packets Flow
Framework for Mitigating LFA
Agile Virtualized Infrastructure
Protecting Internet Infrastructure Against LFA
By rerouting traffic to multiple paths
SDN Approaches
Incremental SDN Deployment
Bloom Filter in SDN
Flooding DDoS Mitigation
SDN solution for MTD
LFADefender
Woodpecker
RADAR
Using SDN approach for controlling the network communication
Link Inspection
Agile Virtualized Infrastructure
Active Link Obfuscation Method
LinkScope
CoDef
SPIFFY
NedHide
Topology Attributes
SDNHoneyNet
Observing the links for LFA detection and mitigation

Figure 13: Categories of LFA mitigation techniques.

5.1.1. Traffic Engineering Principles-based Approaches

A proactive LFA mitigation technique based on traceroute packets has been proposed by Takayuki et al. [34]. The traceroute packets flow increases in a region where LFA occurs because the adversary creates a link map before attacking a target link. The traceroute packets increase behavior is independent of link congestion, however, a limitation of this technique is that it is hard to detect traceroute packets from benign and attack hosts. A relational algebra-based approach to defend against LFA has been proposed in [37]. This technique is based on multiple attackers and defender interactions during rerouting. The flows that tend to change their destination in order to adapt to the new traceroute are identified as suspicious. Dimitrios et al. propose a reactive traffic engineering-based LFA mitigation technique which is based on rerouting the attack traffic [1]. However, a significant drawback of this technique is that the multiple rerouting efforts increase traffic delays and increase network overhead.

Multiple attack detection and mitigation resources have been incorporated in [182] including capacity invocation, blacklists integration, flow filtering, and normal traffic learning. Fid et al present Virtual Networks (VN) deployment to bypass attacked links [38]. A cost incentive mechanism has been proposed between the attack source and destination in order to alleviate LFA [183]. Their solution is based on the fact that source and destination AS never gets any idea of LFA. Therefore effective incentive mechanisms among source and destination AS encourage cooperation between them.

The above-mentioned techniques exploit traffic engineering to detect and defend against LFAs. Further details of these techniques can be accessed in our previous survey [70].

5.1.2. SDN Principles-based Approaches

SDN has revolutionized the traditional network management providing opportunities for better network traffic exploitation for attack mitigation [184]. These techniques are based on SDN deployments where LFA occurs in order to increase network connectivity in case of LFA. Wang et al. introduced a technique namely Woodpecker to upgrade traditional network routers to SDN switches to increase network connectivity [32]. Bloom filtering technique in SDN has been proposed in [40]. Link obfuscation technique during link map construction has been proposed in [41]. However, due to link obfuscation, the resulting links may not be optimal which increases network traffic overhead. In [185], authors have surveyed mitigation techniques against flooding attacks using SDN principles. LFADefender [186] performs data plane device measurements using sflow agents, and provide link selection, and LFA mitigation solutions using SDN. Similarly, an extension of Woodpecker has been proposed in [187] which increases network connectivity to mitigate LFA in traditional networks.

Fig. 13 and 14 contain the citation of literature available on LFA mitigation techniques. The broad categories of LFA mitigation techniques are provided along with their basic working principles.

5.1.3. Techniques based on Link Inspection Methods

These techniques are based on the link inspection during, before, and after an attack to defend against LFA. Qian et al. propose an active link obfuscation method to make it difficult for the adversary to create an accurate link map [32]. The LinkScope technique performs different network measurements to capture network metrics to defend against LFA [188]. RADAR [189] employs commercial of-the-shelf switches to mitigate crossfire attacks, they utilize multiple controller and switch interactions to inspect a number of aggregating flows to detect LFA. However, measuring a huge number of aggregating flows can become an overhead on the network. Soo et al. propose CoDef which is based on the collaboration among different AS during an attack [35]. SPIFFY technique is based on temporary bandwidth expansion during and after the attack [2]. The adversaries can be detected which do not adapt to the bandwidth expansion. NetHide [190] obfuscates the network topology to defend against LFA, however this technique transforms the LFA defense problem into multi objective problem where security and usability requirements are constraints on the solution. The impact of network topology attributes on target selection has been demonstrated by [191].

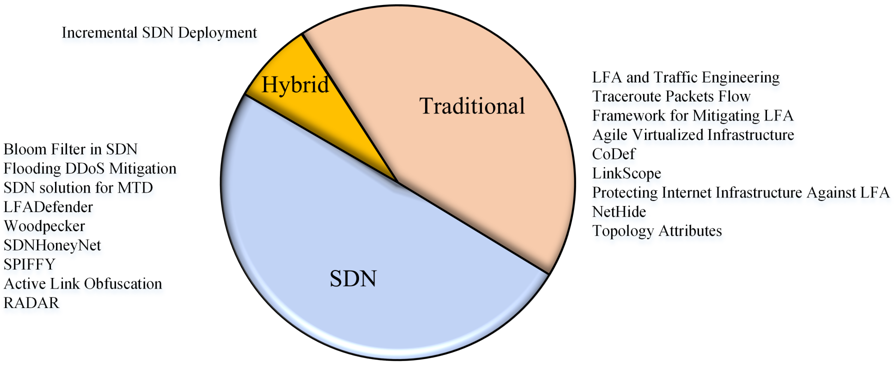

Figure 14: Grouping of techniques based on the type of the experimental setup.

5.2. Comparison of LFA Mitigation Techniques

Due to the ease of configuration and usage, in most of the recent works SDN testbeds have been used to validate proof of concepts (PoC) for most of the network-related research. However, the presented techniques are in no way focused to solve SDN specific issues. And effectively mitigating LFAs on SDN control plane is one such challenge that has yet to be addressed. Fig. 14 intends to categorize techniques presented in the LFA related literature based on the type of the experimental setup. While the wide usage of SDN testbeds in network research signifies its importance, it also gives a false impression to the research community that SDN itself is a very researched topic in the context of LFA. Therefore, this section discusses LFA mitigation techniques in detail while keeping different networking paradigms in focus. A thorough literature review clearly reveals that the work done in the field of LFA mitigation is mostly focused to the traditional networks. To the best of our knowledge, this work is the first effort to identify the LFA problem in SDN's control channel. This section explains the generally available literature in the field of LFA and provides a comparison of different mitigation techniques.

Gkounis et al. propose a reactive defense mechanism against LFA based on "centralized traffic engineering" [1]. Active link obfuscation-based technique has been proposed by [36]. Temporary bandwidth expansion-based mechanism has been presented by [2]. In [36], Wang et al propose a proactive link obfuscation method for LFA mitigation. This solution provides a fake link-map so that the adversaries misfire the target links. SVM is used for adversaries classification that remains accurate only if the training data is available in a very large volume. In [2] Kang et al. present a technique for LFA prevention, that works by expanding the network bandwidth dynamically. On this expansion bots are detected by their constant behaviour. However, legitimate users can also be marked as LFA-bots in case they do not react to the increase in the bandwidth. Further comparison of all the LFA mitigation techniques is given in the Table 5. This table gives an extended form of our preliminary findings [70], and gives comparative details on the available literature chronologically in the domain of

LFA mitigation techniques.

In Table 5 there are four columns each representing some property of the technique used for LFA mitigation. The first column is the solution type which is the name of the proposed solution given by the authors of the proposed work. In the main idea column, a summary of the main contribution of the proposed work is given. After reading the whole work and getting out the summary of the proposed work we identified the limitations of the proposed work. In Fig. 14 we have categorized the LFA mitigation techniques on the basis of the network type used. We have identified the type of testbed used by authors to prove their concept. Some approaches use traditional networks while others employ SDN testbed for validating their research. In the last column of Table 5, we have mentioned the published year of the LFA mitigation technique, we have categorized all the papers in chronological order and it can be seen that most of the work on LFA has started from 2015 and it goes on with a wide range of research can be found in 2018.

We have carried out an extensive literature review of the area and classified the research papers into certain categories. Table 5 illustrates that the research focusing on LFA on different networks started in 2013; this can be explained because LFA on SDN is a relatively new phenomena which poses new challenges to the widely deployed SDN. And because of the wide adaptation of SDN, it can be lethal to the network operation. It can be observed from the Table 5 that most authors have carried out their experiments on SDN to solve LFA in general, not specific to SDN. It is also worth mentioning that none of the works considered experimentation related to LFA mitigation on data plane to the control plane link. The presented techniques can broadly be categorized into traffic engineering, link inspection, and SDN based approaches. A reactive solution can be a bit destructive because LFA may already damage the underlying network by the time of being detected. Such solutions carry the overhead of continuously monitoring the link or underlying network for LFA. This can slow down the network and overburden the infrastructure. On the other hand, proactive solutions can mitigate attacks before they become threats and avoid disastrous situations.

It can be observed that all the work that is discussed in Table 5 is either on the data plane, control plane or on managing traffic on both of the planes. But none experimented on the data plane to control plane link -which is very critical. Due to its widespread deployments, SDN based approaches can be widely seen in the literature. We have identified the weaknesses of each technique that should be addressed while designing a framework.

5.3. Classification of LFA Mitigation Techniques

A deep study of LFA mitigation techniques has shown that researchers have examined this problem from multiple angles. We did not find any single solution that addresses all the challenges posed by this attack. In the absence of an objective criterion it is hard to judge the quality of a suitable solution. In this regard we have proposed a set of quality metrics for systematically examining the strengths and weaknesses of LFA mitigation techniques based on an objective criterion that uses a set of features to rank and quantify them. The qualitative performance features are discussed below:

1. **Detection Accuracy:** It is the accuracy with which an LFA mitigation technique alleviates the flood traffic where false positive rate is low. Higher accuracy is desirable in LFA mitigation strategies. We group the LFA mitigation techniques according to their detection accuracy levels of high, medium, and low.

2. **Detection Time:** It is time that a solution takes to detect LFA after it occurs. Low detection time is desirable in LFA mitigation techniques because if a solution takes much time in order to respond to LFA, then it will not be better solution. The detection time is also classified into high, medium and low, where time-efficient techniques will be categorized as low as their detection time. There is another aspect of detection time where proactive solutions try to establish strong surveillance of the network before the occurrence of LFA. Therefore the proactive solutions have

Table 5: Link flooding mitigation techniques (An extension of our work at [70]).

