Empowering DDoS Attack Mitigation with Programmable Switches

Xiang Chen, Hongyan Liu, Dong Zhang, Qun Huang, Haifeng Zhou, Chunming Wu, and Qiang Yang, *Senior Member, IEEE*

***Abstract*—Distributed denial-of-service (DDoS) attacks have long been the most severe and destructive attack on modern networks. Some solutions place several middleboxes that run security-oriented network functions (SNFs) in the network to defend against DDoS attacks. However, middleboxes are proprietary and fixed-function, making them costly and inflexible when handling attack dynamics. Another class of solutions exploits the capability of software-defined networking (SDN) and network function virtualization (NFV) to run virtualized SNFs on commodity servers. This reduces the cost of DDoS attack mitigation while enabling high flexibility by dynamically removing or adding SNF instances. However, this class of solutions sacrifices packet processing performance and incurs non-trivial end-to-end latency, which is unacceptable for many latency-sensitive Internet services. Recently, the emergence of programmable switches brings a promising alternative solution: arbitrary SNFs can be directly performed in line-rate ASIC pipelines of programmable switches, enabling low-cost, flexible, and high-performance DDoS attack mitigation. In this article, we present an illustrative survey of recent solutions that leverage programmable switches to provide DDoS attack mitigation. Our survey can help understand how to make full use of the benefits of programmable switches to defend against DDoS attacks.**

I. Introduction

DISTRIBUTED denial-of-service (DDoS) attacks continue to be a long-lasting critical threat in modern networks. The number of DDoS attacks in Q1 2020 has doubled compared to that in Q4 2019 while the peak volume of these attacks reaches several terabits per second (Tbps) [1]. These attacks interrupt daily Internet services such as online shopping and stock trading for legitimate users. Also, these attacks impose significant burdens for network administrators. To preserve profit, administrators need to minimize the negative impact of DDoS attacks by rejecting DDoS flows as fast as possible, leading to non-trivial capital expenditure and operating costs in network management.

Many solutions have been proposed to conduct DDoS attack mitigation. These solutions can be classified into two categories. The first class places massive hardware appliances, i.e., middleboxes, in the substrate network [2]. Each middlebox executes a specific security-oriented network function (SNF). The SNF scrubs incoming traffic by examining every packet and discarding malicious packets belonging to DDoS flows at several hundreds of Gigabits per second (Gbps). However, the maintenance of middleboxes is very expensive. For example, according to [3], the owners of large networks spent over a million dollars on middleboxes within five years. Such high financial cost is unacceptable for most companies. Also, middleboxes are proprietary, making them hard to upgrade. Adding a new feature to an existing middlebox typically experiences year-long deployment [3]. Thus, it is impractical for fixed-function middleboxes to cope with attack dynamics. To conclude, the first class of solutions exhibits *high cost* and *low flexibility* when handling DDoS attacks.

Moreover, the second class of solutions leverages the softwarization and virtualization techniques in software-defined networking (SDN) and network function virtualization (NFV) for DDoS attack mitigation. Specifically, these solutions virtualize SNFs traditionally offered by middleboxes, and install these SNFs on the virtual machines (VMs) or containers running on commodity servers. Since the price of servers is significantly lower than that of middleboxes, the overall financial cost is reduced by moving SNFs from traffic scrubbing centers to commodity servers. In addition, as SNFs are softwarized, they can be elastically added or removed to handle attack dynamics at runtime, making the second class of solutions highly flexible. For example, Bohatei [4] makes decisions of elastically scaling SNFs with respect to dynamic traffic volume of DDoS attacks. It also uses the SDN capability of global network control to steer traffic into different SNFs to achieve load balancing. Nevertheless, although this class of solutions provides low-cost and flexible DDoS attack mitigation, they suffer from poor packet processing performance. Even for a relatively low attack rate of 100 K packets per second, software-based SNFs increase the end-to-end per-packet processing latency by 50 μs to 1 ms [5]. Such high performance overhead is unacceptable for many latency-sensitive Internet services like stock trading. To conclude, the second class of solutions achieves low cost and high flexibility, but suffers from *poor performance* in DDoS attack mitigation.