Solution Title	**Main Idea**	**Limitations**	**Year**
Codef [35]	Rerouting traffic toward AS domains not affected by LFA.	Threat of all AS domain attack.	2013
LinkScope [192]	Continuous link inspection and identification using ML.	High false positive rate.	2014
Agile Virtualized Infrastructure [38]	Proactively allocating network resources using VN placement.	Expose the identity of VN.	2015
Traceroute Packets[34]	Traceroute packets increase in LFA regions.	Differentiating adversaries and hosts.	2015
Flooding DDoS [182]	Traffic inspection e.g., blacklist traffic features, and elastic capacity invocation.	Overlap in attack and benign traffic if source IP is spoofed.	2016
Incremental SDN [32]	Upgrade routers to SDN switches to increase network connectivity.	Reactive nature.	2016
Bloom Filter in SDN [40]	Traffic inspection using bloom filtering.	High false positive rate.	2016
SPIFFY [2]	Temporarily bandwidth increase.	High false positive rate.	2016
SDN Approach for MTD [41]	Link obfuscation.	Increased delay.	2016
Framework for LFA [37]	Multiple attacker and defender interaction.	Increased delay.	2016
Interplay of LFA [1]	Defender-based rerouting of attack traffic.	Increased delay.	2016
Link Obfuscation [36]	Providing fake links to the adversaries.	Increased training data.	2017
SDN HoneyNet [193]	Compute graph metrics and deploy fake link map to adversaries	Increase network complexity.	2017
Protecting Internet [183]	ISP cooperation and traffic rerouting by incentivized routing strategy	Traffic overhead due to rerouting.	2018
LFADefender [186]	SDN-based techniques to the vulnerable links.	Increased network delay.	2018
Woodpecker [187]	SDN deployment for increased connectivity.	Intense traffic measurements.	2018
RADAR [189]	Controller-data plane cooperation.	False positive rate.	2018
NetHide [190]	Hiding network topology.	Extra packet processing.	2018
Topology Attributes [191]	Topological attributes for security.	Reactive nature.	2018
CFADefense[194]	Traffic engineering. security.	Extended delay	2019
Signaling Game [195]	Attacker-defender interaction security.	Increased complexity	2019
Stackelberg security[196]	Randomized detection strategies.	Higher traffic delay	2019
LFA-Shield[197]	Traffic detection using deep learning.	Lower accuracy.	2019
BALANCE[198]	Hybrid SDN deployment.	Reactive nature.	2020
LFA Using ML[199]	Employing ML approaches. deployment.	Slower response.	2020

Table 6: Comparison of LFA mitigation techniques on the basis of performance metrics.

Technique	**Accuracy**	**Time**	**Solution**	**Approach**	**Complexity**	**Analysis Method**	**Scalability**	**Evaluation**
CoDef [35]	High	Low	Reactive	Traditional	High	Traceroute	High	Simulation
Linkscope [192]	High	Low	Reactive	Link Inspection	Low	Link Inspection	High	Real testbed
Agile Virtualized [38]	High	Medium	Proactive	Traditional	Low	VN placement	High	Real testbed
Traceroute Packets [34]	Low	Low	Proactive	Traceroute	Low	Traceroute	Low	Simulation
Flooding DDoS [182]	Medium	High	Proactive	Machine Learning	High	SDN-based	High	Real testbed
Incremental SDN[32]	Low	High	Reactive	Woodpecker	High	SDN-based	Low	Simulation
Bloom Filter in SDN [40]	Low	High	Reactive	Bloom filter	High	SDN-based	Low	Simulation
SPIFFY [40]	Medium	High	Reactive	Traditional	High	Link Inspection	Medium	Simulation
SDN MTD [41]	Low	High	Proactive	SDN-based	Medium	SDN-based	Low	Simulation
Framework for LFA [37]	Medium	High	Reactive	Traditional	low	Traffic Rerouting	Medium	Simulation
Interplay of LFA[1]	Medium	High	Reactive	Traditional	Medium	Traffic Rerouting	Medium	Simulation
Link Obfuscation [36]	High	Low	Proactive	Link obfuscation	medium	Link inspection	High	Simulations
SDN HoneyNet [193]	Low	High	Reactive	HoneyNet	Low	Link Inspection	Low	Simulation
Protecting Internet[183]	High	Medium	Proactive	Traditional	Low	Traffic Engineering	High	Real testbed
LFADefender [186]	Low	Low	Proactive	Traditional	Low	SDN-based	High	Real testbed
Woodpecker [187]	Medium	High	Reactive	Woodpecker	Low	SDN-based	Medium	Real testbed
RADAR [189]	High	Low	Proactive	Traditional	Low	Traffic Engineering	High	Real testbed
NetHide [190]	Medium	Medium	Reactive	Link Inspection	Medium	Traffic Engineering	Low	Simulation
Topology Attributes [191]	Medium	High	Reactive	Link Inspection	Medium	Traffic Engineering	low	Simulation
CFADefense [194]	Low	High	Reactive	Link Inspection	Low	SDN Measurements	low	Simulation
Signaling Game [195]	High	Low	Proactive	Game Theory	Low	Traffic Rerouting	High	Real testbed
Stackelberg security [196]	High	Low	Proactive	Game Theory	Low	Traffic Measurement	High	Simulation
LFA-Shield [197]	Low	High	Reactive	SDN-based	Low	SDN Measurements	low	Simulation
BALANCE[198]	High	High	Reactive	SDN-based	Low	Central Management	High	Simulation
LFA Using ML[199]	Low	High	Reactive	ML	Low	Traffic Engineering	low	Simulation

low detection time.

3. **Solution Type:** As discussed earlier, there are broadly two solution types i.e. proactive and reactive. The proactive strategies provide mechanisms to avoid LFA before it occurs, as the network traffic is handled in a way that it is very difficult for the adversary to launch an attack. And the reactive strategies perform the alleviation in response to the occurrence of the attack. Proactive solutions are generally considered as a better choice against LFA as it targets to minimize the likelihood of any damage caused by LFAs.

4. **Approach Used:** It corresponds to the type of approach used in the experiment, for example traditional, link obfuscation, and machine learning. Traditional techniques correspond to the approaches widely being used for network operation like packet inspection and flow filtering. Machine learning techniques correspond to using machine learning approaches for detection and mitigation of LFA like incoming traffic classification to identify malicious flows and adversaries. Link inspection techniques correspond to constantly observing the links to identify the malicious activity over the network. Traceroute techniques use traceroute packets in order to alleviate and detect an attack. These techniques analyze traceroute packets increase phenomena to identify the malicious activity, because adversaries send multiple traceroute packets to create a link map of the network before launching the attack. For attack mitigation, these techniques reroute the attack traffic so that it is unable to reach the destination. In the same way, link obfuscation techniques, deceive the adversary to create a correct link map and identify the potential target by obfuscating the network links. HoneyNet techniques provide a virtual connectivity over the nodes attacked by the adversary, this way traffic can be bypassed from the attack point. The Bloom filtering approach uses probabilistic data structures to detect LFA. In the same way, Woodpecker technique incrementally deploys SDN strategies to alleviate LFA and to restore the centralized control.

5. **Solution Complexity:** Complexity is the amount of resources that will be consumed by the solution in order to mitigate the attack. These resources may corresponds to time to detect or mitigate an attack or other resource requirements taken by the solution to complete its operation. Low complexity values are preferred for a good solution against LFA. Here we classify the techniques into low, medium, and high complexity solutions by analyzing the experimental setup provided by each technique.

6. **Analysis Method:** It corresponds to the type of solution that has been used by the underlying approach. The examples of analysis methods are: traffic engineering, traceroute, and SDN-based approaches. The difference between the approach used and analysis method is that the approach quantifies specific categories of the technique used against LFA. However, analysis method quantifies the solution into more broader categories of the techniques.

7. **Scalability:** The capability of a solution to be deployed on large scale networks is called the scalability. We categories the scalability of solutions as, high, medium, and low. Current networks are expanding continuously, so a good solution must possess the quality of being high scalable. Most of the available research is based on simulation results, which cannot be tested for scalability.

8. **Evaluation:** Different methods are used to perform the evaluation of the proposed solution. Some authors use simulations to demonstrate their concept while others use real testbeds. Preference is given to solutions that are tested on real testbeds.

In Table 6 we have compared LFA mitigation techniques based on above mentioned metrics. Techniques such as Codef [35] and Linkscope [192] have high accuracy however they work in a reactive manner that makes them vulnerable to the security issues. Alternatively, the Agile Virtualized [38] technique possesses the qualities for a good solution as it has higher accuracy in detecting the

Table 7: Machine learning techniques in flood attack mitigation.

Technique Name	**Machine Learning Algorithm**	**Features Used**
Distributed SOM [170]	Distributed Self Organizing Maps	Number of flows
		Number of packets per flow
		Number of bytes per flow
		Time duration
Adaptive Artificial Immune Networks [200]	Artificial Immune System	Traffic Analysis
DyProSD [201]	C4.5 Naive Bayes Decision Tree	Source IP
		Destination IP
		Sampled Interval Time
		Flow ID
		Total Number of Connections
OpenFlowSIA [202]	SVM	Network Protocol
		Source IP
		Source Port
		Destination IP
		Destination Port
DDoS Flooding Attack Detection Algorithm [89]	Hop Count Filter Algorithm	Source IP
		Destination IP
TDFA [203]	IP Trace-back Algorithm	Traffic Analysis

malicious flows as well as it is highly scalable. it also works in a proactive manner and moreover they have tested their solution on a real testbed. SDN HoneyNet [193] is a novel solution however, it has low detection accuracy and it can suffer from the problem of more time to deploy the HoneyNet topology because it computes different graph-based measurements to identify the vulnerable nodes. Another drawback of this technique is that it is not tested in the real testbed scenarios. SDN approach for moving target defense provides a solution that works fine in both proactive and reactive scenarios. In the same way, it takes a lot time in order to detect LFA. This solution has not been tested on real testbed and it also lacks in qualifying scalability requirements.

The selection for a reliable solution against LFA depends on many factors as provided in the comparison metrics. It depends on the requirements of the environment where a solution has to be deployed. Because no solution is perfect in terms of all the defined metrics. In some situations we require highest accuracy, so we can compromise some other quality metrics in order to attain accuracy requirements. Some basic insights for choosing a mitigation strategy is that the solution should have low false positive rate and low attack detection time. No solution can be perfect in terms of all the performance metrics specified in the Table 6. An optimal strategy against LFA involves a trade-off, however, a solution needs to be scalable to be practical. However, we categories the state of the art Protecting Internet Infrastructure Against LFA [183] as a reliable solution because it is a novel strategy which has high detection accuracy and works proactively.The fact that LFA uses low rate traffic to attack potential targets, makes it difficult to mitigate. However, this strategy incorporates the source and destination ASes coordination against LFA. In the same way this solution has also been tested on a real testbed and also qualifies the scalability requirements. The underlying detection and mitigation technique is also traditional which can easily be implemented in current network environments. This technique incorporates the incentivized-based strategy on both source as well as destination sides to collaborate against LFA.

Due to the exponential growth in the field of cloud computing [204, 205] and big data [206], machine learning techniques are widely being used for optimization [207, 204, 208] and classification [209, 210]. In this regard, we have identified multiple flood attack mitigation related research in Table 7. This table contains the list of ML techniques and features used for flood attack detection. Traffic features are used to train the ML classifier, it can be noted that most of the researchers used source and destination related traffic features for flood traffic classification.

5.4. Case study: Link flooding mitigation in industrial automation

The growing popularity of Industrial Internet of Things (IIoT) and Industry 4.0 are spurring the use of innovative network technologies in industrial automation. This increase in industrial automation communications is a result of the growing shift to harness the productivity and efficiency of manufacturing and process automation while requiring a minimum intervention of a human operator. As the protocols involved evolve from serial bus technologies to Ethernet, engineers are looking to harness the benefits of Software Defined Networking for achieving flexibility in network design of the manufacturing floor.

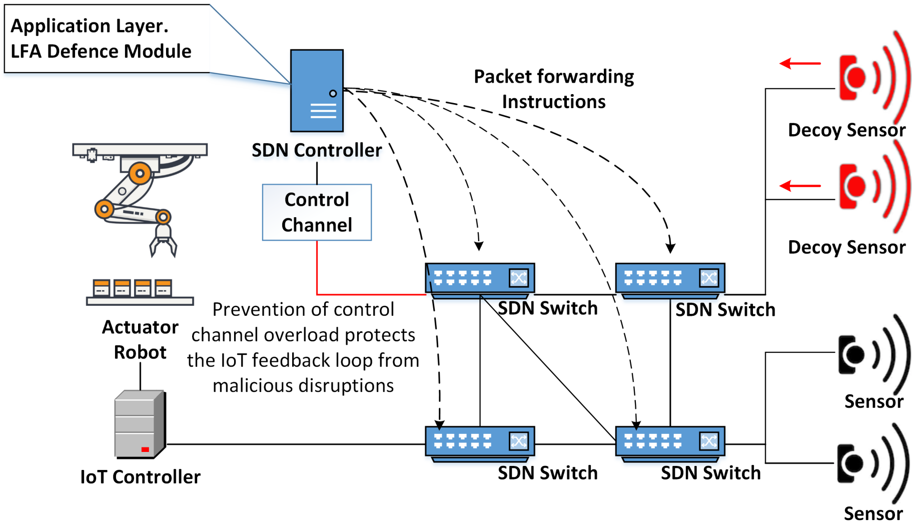

Figure 15: Case study of link flooding attack mitigation in industrial automation.

This section presents a proposed theoretical framework for mitigating link flooding attacks using industrial automation as a case study. A sample scenario has been illustrated in Figure 15

Unlike IT networks, industrial control networks do not undergo frequent changes. While IT networks are dynamic and flexible, control networks are responsible for critical processes and high-speed decision making, which demand a network that is much more predictable and deterministic. In traditional networking, the switches that forward packets also determine the network path to send those packets through, using protocols such as RSTP. In SDN, by contrast, the decision making functions are removed from the switches and handled instead by the centralized SDN controller. The switches, in turn, receive packet-forwarding instructions from the SDN controller. The SDN controller and switches are shown in blue in Figure 15. This architecture enables the switches to focus solely on the physical forwarding of packets. Each device knows in advance what to do in case of a network failure. Since there is no need to negotiate forwarding paths, as in an RSTP Ethernet network, there is almost no delay in forwarding packets when there is a failure, which speeds up recovery and minimizes packet loss.

IIoT leverages the concept of the sensor-controller-actuator feedback loop. The scenario illustrated comprises an IoT controller running a dedicated service such as the sorting function of an actuator, namely a robotic arm. This controller is functionally dependent on the context information from the overlapping set of sensors that monitor the speed and quality of the objects being sorted on a conveyor belt. Sensors measure environmental features and are vulnerable to fabrication attacks whereby an attacker can create decoys designed to send crafted messages to the controller. In the figure, the decoys are depicted in red. At the start of the attack, the attacker creates a link map of the network, calculates the attack-cost strategy, and ascertains the number of decoys that can occupy the links. Finally, the adversary sends traffic from the decoys to occupy the critical link thereby disrupting legitimate control traffic. This is easily possible because of the way the control networks are programmatically set up with clearly demarcated primary and backup links as described earlier. Disruption of the SDN controller affects the feedback loop of the IoT controller ultimately causing the factory robot to malfunction.

An LFA detection module implemented as an application in the SDN controller gathers key network traffic statistics from the switches, proactively identifying the presence of decoys and can prevent this attack from ever occurring. There are several variations of the LFA attack shown in this scenario. For example, the attacker can use rogue sensors as decoys instead of compromising existing ones, or manipulate the IoT controller to act on his behalf.

5.5. Ongoing Large-scale Projects

To gauge the viability of newly developed LFA mitigation techniques and network security research in general, emulators, simulators as well as real testbeds are employed. Compared with simulators and emulation platforms, testbeds are a preferred choice because they utilize real network components and actual traffic can be observed. The past 10 years have seen several noteworthy large-scale testbed projects come up to serve the SDN security community. Using network slicing technologies and cloud management these projects are transparently "renting" out isolated, and virtual network resources to researchers from common physical network infrastructure. Just as the arpanet was the precursor to today's internet, researchers consider such large-scale SDN testbeds to growth into the next-generation Internet. This section summarizes the projects most relevant to LFA research being currently conducted.

DeterLab [211] is a 700 node scientific computing facility and heavily relies on worldwide as a testbed platform for conducting SDN security related research. It is hosted at the Information Sciences Institute, of the University of Southern California (ISI) and University of California at Berkeley. Since 2003, with funding from NSF, DHS, and DARPA, DETERLab has grown into a facility where over 900 researchers have conducted network and cybersecurity experimentation. DETERLab users have conducted hundreds of research projects and more than 13,000 students have received hands-on cybersecurity training via DETERLab.

Research efforts are underway to determine performance impact of flooding attacks and their countermeasures on SDN derived from the Internet Topology Zoo (ITZ) project [212]. This is a repository of over two hundred and fifty network topologies. Network operators upload topologies of real networks they work with to the ITZ repository. It contains topologies from AboveNet to Zamren. ITZ uses a graphical format, based on XML that contains detailed information for setting up testbed networks based on real world topologies.

The Global Environment for Network Innovation project better known as GENI [213] is funded by the National Science Foundation (NSF). It employs innovative network technologies like virtualization, OpenFlow and software defined networks, to conduct large-scale network design experiments. Security researchers developing solutions for link flooding attacks using GENI have access to a distributed virtual laboratory with resources deployed on over 40 university campuses across the U.S.

OFELIA [214] is an European Seventh Framework Programme (FP7) project that started in 2010. OFELIA's goal is to design a geographically dispersed next generation Internet that has multiple layers and domains to support a variety of traffic types including that for software defined networks. It is also being used for developing DDoS mitigation solutions.

OneLab [215] is another FP7 funded project, that attempts to combine some existing testbeds to be an open, scalable, and sustainable SDN infrastructure for next generation Internet research. Since its inception in 2011, the project has deployed a variety of novel network protocols and applications in its branch testbeds. OneLab is utilized in particular by the EXPRESS project by CNIT researchers in Italy to support SDN traffic engineering rules; to program security or launch monitoring actions, as a function of anomaly detection warnings.

Other noteworthy testbeds worth mentioning are the Australia Wide-Area SDN (AARNET) testbed [216] and the OpenSDNCore [217] testbed in Berlin.

6. Research Issues and Future Directions

We have identified multiple open research issues after an in-depth analysis of research on LFA in SDN. Generally, a lot of work has been performed to mitigate LFA in traditional networks, however, relatively less attention has been given on LFA in the context of SDN. As the current networks are widely deploying SDNs for network operations, there is a strong need to provide LFA mitigation strategy for SDN [194]. The Table 5 highlights it as a major research challenge.

6.1. Control Channel LFA

Devising a solution for control channel LFA has a potential to become a valuable scientific contribution, as mentioned in section II. Following guidelines should be considered while developing such a solution.

- Any solution for control channel LFA should be an independent application to be efficient and effective.

- To minimize the chances of application crashing the solution should be architected in a way that limits its external communication, thereby increasing its reliability.

- In the current network circumstances, the traffic consistency is random, so any acceptable solution must be scalable according to the incoming traffic.

- In real time network scenarios traffic inspection and analysis takes a significant amount of time so, it is difficult to provide a real-time solution for LFA. However, there is a strong need for a solution that works on real-time basis because LFA can immediately disrupt the communication with the vital resource which can result in vital information loss.

6.2. Pattern Matching and Machine Learning as a First Line of Defence

The adversaries exhibit special traffic patterns when they attack. In this regard, the identification and mitigation of these patterns can help against LFA. Efficient pattern matching sits at the heart of the high-speed network traffic monitoring. Intrusion detection systems, web application firewalls and deep content inspection use predefined patterns to identify malicious traffic and content [218, 219]. For providing a solution against LFA on SDN, pattern matching techniques can be designed which helps in analyzing traffic patterns on line speed [220]. The normal traffic features can be represented as signatures and can efficiently be used to identify malicious patterns in the network [173]. Pattern matching techniques can be utilized as the first line of defense against LFA by filtering the flood from the known attackers. When traffic streams are processed through the pattern matching engine, the traffic can be further analyzed using machine learning techniques.

Machine learning techniques are widely being deployed for network traffic classification [221]. Table 7 describes the machine learning techniques used for flood traffic classification. A solution based on machine learning techniques can be devised to classify flood traffic on control channel [222]. Similar to a solution provided by [223], for pattern matching in internet of things, a solution can be provided for LFA attack patterns identification using historical attack data, which can be used to classify traffic flow statistics into benign and flooding flow categories. This vital identification can be further utilized to mitigate the flood traffic using any state of the art mitigation strategy.

6.3. Security Against Sources of LFA

The attack traffic in LFA is persistently sent from the adversaries because most of the times they use network of bots to send low rate traffic. All the current solutions work by mitigating these attacks on the victim side, there is a strong need to provide solutions at the source side of LFA. However, it is very difficult to identify the sources of LFA because the intensity of traffic at the source sides is very low which is very hard to discriminate.

A better solution to alleviate the sources of LFA is through coordination between the ASes at the source and destination sides. However, the source ASes need economic incentives to collaborate with the destination ASes. Source side security solutions can also be provided by implementing highly secure authentication services to identify and alleviate malicious hosts.