The use of programmable switches sheds light on how to achieve low-cost, flexible, and high-performance DDoS attack mitigation [6]. Specifically, the benefits of adopting programmable switches for DDoS attack mitigation are threefold. First, according to the latest market prices, the DDoS attack mitigation built on programmable switches achieves orders-of-magnitude *lower cost* compared to other classes of solutions. The reason is that its monetary cost is determined by the price of programmable switches, which depends on

Xiang Chen, Hongyan Liu, and Chunming Wu are with the College of Computer Science and Technology, Zhejiang University. Dong Zhang is with the College of Computer Science and Big Data, Fuzhou University. Qun Huang is with the School of Electronics Engineering and Computer Science, Peking University. Haifeng Zhou is with the College of Control Science and Engineering, Zhejiang University. Qiang Yang is with the College of Electrical Engineering, Zhejiang University.

Corresponding author: Chunming Wu (wuchunming@zju.edu.cn).

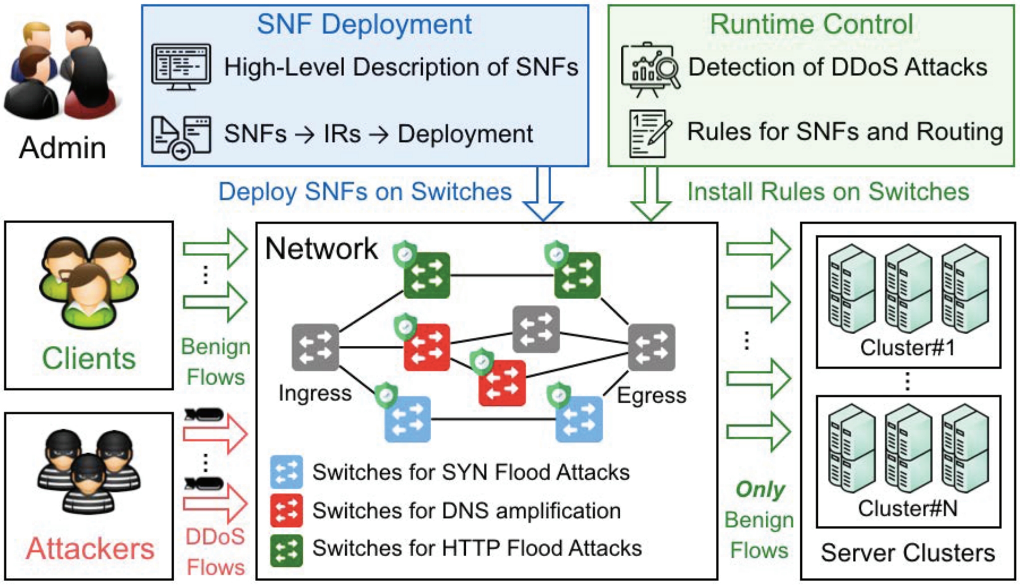

Admin
SNF Deployment
High-Level Description of SNFs
SNFs → IRs → Deployment
Runtime Control
Detection of DDoS Attacks
Rules for SNFs and Routing
Deploy SNFs on Switches
Install Rules on Switches
Clients
Benign Flows
Attackers
DDoS Flows
Network
Ingress
Egress
Switches for SYN Flood Attacks
Switches for DNS amplification
Switches for HTTP Flood Attacks
Only Benign Flows
Cluster#1
Cluster#N
Server Clusters

Fig. 1: *Overview of DDoS attack mitigation with programmable switches. The SNFs running on programmable switches detect DDoS flows and discard these flows at line rate, thereby protecting Internet services from being affected.*

both the die size and circuit complexity of the built-in ASIC chip. The ASIC chip used by programmable switches has the similar die size as traditional switch ASIC chips while exhibiting lower circuit complexity because of its pipeline-based design. Thus, the price of a programmable switch is viewed as low-cost, thereby making the DDoS attack mitigation built on programmable switches low-cost.Second, administrators can leverage high-level programming languages to change the packet processing logic of programmable switches on demand. Thus, they are able to customize their SNFs and run SNFs on programmable switches. Such a top-down SNF deployment enables *high flexibility*. Third, programmable switches provide strong guarantees on line-rate packet processing performance (e.g., 6.4 Tbps per switch [7]). Thus, switch-assisted SNFs can handle large DDoS attacks, enabling *high performance*.