6.4. Need for Robust Solutions

A much needed future direction is to develop a solution to mitigate LFA in SDN as an application. Most of the current controllers use REST API to communicate with the physical hardwares in the network. REST API can be utilized to collect flow statistics which subsequently can be used for surveillance of the control channel.

With the rapid developments in the field of information and communication technology, need for reliable networks have been increased. Most importantly, with the immense development of less-secure sensing technologies, the attack surface for the adversaries has been increased [224]. The IoT devices serve as readily-available devices which can be easily exploited by the adversaries to generate attacks. Therefore, robust solutions against LFA are extremely vital in the massively deployed sensing networks. LFA has become one of the most dangerous attacks on the networks. So, there is a dire need to provide highly robust defense mechanism against these attacks. Most of the available solutions for LFA are reactive in their implementation. With the advent of big data technologies most of the organizations are operating online; so, highly secure networks are preferred by the organizations. Therefore, need for proactive and reliable solutions have been increased.

Most of the current solutions against LFA are tested on simulations scenarios which do not guarantee their effectiveness if deployed in real network environments. Therefore there is a strong need to provide solutions evaluated on on real testbeds. This way the requirements of scalability, complexity, and real-time accuracy can be validated.

7. Conclusion

Since the rise of cloud services, big data processing on large server farms and changing traffic patterns, there has been an increasing need for dynamically adaptable and manageable network architectures. The SDN ecosystem has grown to fill this gap. After a complete revamp of the data-center networks market, SDN is now growing popular in mobile networking and wide-area networks. It allows seamless management of large complex networks by centralizing the network intelligence into one network control component. Unfortunately, intelligence centralization has its own disadvantages related to security. LFA attacks are considered crippling for traditional networks and are even more devastating for the SDN ecosystem. Most of the existing solutions do not focus on the unique ways in which LFA can harm SDN and hence stand ineffective.

In this work, we have performed an in-depth study of the outcomes of LFA on all SDN planes. Subsequently, we evaluated the effect of LFA on all the SDN variants e.g. SDMN, SDWN, and SDMN. The remaining part of this work focused on surveying mitigation techniques proposed for securing SDN infrastructure from LFA. We proposed a set of metrics for judging the quality of suitable solutions and then ranked the surveyed techniques based on this criterion. While no single technique ranked highly on the criteria, we found certain categories of techniques did better than others. Finally, we discussed avenues for future research such as employing the use of machine learning and pattern matching to improve the mitigation. We also provided guidelines when implementing mitigation techniques that would allow them to be more proactive and robust.

References

[1] D. Gkounis, V. Kotronis, C. Liaskos, X. Dimitropoulos, On the interplay of link-flooding attacks and traffic engineering, ACM SIGCOMM Computer Communication Review 46 (1) (2016) 5–11.

[2] M. S. Kang, V. D. Gligor, V. Sekar, Spiffy: Inducing cost-detectability tradeoffs for persistent link-flooding attacks, in: Proc. of the Network and Distributed System Security Symposium (NDSS), 2016, 2016, pp. 1–1.

[3] M. S. Kang, S. B. Lee, V. D. Gligor, The crossfire attack, in: Proc. IEEE Symposium on Security and Privacy, 2013, 2013, pp. 127–141.

[4] K. Slavov, D. Migault, M. Pourzandi, identifying and addressing vulnerabilities and security issues in sdn, Ericsson Technology Review (2015).

[5] Y. Xie, S.-Z. Yu, Monitoring the application-layer ddos attacks for popular websites, IEEE/ACM Transactions on Networking (TON) 17 (1) (2009) 15–25.

[6] J. Yu, Z. Li, H. Chen, X. Chen, A detection and offense mechanism to defend against application layer ddos attacks, in: Third International Conference on Networking and Services, (ICNS), 2007, pp. 54–54.

[7] S. Ranjan, R. Swaminathan, M. Uysal, A. Nucci, E. Knightly, Ddos-shield: Ddos-resilient scheduling to counter application layer attacks, IEEE/ACM Transactions on networking 17 (1) (2009) 26–39.

[8] Y. Jarraya, T. Madi, M. Debbabi, A survey and a layered taxonomy of software-defined networking, IEEE communications surveys & tutorials 16 (4) (2014) 1955–1980.

[9] R. C. F.-T. in Software-Defined Networking, Ravana: Controller fault-tolerance in software-defined networking, ACM, SOSR 2015 8 (10) (2015) 12.

[10] K.-y. Chen, A. R. Junuthula, I. K. Siddhrau, Y. Xu, H. J. Chao, Sdnshield: Towards more comprehensive defense against ddos attacks on sdn control plane, Communications and Network Security (CNS), 2016 IEEE Conference on 10 (31) (2016).

[11] W. Zhou, L. Li, M. Luo, W. Chou, Rest api design patterns for sdn northbound api, in: 28th, IEEE International Conference on Advanced Information Networking and Applications Workshops, (WAINA), 2014, pp. 358–365.

[12] P. Sharma, S. Banerjee, S. Tandel, R. Aguiar, R. Amorim, D. Pinheiro, Enhancing network management frameworks with sdn-like control, in: Proc. IFIP/IEEE International Symposium on Integrated Network Management, (IM), 2013, pp. 688–691.

[13] A. Hakiri, A. Gokhale, P. Berthou, D. C. Schmidt, T. Gayraud, Software-defined networking: Challenges and research opportunities for future internet, Computer Networks 75 (2014) 453–471.

[14] S. Cho, S. Chung, W. Lee, I. Joe, J. Park, S. Lee, W. Kim, An software defined networking architecture design based on topic learning-enabled data distribution service middleware, Advanced Science Letters 21 (3) (2015) 461–464.

[15] S. Sezer, S. Scott-Hayward, P. K. Chouhan, B. Fraser, D. Lake, J. Finnegan, N. Viljoen, M. Miller, N. Rao, Are we ready for sdn? im-

plementation challenges for software-defined networks, IEEE Communications Magazine 51 (7) (2013) 36–43.

[16] A. Dixit, F. Hao, S. Mukherjee, T. Lakshman, R. Kompella, Towards an elastic distributed sdn controller, in: ACM SIGCOMM Computer Communication Review, Vol. 43, 2013, pp. 7–12.

[17] B. Chandrasekaran, T. Benson, Tolerating sdn application failures with legosdn, ACM, HotNets-XIII 9 (2014) 7.

[18] R. Sahay, G. Blanc, Z. Zhang, H. Debar, Towards autonomic ddos mitigation using software defined networking, in: NDSS Workshop on Security of Emerging Networking Technologies. Internet society, 2015, p. 7.

[19] P. Zhang, H. Wang, C. Hu, C. Lin, On denial of service attacks in software defined networks, Network Forensics and Surveillance for Emerging Networks 10 (2016) 6.

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

[21] L. Dridi, M. F. Zhani, Sdn-guard: Dos attacks mitigation in sdn networks, 5th IEEE International Conference on Cloud Networking 2 (9) (2016) 5.

[22] K. Kalkan, G. Gur, F. Alagoz, Defense mechanisms against ddos attacks in sdn environment, IEEE Communications Magazine 55 (09) (2017) 175 – 179.

[23] K. ElDefrawy, T. Kaczmarek, Byzantine fault tolerant software-defined networking (sdn) controllers, in: IEEE 40th Annual Computer Software and Applications Conference (COMPSAC), 2016, p. 7.

[24] L. Xu, J. Huang, S. Hong, J. Zhang, G. Gu, Attacking the brain: Races in the sdn control plane, 26th USENIX Security Symposium 9 (2017) 19.

[25] B. Yuan, D. Zou, S. Yu, H. Jin, W. Qiang, J. Shen, Defending against flow table overloading attack in software-defined networks, Transactions on Services Computing 10 (09) (2016) 14.

[26] T. Sasaki, C. Pappas, T. Lee, T. Hoefler, A. Perrig, Sdnsec: Forwarding accountability for the sdn data plane, 25th International Conference on Computer Communication and Networks, (ICCCN) 19 (10) (2016).

[27] S. Khan, A. Gani, A. W. A. Wahab, M. Guizani, M. K. Khan, Topology discovery in software defined networks: Threats, taxonomy, and state-of-the-art, IEEE Communications Surveys & Tutorials 19 (1) (2017) 303–324.

[28] K. Sood, S. Yu, Y. Xiang, Performance analysis of software-defined network switch using m/geo/1 model, IEEE Communications Letters (2016).

[29] R. S. Ramanujan, M. N. Kaddoura, X. Wu, K. S. Millikin, Protecting networks from access link flooding attacks, uS Patent 7,356,596 (2008).

[30] Q. Niyaz, W. Sun, M. Alam, Impact on sdn powered network services under adversarial attacks, Procedia Computer Science 62 (2015) 228–235.

[31] H. Kim, M. Schlansker, J. R. Santos, J. Tourrilhes, Y. Turner, N. Feamster, Coronet: Fault tolerance for software defined networks, in: 20th IEEE International Conference on Network Protocols, (ICNP), Vol. 2, 2012, p. 2.

[32] L. Wang, Q. Li, Y. Jiang, J. Wu, Towards mitigating link flooding attack via incremental sdn deployment, in: IEEE Symposium on Computers and Communication, (ISCC), 2016, pp. 397–402.

[33] S. Yu, W. Zhou, W. Jia, S. Guo, Y. Xiang, F. Tang, Discriminating ddos attacks from flash crowds using flow correlation coefficient, IEEE Transactions on Parallel and Distributed Systems 23 (6) (2011) 1073–1080.

[34] T. Hirayama, K. Toyoda, I. Sasase, Fast target link flooding attack detection scheme by analyzing traceroute packets flow, in: Proc. IEEE International Workshop on Information Forensics and Security (WIFS), 2015, pp. 1–6.

[35] S. B. Lee, M. S. Kang, V. D. Gligor, Codef: Collaborative defense against large-scale link-flooding attacks, in: Proc. ACM 9th Internatioal Conference on Emerging networking experiments and technologies, 2013, pp. 417–428.

[36] Q. Wang, F. Xiao, M. Zhou, Z. Wang, H. Ding, Mitigating link-flooding attacks with active link obfuscation, arXiv preprint arXiv:1703.09521 (2017).

[37] C. Liaskos, V. Kotronis, X. Dimitropoulos, A novel framework for modeling and mitigating distributed link flooding attacks, in: Proc. IEEE 35th International Conference on Computer Communications, INFOCOM, 2016, pp. 1–9.

[38] F. Gillani, E. Al-Shaer, S. Lo, Q. Duan, M. Ammar, E. Zegura, Agile virtualized infrastructure to proactively defend against cyber attacks, in: Proc. IEEE Conference on Computer Communications, (INFOCOM), 2015, pp. 729–737.

[39] L. Xue, X. Luo, E. W. Chan, X. Zhan, Towards detecting target link flooding attack, in: Proc. 28th Large Installation System Administration Conference (LISA), 2015, pp. 81–96.

[40] X. Peng, Z. Li, H. Qi, W. Qu, H. Yu, An efficient ddos detection with bloom filter in sdn, in: Trustcom Bigdatase Ispa, 2017, pp. 1–6.

[41] A. Aydeger, N. Saputro, K. Akkaya, M. Rahman, Mitigating crossfire attacks using sdn-based moving target defense, in: IEEE 41st Conference on Local Computer Networks, (LCN), 2016, pp. 627–630.

[42] O. Vermesan, P. Friess, P. Guillemin, S. Gusmeroli, H. Sundmaeker, A. Bassi, I. S. Jubert, M. Mazura, M. Harrison, M. Eisenhauer, et al., Internet of things strategic research roadmap, Internet of Things-Global Technological and Societal Trends 1 (2011) (2011) 9–52.

[43] H. Wang, Y. Zhang, J. Cao, et al., Effective collaboration with information sharing in virtual universities., IEEE Trans. Knowl. Data Eng. 21 (6) (2009) 840–853.

[44] S. Sezer, S. Scott-Hayward, P. K. Chouhan, B. Fraser, D. Lake, J. Finnegan, N. Viljoen, M. Miller, N. Rao, Are we ready for sdn? implementation challenges for software-defined networks, IEEE Communications Magazine 51 (7) (2013) 36–43.

[45] J. M. Corchado, J. Bajo, D. I. Tapia, A. Abraham, Using heterogeneous wireless sensor networks in a telemonitoring system for healthcare, IEEE transactions on information technology in biomedicine 14 (2) (2010) 234–240.

[46] B. Barritt, W. Eddy, Temporospatial sdn for aerospace communications, in: AIAA SPACE Conference and Exposition, 2015, p. 4656.

[47] W. Rafique, M. Khan, N. Sarwar, W. Dou, Sociorank*: A community and role detection method in social networks, Computers & Electrical Engineering 76 (2019) 122–132.

[48] M. E. Kabir, H. Wang, E. Bertino, A role-involved purpose-based access control model, Information Systems Frontiers 14 (3) (2012) 809–822.

[49] Y. Qu, S. Yu, W. Zhou, S. Peng, G. Wang, K. Xiao, Privacy of things: Emerging challenges and opportunities in wireless internet of things, IEEE Wireless Communications 25 (6) (2018) 91–97.

[50] Y.-H. Zhang, Y.-J. Gong, H.-X. Zhang, T.-L. Gu, J. Zhang, Toward fast niching evolutionary algorithms: A locality sensitive hashing-based ap-

proach, IEEE Transactions on Evolutionary Computation 21 (3) (2017) 347–362.

[51] X. Sun, H. Wang, J. Li, T. M. Truta, Enhanced p-sensitive k-anonymity models for privacy preserving data publishing, Transactions on Data Privacy 1 (2) (2008) 53–66.

[52] M. Li, X. Sun, H. Wang, Y. Zhang, J. Zhang, Privacy-aware access control with trust management in web service, World Wide Web 14 (4) (2011) 407–430.

[53] M. E. Kabir, H. Wang, Conditional purpose based access control model for privacy protection, in: Proceedings of the Twentieth Australasian Conference on Australasian Database-Volume 92, Australian Computer Society, Inc., 2009, pp. 135–142.

[54] X. Sun, H. Wang, J. Li, Y. Zhang, Satisfying privacy requirements before data anonymization, The Computer Journal 55 (4) (2012) 422–437.

[55] W. Shi, W.-N. Chen, Y. Lin, T. Gu, S. Kwong, J. Zhang, An adaptive estimation of distribution algorithm for multi-policy insurance investment planning, IEEE Transactions on Evolutionary Computation (2017).

[56] R. S. Ross, M. McEvilley, J. C. Oren, Systems security engineering: Considerations for a multidisciplinary approach in the engineering of trustworthy secure systems, Tech. rep., National Institute of Standards and Technology (2018).

[57] J. Shu, X. Jia, K. YANG, H. Wang, Privacy-preserving task recommendation services for crowdsourcing, IEEE Transactions on Services Computing (2018).

[58] Y.-H. Jia, W.-N. Chen, T. Gu, H. Zhang, H. Yuan, S. Kwong, J. Zhang, Distributed cooperative co-evolution with adaptive computing resource allocation for large scale optimization, IEEE Transactions on Evolutionary Computation (2018).

[59] Y. Shen, T. Zhang, Y. Wang, H. Wang, X. Jiang, Microthings: A generic iot architecture for flexible data aggregation and scalable service cooperation, IEEE Communications Magazine 55 (9) (2017) 86–93.

[60] W. Rafique, M. Khan, X. Zhao, N. Sarwar, W. Dou, A blockchain-based framework for information security in intelligent transportation systems, in: I. S. Bajwa, T. Sibalija, D. N. A. Jawawi (Eds.), Intelligent Technologies and Applications, 2020, pp. 53–66.

[61] W. Rafique, M. Khan, N. Sarwar, W. Dou, A security framework to protect edge supported software defined internet of things infrastructure, in: International Conference on Collaborative Computing: Networking, Applications and Worksharing, Springer, 2019, pp. 71–88.

[62] S. Yu, G. Zhao, W. Dou, S. James, Predicted packet padding for anonymous web browsing against traffic analysis attacks, IEEE Transactions on Information Forensics and Security 7 (4) (2012) 1381–1393.

[63] C. Juan, C. Jenny, F. Juan, Security in sdn: A comprehensive survey, Journal of Network and Computer Applications 159 (2020) 102–595.

[64] P. Maninder, B. Abhinav, New-flow based ddos attacks in sdn: Taxonomy, rationales, and research challenges, Computer Communications 154 (2020) 509 – 527.

[65] A. Abdou, P. C. van Oorschot, T. Wan, Comparative analysis of control plane security of sdn and conventional networks, IEEE Communications Surveys Tutorials 20 (4) (2018) 3542–3559.

[66] S. Scott-Hayward, S. Natarajan, S. Sezer, A survey of security in software defined networks, IEEE Communications Surveys & Tutorials 18 (1) (2016) 623–654.

[67] D. B. Rawat, S. R. Reddy, Software defined networking architecture, security and energy efficiency: A survey, IEEE Communications Surveys Tutorials 19 (1) (2017) 325–346.

[68] S. Dong, K. Abbas, R. Jain, A survey on distributed denial of service (ddos) attacks in sdn and cloud computing environments, IEEE Access 7 (2019) 80813–80828.

[69] T. Dargahi, A. Caponi, M. Ambrosin, G. Bianchi, M. Conti, A survey on the security of stateful sdn data planes, IEEE Communications Surveys & Tutorials 19 (3) (2017) 1701–1725.

[70] R. U. Rasool, H. Wang, W. Rafique, J. Yong, J. Cao, A study on securing software defined networks, in: International Conference on Web Information Systems Engineering, Springer, 2017, pp. 479–489.

[71] D. B. Rawat, S. R. Reddy, Software defined networking architecture, security and energy efficiency: A survey, IEEE Communications Surveys & Tutorials 19 (1) (2017) 325–346.

[72] M. ALAN, G. GUILBERT, T. ARCHIE, How the cyberattack on spamhaus unfolded, New York Times (2013).

[73] A. Studer, A. Perrig, The coremelt attack, in: Proc. Springer International Conference ESORICS, 2009, Vol. 5789, 2009, pp. 37–52.

[74] S. Yu, S. Guo, I. Stojmenovic, Fool me if you can: Mimicking attacks and anti-attacks in cyberspace, IEEE Transactions on Computers 64 (1) (2013) 139–151.

[75] S. Yu, Y. Tian, S. Guo, D. O. Wu, Can we beat ddos attacks in clouds?, IEEE Transactions on Parallel and Distributed Systems 25 (9) (2013) 2245–2254.

[76] S. Yu, Big privacy: Challenges and opportunities of privacy study in the age of big data, IEEE access 4 (2016) 2751–2763.

[77] S. Yu, W. Zhou, R. Doss, W. Jia, Traceback of ddos attacks using entropy variations, IEEE Transactions on Parallel and Distributed Systems 22 (3) (2010) 412–425.

[78] G. Yang, H. Hosseini, D. Sahabandu, A. Clark, J. Hespanha, R. Poovendran, Modeling and mitigating the coremelt attack, American Control Conference. (2017).

[79] A. P. Athreya, X. Wang, Y. S. Kim, Y. Tian, P. Tague, Resistance is not futile: Detecting ddos attacks without packet inspection, Information Security Applications: 14th International Workshop 4 (379) (2013) 15.

[80] S. Yu, W. Zhou, S. Guo, M. Guo, A feasible ip traceback framework through dynamic deterministic packet marking, IEEE Transactions on Computers 65 (5) (2015) 1418–1427.

[81] S. A. Inc, Highleyman, History's largest ddos attack, url-http://www.availabilitydigest.com/publicarticles/0804/spamhaus.pdf , accessed: 30-09-2017 (2013).

[82] C. Yoon, S. Lee, H. Kang, T. Park, S. Shin, V. Yegneswaran, P. Porras, G. Gu, Flow wars: Systemizing the attack surface and defenses in software-defined networks, IEEE/ACM Transactions on Networking 25 (6) (2017) 3514–3530.

[83] Y. Choi, Implementation of content-oriented networking architecture (cona): a focus on ddos countermeasure, in: Proc of 1st European NetFPGA Developers Workshop, 2010, pp. 1–10.

[84] S. GAO, Z. PENG, B. XIAO, A. HU, K. REN, Flooddefender: Protecting data and control plane resources under sdn-aimed dos attacks, in: IEEE Conference on Computer Communications, INFOCOM, 2017, pp. 1–9.

[85] W. Xia, P. Zhao, Y. Wen, H. Xie, A survey on data center networking (dcn): Infrastructure and operations, IEEE communications surveys & tutorials 19 (1) (2017) 640–656.

[86] H. Wang, L. Xu, G. Gu, Of-guard: A dos attack prevention extension in software-defined networks, The Open Network Summit (ONS) (2014).

[87] P. Porras, S. Shin, V. Yegneswaran, M. Fong, M. Tyson, G. Gu, A security enforcement kernel for openflow networks, in: Proceedings of the first workshop on Hot topics in software defined networks, 2012, pp. 121–126.

[88] R. Sherwood, G. Gibb, K.-K. Yap, G. Appenzeller, M. Casado, N. McKeown, G. Parulkar, Flowvisor: A network virtualization layer, OpenFlow Switch Consortium, Tech. Rep 1 (2009) 132.

[89] C. Li, J. Yang, Z. Wang, F. Li, Y. Yang, A lightweight ddos flooding attack detection algorithm based on synchronous long flows, in: Proc. IEEE Global Communications Conference, (GLOBECOM), 2013, pp. 1–6.

[90] M. Ambrosin, M. Conti, F. De Gaspari, R. Poovendran, Lineswitch: Tackling control plane saturation attacks in software-defined networking, IEEE/ACM Transactions on Networking 25 (10) (2016) 14.

[91] A. Tootoonchian, Y. Ganjali, Hyperflow: A distributed control plane for openflow, in: Proceedings of the 2010 Internet Network Management Conference on Research on Enterprise Networking, 2010, pp. 3–3.

[92] S. Matsumoto, S. Hitz, A. Perrig, Fleet: Defending sdns from malicious administrators, in: Proc. ACM 3rd International workshop on Hot topics in software defined networking, 2013, pp. 103–108.

[93] A. Zaalouk, R. Khondoker, R. Marx, K. Bayarou, Orchsec: An orchestrator-based architecture for enhancing network-security using network monitoring and sdn control functions, in: Proc. IEEE Network Operations and Management Symposium (NOMS), 2014, 2014, pp. 1–9.

[94] S. Shirali-Shahreza, Y. Ganjali, Flexam: flexible sampling extension for monitoring and security applications in openflow, in: Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking, 2013, pp. 167–168.

[95] S. Shin, V. Yegneswaran, P. Porras, G. Gu, Avant-guard: Scalable and vigilant switch flow management in software-defined networks, in: Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security, 2013, pp. 413–424.

[96] S. Shenker, M. Casado, T. Koponen, N. McKeown, The future of networking, and the past of protocols, Open Networking Summit 20 (2011) 1–30.

[97] T. Ohanian, Moving toward zero infrastructure broadcasting, SMPTE Motion Imaging Journal 125 (8) (2016) 49–59.

[98] C. Röpke, T. Holz, Sdn rootkits: Subverting network operating systems of software-defined networks, in: Proc. Springer International Workshop on Recent Advances in Intrusion Detection, 2013, pp. 339–356.

[99] J. Sonchack, J. M. Smith, A. J. Aviv, E. Keller, Enabling practical software-defined networking security applications with ofx, in: Proc. of the Network and Distributed System Security Symposium, (NDSS), Vol. 16, 2016, pp. 1–15.

[100] A. Khurshid, W. Zhou, M. Caesar, P. Godfrey, Veriflow: Verifying network-wide invariants in real time, in: Proc. 1st ACM workshop on Hot topics in software defined networks, 2012, pp. 49–54.

[101] A. Martin, Dynamic filtering for sdn api calls across a security boundary (2016).

[102] H.-z. Wang, P. Zhang, L. Xiong, X. Liu, C.-c. Hu, A secure and high-performance multi-controller architecture for software-defined networking, Frontiers of Information Technology & Electronic Engineering 17 (7) (2016) 634–646.

[103] The open networking foundation: Sdn architecture overview (Accessed on May 29, 2020).

URL https://www.opennetworking.org/images/stories/downloads/sdn-resources/technical-reports/TR_SDN-ARCH-Overview-1.1-11112014.02.pdf

[104] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, J. Turner, Openflow: enabling innovation in campus networks, ACM SIGCOMM Computer Communication Review 38 (2) (2008) 69–74.

[105] S. Khan, A. Gani, A. W. A. Wahab, A. Abdelaziz, M. A. Bagiwa, Fml: A novel forensics management layer for software defined networks, in: Proc. IEEE 6th International Conference on Cloud System and Big Data Engineering, (Confluence), 2016, pp. 619–623.

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

[107] Y. Cai, F. R. Yu, C. Liang, B. Sun, Q. Yan, Software-defined device-to-device (d2d) communications in virtual wireless networks with imperfect network state information (nsi), IEEE Transactions on Vehicular Technology 65 (9) (2016) 7349–7360.

[108] A. Csoma, B. Sonkoly, L. Csikor, F. Németh, A. Gulyas, W. Tavernier, S. Sahhaf, Escape: Extensible service chain prototyping environment using mininet, click, netconf and pox, in: ACM SIGCOMM Computer Communication Review, 2014, Vol. 44, 2014, pp. 125–126.

[109] M. Brandt, R. Khondoker, R. Marx, K. Bayarou, Security analysis of software defined networking protocols—openflow, of-config and ovsdb, in: Proc. IEEE 5th International Conference on Communications and Electronics,(ICCE), 2014, pp. 1–6.

[110] Y. Qian, W. You, K. Qian, Openflow flow table overflow attacks and countermeasures, in: Proc. IEEE European Conference on Networks and Communications, (EuCNC), 2016, pp. 205–209.

[111] T. Xu, D. Gao, P. Dong, C. H. Foh, H. Zhang, Mitigating the table-overflow attack in software-defined networking, IEEE Transactions on Network and Service Management 14 (4) (2017) 1086–1097.

[112] L. Zhang, S. Wang, S. Xu, R. Lin, H. Yu, Timeoutx: An adaptive flow table management method in software defined networks, in: Proc. IEEE Global Communications Conference, (GLOBECOM), 2015, pp. 1–6.

[113] B. Leng, L. Huang, X. Wang, H. Xu, Y. Zhang, A mechanism for reducing flow tables in software defined network, in: Proc. IEEE Conference on Communications, (ICC), 2015, pp. 5302–5307.

[114] I. Ahmad, S. Namal, M. Ylianttila, A. Gurtov, Security in software defined networks: A survey, IEEE Communications Surveys & Tutorials 17 (4) (2015) 2317–2346.

[115] K. Giotis, C. Argyropoulos, G. Androulidakis, D. Kalogeras, V. Maglaris, Combining openflow and sflow for an effective and scalable anomaly detection and mitigation mechanism on sdn environments, Computer Networks 62 (2014) 122–136.

[116] R. Kandoi, M. Antikainen, Denial-of-service attacks in openflow sdn networks, in: Proc. IFIP/IEEE Internatioal Symposium on Integrated Network Management (IM), 2015, 2015, pp. 1322–1326.

[117] H. T. N. Tri, K. Kim, Assessing the impact of resource attack in software defined network, in: Proc. IEEE International Conference on Information Networking, (ICOIN), 2015, pp. 420–425.

[118] D. Smyth, S. McSweeney, D. O'Shea, V. Cionca, Detecting link fabrication attacks in software-defined networks, in: Proc. IEEE 26th Internatioal

Conference on Computer Communication and Networks, (ICCCN), 2017, pp. 1–8.

[119] Y. Xie, S.-Z. Yu, A novel model for detecting application layer ddos attacks, in: Proc. IEEE First International Multi-Symposiums onComputer and Computational Sciences, (IMSCCS), Vol. 2, 2006, pp. 56–63.

[120] W. Zhou, W. Jia, S. Wen, Y. Xiang, W. Zhou, Detection and defense of application-layer ddos attacks in backbone web traffic, Future Generation Computer Systems 38 (2014) 36–46.

[121] S. M. Mousavi, M. St-Hilaire, Early detection of ddos attacks against sdn controllers, in: Proc. IEEE International Conference on Computing, Networking and Communications, (ICNC), 2015, pp. 77–81.

[122] C. J. Bernardos, A. De La Oliva, P. Serrano, A. Banchs, L. M. Contreras, H. Jin, J. C. Zuniga, An architecture for software defined wireless networking, IEEE wireless communications 21 (3) (2014) 52–61.

[123] Y. Zhang, Y. Shen, H. Wang, Y. Zhang, X. Jiang, On secure wireless communications for service oriented computing, IEEE Transactions on Services Computing 11 (2) (2018) 318–328.

[124] Y. Shi, Y. T. Hou, J. Liu, S. Kompella, Bridging the gap between protocol and physical models for wireless networks, IEEE Transactions on Mobile Computing 12 (7) (2013) 1404–1416.

[125] J. Schulz-Zander, C. Mayer, B. Ciobotaru, S. Schmid, A. Feldmann, Unified programmability of virtualized network functions and software-defined wireless networks, IEEE Transactions on Network and Service Management (2017).

[126] H. Wang, H. Tang, S. Zhang, Joint optimization in software defined wireless networks with network coded opportunistic routing, in: Proc. IEEE 14th International Conference on Mobile Ad Hoc and Sensor Systems (MASS), 2017, 2017, pp. 298–302.

[127] C.-F. Liu, S. Samarakoon, M. Bennis, H. V. Poor, Fronthaul-aware software-defined wireless networks: Resource allocation and user scheduling, IEEE Transactions on Wireless Communications (2017).

[128] C. Liang, Y. He, F. R. Yu, N. Zhao, Enhancing qoe-aware wireless edge caching with software-defined wireless networks, IEEE Transactions on Wireless Communications 16 (10) (2017) 6912–6925.

[129] P. Graubner, M. Sommer, M. Hollick, B. Freisleben, Dynamic role assignment in software-defined wireless networks, in: Proc. IEEE Symposium on Computers and Communications, SCC), 2017, pp. 760–766.

[130] B. O. Kahjogh, G. Bernstein, Energy and latency optimization in software defined wireless networks, in: Proc. IEEE 9th Internatioal Conference on Ubiquitous and Future Networks (ICUFN), 2017, pp. 714–719.

[131] X. Liu, A. Liu, Z. Li, Adaptive broadcast times for program codes in software defined wireless networks, in: Proc. IEEE International Conference on Mobile Ad-Hoc and Sensor Networks, (MSN), 2016, pp. 405–408.

[132] M. J. Abdel-Rahman, E. A. Mazied, A. MacKenzie, S. Midkiff, M. R. Rizk, M. El-Nainay, On stochastic controller placement in software-defined wireless networks, in: Proc. IEEE Conference on Wireless Communications and Networking Conference, (WCNC), 2017, pp. 1–6.

[133] K. Mizuyama, Y. Taenaka, K. Tsukamoto, Estimation based adaptable flow aggregation method for reducing control traffic on software defined wireless networks, in: Proc. IEEE International Conference on Pervasive Computing and Communications Workshops, (PerCom Workshops), 2017, pp. 363–368.

[134] B. Cao, Y. Li, C. Wang, G. Feng, S. Qin, Y. Zhou, Resource allocation in software defined wireless networks, IEEE Network 31 (1) (2017) 44–51.

[135] M. Feng, S. Mao, T. Jiang, Enhancing the performance of future wireless networks with software-defined networking, Frontiers of Information Technology & Electronic Engineering 17 (2016) 606–619.

[136] X. Zhang, Z. Xu, L. Fan, S. Yu, Y. Qu, Near-optimal energy-efficient algorithm for virtual network function placement, IEEE Transactions on Cloud Computing (2019).

[137] N. Zhang, N. Cheng, N. Lu, H. Zhou, J. W. Mark, X. Shen, Risk-aware cooperative spectrum access for multi-channel cognitive radio networks, IEEE Journal on Selected Areas in Communications 32 (3) (2014) 516–527.

[138] Y. T. Hou, Y. Shi, H. D. Sherali, Optimal spectrum sharing for multi-hop software defined radio networks, in: Proc. IEEE 26th International Conference on Computer Communications, INFOCOM, 2007, pp. 1–9.

[139] H. Daojing, C. Sammy, G. Mohsen, Securing software defined wireless networks, IEEE Communications Magazine 0163-6804 (0163-6804/16) (2016) 1–6.

[140] I. Ahmad, S. N. Karunarathna, M. Ylianttila, A. Gurtov, Load balancing in software defined mobile networks, Software Defined Mobile Networks (SDMN): Beyond LTE Network Architecture (2015) 225–245.

[141] H. Selvi, S. Güner, G. Gür, F. Alagöz, The controller placement problem in software defined mobile networks (sdmn), Software Defined Mobile Networks (SDMN): Beyond LTE Network Architecture (2015) 129–147.

[142] R. G. L. Narayanan, Software defined networks for mobile application services, Software Defined Mobile Networks (SDMN): Beyond LTE Network Architecture (2015) 209–224.

[143] C. Liang, F. R. Yu, Enhancing mobile edge caching with bandwidth provisioning in software-defined mobile networks, in: Proc. IEEE International Conference on Communications, (ICC), 2017, pp. 1–6.

[144] M. Liyanage, M. Ylianttila, A. Gurtov, Securing the control channel of software-defined mobile networks, in: Proc. IEEE 15th International Symposium on World of Wireless, Mobile and Multimedia Networks, (WoWMoM), 2014, pp. 1–6.

[145] C. Liang, F. R. Yu, Bandwidth provisioning in cache-enabled software-defined mobile networks: A robust optimization approach, in: IEEE 84th International Vehicular Technology Conference, (VTC-Fall), 2016, pp. 1–5.

[146] M. Liyanage, A. B. Abro, M. Ylianttila, A. Gurtov, Opportunities and challenges of software-defined mobile networks in network security, IEEE Security & Privacy 14 (4) (2016) 34–44.

[147] I. Ahmad, M. Liyanage, S. Namal, M. Ylianttila, A. Gurtov, M. Eckert, T. Bauschert, Z. Faigl, L. Bokor, E. Saygun, New concepts for traffic, resource and mobility management in software-defined mobile networks, in: Proc. 12th IEEE Annual Conference on Wireless On-demand Network Systems and Services, (WONS), 2016, pp. 1–8.

[148] M. Liyanage, I. Ahmed, M. Ylianttila, J. L. Santos, R. Kantola, O. L. Perez, M. U. Itzazelaia, E. M. de Oca, A. Valtierra, C. Jimenez, Security for future software defined mobile networks, in: Proc. IEEE 9th International Conference on Next Generation Mobile Applications, Services and Technologies, 2015, 2015, pp. 256–264.

[149] L. J. Chaves, V. M. Eichemberger, I. C. Garcia, E. R. M. Madeira, Integrating openflow to lte: Some issues toward software-defined mobile networks, in: Proc. IEEE 7th International Conference on New Technologies, Mobility and Security, (NTMS), 2015, pp. 1–5.

[150] R. Riggio, M. K. Marina, T. Rasheed, Interference management in software-defined mobile networks, in: Proc. IEEE International Symposium onIntegrated Network Management, (IM), 2015, pp. 626–632.

[151] I. Katzela, M. Naghshineh, Channel assignment schemes for cellular mobile telecommunication systems: A comprehensive survey, IEEE personal communications 3 (3) (1996) 10–31.

[152] F.-Y. Wang, Parallel control and management for intelligent transportation systems: Concepts, architectures, and applications, IEEE Transactions on Intelligent Transportation Systems 11 (3) (2010) 630–638.

[153] J. Chung, G. Gonzalez, I. Armuelles, T. Robles, R. Alcarria, A. Morales, Experiences and challenges in deploying openflow over real wireless mesh networks, IEEE Latin America Transactions 11 (3) (2013) 955–961.

[154] W. Zhao, J. Xie, Imex: intergateway cross-layer handoffs in internet-based infrastructure wireless mesh networks, IEEE Transactions on Mobile Computing 11 (10) (2012) 1585–1600.

[155] F. Yang, V. Gondi, J. O. Hallstrom, K.-C. Wang, G. Eidson, Openflow-based load balancing for wireless mesh infrastructure, in: Proc. IEEE 11th International Conference on Consumer Communications and Networking Conference (CCNC), 2014, 2014, pp. 444–449.

[156] M. K. Marina, S. R. Das, A. P. Subramanian, A topology control approach for utilizing multiple channels in multi-radio wireless mesh networks, Computer networks 54 (2) (2010) 241–256.

[157] K. Sood, S. Liu, S. Yu, Y. Xiang, Dynamic access point association using software-defined networking, IEEE Telecommunication Networks and Applications Conference (IEEE ITNAC) (2015) 6.

[158] P. Dely, A. Kassler, N. Bayer, Openflow for wireless mesh networks, in: Proc. IEEE 20th Internatioal Conference on Computer Communications and Networks, (ICCCN), 2015, pp. 1–6.

[159] A. Detti, C. Pisa, S. Salsano, N. Blefari-Melazzi, Wireless mesh software defined networks (wmsdn), in: Proc. IEEE 9th Internatioal Conference

on Wireless and Mobile Computing, Networking and Communications, (WiMob), 2013, pp. 89–95.

[160] H. Huang, P. Li, S. Guo, W. Zhuang, Software-defined wireless mesh networks: architecture and traffic orchestration, IEEE network 29 (4) (2015) 24–30.

[161] J. Chen, B. Liu, H. Zhou, Q. Yu, G. Lin, X. Shen, Qos-driven efficient client association in high-density software defined wlan, IEEE Transactions on Vehicular Technology (2017).

[162] A. Amelyanovich, M. Shpakov, A. Muthanna, M. Buinevich, A. Vladyko, Centralized control of traffic flows in wireless lans based on the sdn concept, in: IEEE Conference on Systems of Signal Synchronization, Generating and Processing in Telecommunications, (SINKHROINFO), 2017, pp. 1–5.

[163] X. Sang, Q. Wu, H. Li, Client-network collaborative load balancing mechanism for wlan based on sdn and 802.11 u, in: Proc. IEEE 13th International Conference on Wireless Communications and Mobile Computing Conference, (IWCMC), 2017, pp. 506–511.

[164] M.-C. Chan, C. Chen, J.-X. Huang, T. Kuo, L.-H. Yen, C.-C. Tseng, Opennet: A simulator for software-defined wireless local area network, 2014 IEEE Wireless Communications and Networking Conference (WCNC) 4 (10) (2014) 5.

[165] M. Antikainen, T. Aura, M. Särelä, Spook in your network: Attacking an sdn with a compromised openflow switch, in: Proc. Springer Nordic Conference on Secure IT Systems, 2014, pp. 229–244.

[166] J. Leng, Y. Zhou, J. Zhang, C. Hu, An inference attack model for flow table capacity and usage: Exploiting the vulnerability of flow table overflow in software-defined network, arXiv preprint arXiv:1504.03095 (2015).

[167] S. Shin, G. Gu, Attacking software-defined networks: A first feasibility study, in: Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking, 2013, pp. 165–166.

[168] T. Chin, X. Mountrouidou, X. Li, K. Xiong, Selective packet inspection to detect dos flooding using software defined networking (sdn), in: Proc. IEEE 35th Internatioal Conference on Distributed Computing Systems Workshops, (ICDCSW), 2015, pp. 95–99.

[169] N.-N. Dao, J. Park, M. Park, S. Cho, A feasible method to combat against ddos attack in sdn network, in: Proc. IEEE International Conference on Information Networking, (ICOIN), 2015, pp. 309–311.

[170] T. V. Phan, N. K. Bao, M. Park, Distributed-som: A novel performance bottleneck handler for large-sized software-defined networks under flooding attacks, Journal of Network and Computer Applications 91 (2017) 14–25.

[171] H. Wang, L. Xu, G. Gu, Floodguard: A dos attack prevention extension in software-defined networks, in: Proc. IEEE/IFIP 45th AnnualInternational Conference onDependable Systems and Networks, (DSN), 2015, pp. 239–250.

[172] D. Kreutz, F. Ramos, P. Verissimo, Towards secure and dependable software-defined networks, ACM SIGCOMM, HotSDN13 5 (2013).

[173] K. Cheng, L. Wang, Y. Shen, H. Wang, Y. Wang, X. Jiang, H. Zhong, Secure k-nn query on encrypted cloud data with multiple keys, IEEE Transactions on Big Data (2017).

[174] G. B. A. Solutions, Global computing facilities, url-http://www.gartner.com/newsroom/id/3666917 , accessed: 2017-09-30 (2017).

[175] N. W. F. IDG, Software defined networking news and trends, urlhttp://www.networkworld.com/article/2981667/cisco-

subnet/software-defined-networking-trend-or-technology-movement.html , accessed: 2017-09-30 (2017).

[176] S. A. Mehdi, J. Khalid, S. A. Khayam, Revisiting traffic anomaly detection using software defined networking, in: Proc. Springer International Workshop on Recent Advances in Intrusion Detection, 2011, pp. 161–180.

[177] S. Shin, P. A. Porras, V. Yegneswaran, M. W. Fong, G. Gu, M. Tyson, Fresco: Modular composable security services for software-defined networks, in: Proc. of the Network and Distributed System Security Symposium, (NDSS), 2013, pp. 1–6.

[178] S. Lim, J. Ha, H. Kim, Y. Kim, S. Yang, A sdn-oriented ddos blocking scheme for botnet-based attacks, in: Proc. IEEE 6th International Conference on Ubiquitous and Future Networks, (ICUFN), 2014, pp. 63–68.

[179] Q. Yan, F. R. Yu, Distributed denial of service attacks in software-defined networking with cloud computing, IEEE Communications Magazine 53 (4) (2015) 52–59.

[180] A. Akhunzada, A. Gani, N. B. Anuar, A. Abdelaziz, M. K. Khan, A. Hayat, S. U. Khan, Secure and dependable software defined networks, Journal of Network and Computer Applications 61 (2016) 199–221.

[181] A. Blenk, A. Basta, M. Reisslein, W. Kellerer, Survey on network virtualization hypervisors for software defined networking, IEEE Communications Surveys & Tutorials 18 (1) (2016) 655–685.

[182] A. Kalliola, K. Lee, H. Lee, T. Aura, Flooding ddos mitigation and traffic management with software defined networking, in: Proc. IEEE 4th International Conference on Cloud Networking, (CloudNet), 2015, pp. 248–254.

[183] X. Ma, J. Li, Y. Tang, B. An, X. Guan, Protecting internet infrastructure against link flooding attacks: A techno-economic perspective, Information Sciences (2018).

[184] B. Feng, H. Zhang, H. Zhou, S. Yu, Locator/identifier split networking: A promising future internet architecture, IEEE Communications Surveys & Tutorials 19 (4) (2017) 2927–2948.

[185] N. Z. Bawany, J. A. Shamsi, K. Salah, Ddos attack detection and mitigation using sdn: Methods, practices, and solutions, Arab J Sci Eng 42 (5) (2017) 425–441.

[186] J. Wang, R. Wen, J. Li, F. Yan, B. Zhao, F. Yu, Detecting and mitigating target link-flooding attacks using sdn, IEEE Transactions on Dependable and Secure Computing 16 (6) (2018) 944–956.

[187] L. Wang, Q. Li, Y. Jiang, X. Jia, J. Wu, Woodpecker: Detecting and mitigating link-flooding attacks via sdn, Computer Networks 147 (2018) 1–13.

[188] L. Xue, X. Luo, E. W. Chan, X. Zhan, Towards detecting target link flooding attack, in: Proc. 28th Large Installation System Administration Conference, LISA, 2014, pp. 1–6.

[189] J. Zheng, Q. Li, G. Gu, J. Cao, D. K. Yau, J. Wu, Realtime ddos defense using cots sdn switches via adaptive correlation analysis, IEEE Transactions on Information Forensics and Security 13 (7) (2018) 1838–1853.

[190] R. Meier, P. Tsankov, V. Lenders, L. Vanbever, M. Vechev, Nethide: secure and practical network topology obfuscation, in: 27th {USENIX} Security Symposium ({USENIX} Security 18), 2018, pp. 693–709.

[191] C. Liaskos, S. Ioannidis, Network topology effects on the detectability of crossfire attacks, IEEE Transactions on Information Forensics and Security 13 (7) (2018) 1682–1695.

[192] L. Xue, X. Ma, X. Luo, E. W. Chan, T. T. Miu, G. Gu, Linkscope: Toward detecting target link flooding attacks, IEEE Transactions on Information Forensics and Security 13 (10) (2018) 2423–2438.

[193] J. Kim, S. Shin, Software-defined honeynet:towards mitigating link flooding attacks, 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops 4 (09) (2017) 2.

[194] W. Rafique, X. He, Z. Liu, Y. Sun, W. Dou, Cfadefense: A security solution to detect and mitigate crossfire attacks in software-defined iot-edge infrastructure, in: 2019 IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems (HPCC/SmartCity/DSS), IEEE, 2019, pp. 500–509.

[195] A. Aydeger, M. H. Manshaei, M. A. Rahman, K. Akkaya, Strategic defense against stealthy link flooding attacks: A signaling game approach, arXiv preprint arXiv:1912.10073 (2019).

[196] X. Ma, B. An, M. Zhao, X. Luo, L. Xue, Z. Li, T. Miu, X. Guan, Randomized security patrolling for link flooding attack detection, IEEE Transactions on Dependable and Secure Computing (2019, In Pres) 1–1.

[197] J. Xing, J. Cai, B. Zhou, C. Wu, A deep convnet-based countermeasure to mitigate link flooding attacks using software-defined networks, in: 2019 IEEE Symposium on Computers and Communications (ISCC), 2019, pp. 1–6.

[198] N. Ravi, S. M. Shalinie, D. D. J. Theres, Balance: Link flooding attack detection and mitigation via hybrid-sdn, IEEE Transactions on Network and Service Management (2020) 1–1.

[199] Y.-H. Chen, P.-T. Jan, C.-N. Lai, C. Huang, C.-H. Chang, Y.-C. Huang, Detecting linking flooding attacks using deep convolution network, in: Proceedings of the 2020 the 3rd International Conference on Computers in Management and Business, 2020, pp. 70–74.

[200] J. M. Vidal, A. L. S. Orozco, L. J. G. Villalba, Adaptive artificial immune

networks for mitigating dos flooding attacks, Swarm and Evolutionary Computation (2017).

[201] D. Boro, D. K. Bhattacharyya, Dyprosd: a dynamic protocol specific defense for high-rate ddos flooding attacks, Microsystem Technologies 23 (3) (2017) 593–611.

[202] T. V. Phan, T. Van Toan, D. Van Tuyen, T. T. Huong, N. H. Thanh, Openflowsia: An optimized protection scheme for software-defined networks from flooding attacks, in: Proc. IEEE 6th International Conference on Communications and Electronics (ICCE), 2016, 2016, pp. 13–18.

[203] V. A. Foroushani, A. N. Zincir-Heywood, Tdfa: Traceback-based defense against ddos flooding attacks, in: Proc. IEEE 28th Internatioal Conference on Advanced Information Networking and Applications, (AINA), 2014, pp. 597–604.

[204] Q. Yang, W.-N. Chen, J. Da Deng, Y. Li, T. Gu, J. Zhang, A level-based learning swarm optimizer for large-scale optimization, IEEE Transactions on Evolutionary Computation 22 (4) (2018) 578–594.

[205] Z.-H. Zhan, X.-F. Liu, H. Zhang, Z. Yu, J. Weng, Y. Li, T. Gu, J. Zhang, Cloudde: A heterogeneous differential evolution algorithm and its distributed cloud version, IEEE Transactions on Parallel and Distributed Systems 28 (3) (2017) 704–716.

[206] S. Yu, M. Liu, W. Dou, X. Liu, S. Zhou, Networking for big data: A survey, IEEE Communications Surveys & Tutorials 19 (1) (2016) 531–549.

[207] Z.-J. Wang, Z.-H. Zhan, Y. Lin, W.-J. Yu, H.-Q. Yuan, T.-L. Gu, S. Kwong, J. Zhang, Dual-strategy differential evolution with affinity propagation clustering for multimodal optimization problems, IEEE Transactions on Evolutionary Computation 22 (6) (2018) 894–908.

[208] J. Chen, K. Li, Z. Tang, K. Bilal, S. Yu, C. Weng, K. Li, A parallel random forest algorithm for big data in a spark cloud computing environment,

IEEE Transactions on Parallel and Distributed Systems 28 (4) (2016) 919–933.

[209] Y.-J. Gong, J. Zhang, Y. Zhou, Learning multimodal parameters: A bare-bones niching differential evolution approach, IEEE transactions on neural networks and learning systems 29 (7) (2018) 2944–2959.

[210] Y.-J. Gong, J.-J. Li, Y. Zhou, Y. Li, H. S.-H. Chung, Y.-H. Shi, J. Zhang, Genetic learning particle swarm optimization, IEEE transactions on cybernetics 46 (10) (2016) 2277–2290.

[211] J. Mirkovic, T. V. Benzel, T. Faber, R. Braden, J. T. Wroclawski, S. Schwab, The deter project: Advancing the science of cyber security experimentation and test, in: 2010 IEEE International Conference on Technologies for Homeland Security (HST), IEEE, 2010, pp. 1–7.

[212] S. Knight, H. X. Nguyen, N. Falkner, R. Bowden, M. Roughan, The internet topology zoo, IEEE Journal on Selected Areas in Communications 29 (9) (2011) 1765–1775.

[213] M. Berman, C. Elliott, L. Landweber, Geni: Large-scale distributed infrastructure for networking and distributed systems research, in: 2014 IEEE Fifth International Conference on Communications and Electronics (ICCE), IEEE, 2014, pp. 156–161.

[214] M. Suñé, L. Bergesio, H. Woesner, T. Rothe, A. Köpsel, D. Colle, B. Puype, D. Simeonidou, R. Nejabati, M. Channegowda, et al., Design and implementation of the ofelia fp7 facility: The european openflow testbed, Computer Networks 61 (2014) 132–150.

[215] OneLab, OneLab Future Internet Testbeds, accessed: 2020-05-28 (2011). URL `onelab.eu`

[216] aarnet, Virtualisation Testbed for NFV/SDN Environment, accessed: 2020-05-28 (2020).
URL `www.aarnet.edu.au`

[217] Fraunhofer Fokus, Virtualisation Testbed for NFV/SDN Environment, accessed: 2020-05-28 (2020).
URL `www.opensdncore.org`

[218] E. Kabir, J. Hu, H. Wang, G. Zhuo, A novel statistical technique for intrusion detection systems, Future Generation Computer Systems 79 (2018) 303–318.

[219] H. Wang, X. Yi, E. Bertino, L. Sun, Protecting outsourced data in cloud computing through access management, Concurrency and computation: Practice and Experience 28 (3) (2016) 600–615.

[220] J. Zhang, H. Li, X. Liu, Y. Luo, F. Chen, H. Wang, L. Chang, On efficient and robust anonymization for privacy protection on massive streaming categorical information, IEEE Transactions on Dependable and Secure Computing 14 (5) (2015) 507–520.

[221] E. Kabir, A. Mahmood, H. Wang, A. Mustafa, Microaggregation sorting framework for k-anonymity statistical disclosure control in cloud computing, IEEE Transactions on Cloud Computing (2015).

[222] R. U. Rasool, U. Ashraf, K. Ahmed, H. Wang, W. Rafique, Z. Anwar, Cyberpulse: A machine learning based link flooding attack mitigation system for software defined networks, IEEE Access 7 (2019) 34885–34899.

[223] R. ur Rasool, M. Najam, H. F. Ahmad, H. Wang, Z. Anwar, A novel json based regular expression language for pattern matching in the internet of things, Journal of Ambient Intelligence and Humanized Computing (2018) 1–19.

[224] W. Rafique, L. Qi, I. Yaqoob, I. Muhammad, R. ur Rasool, W. Dou, Complementing iot services through software defined networking and edge computing: A comprehensive survey, IEEE Communications Surveys and Tutorials, 2020, In Press (2020, In Press).