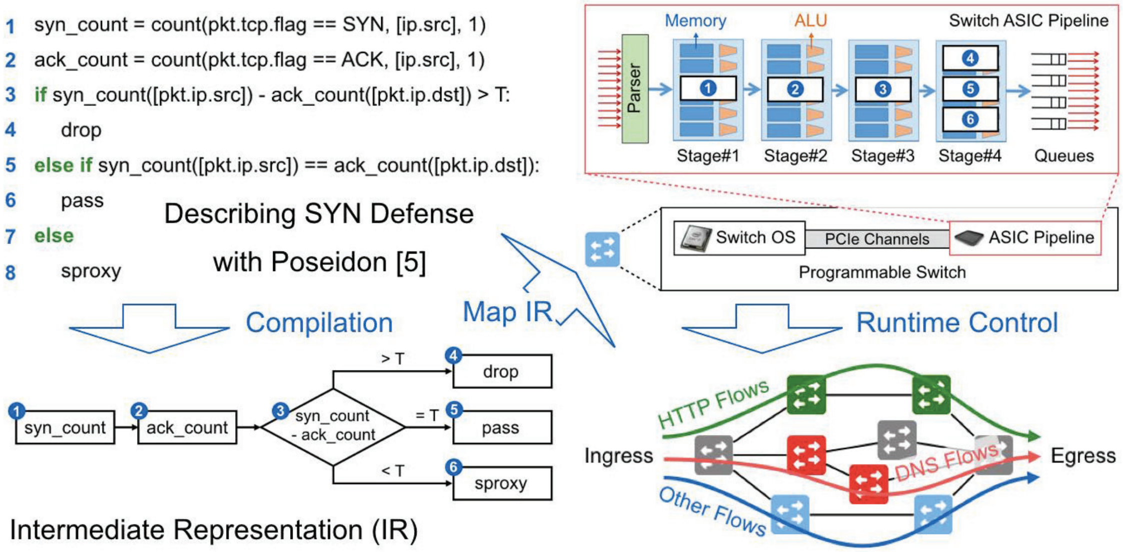
Although some excellent surveys have been done on DDoS defense, none of them consider the new paradigm of empowering DDoS attack mitigation with programmable switches in the literature. In this article, we fill this gap by presenting a survey of the recent solutions that leverage programmable switches to mitigate DDoS attacks. Our survey aims to make it easier for network administrators to choose appropriate switch-assisted DDoS mitigation solutions on demand. To the best of our knowledge, this is the first survey that focuses on switch-assisted DDoS mitigation and classifies existing solutions with respect to different aspects. We believe that the initial steps this article has taken help readers understand how to make full use of programmable switches to achieve low-cost, flexible, and high-performance DDoS mitigation.

II. DDoS Attack Mitigation with Programmable Switches

Figure 1 presents an overview of DDoS attack mitigation with programmable switches. At runtime, attackers send DDoS flows to the network to disturb normal Internet services. To this end, the SNFs running on programmable switches distinguish malicious DDoS flows from incoming traffic and discard these flows at line rate. As a result, only benign flows can pass through the network and reach server clusters, thus preventing normal Internet services. We further articulate the workflow of mitigating DDoS attacks with programmable switches, which includes three steps, i.e., high-level description of SNFs, SNF deployment, and runtime control.

Step#1: High-level description of SNFs. First of all, administrators use *high-level programming languages* to describe SNFs on demand. They can only focus on their logic of realizing SNFs without the need of caring about unnecessary details. Specifically, an SNF comprises several packet processing operations that detect malicious packets and handle different types of packets. Figure 2 shows the user program corresponding to an example of SNFs, i.e., SYN flood defense. This SNF is written in the high-level programming language offered by a recent framework that implements switch-assisted SNFs, Poseidon [5]. It aims to mitigate SYN flood attacks, which are the most popular type of DDoS attacks in recent years (e.g., more than 92.6% of DDoS attacks that happened in Q1 2020 are SYN flood attacks [1]). To achieve this, the SNF employs two switch counters, syn_count and ack_count, to detect malicious SYN packets (lines 1-2). These counters count the number of SYN packets and ACK packets per second, respectively. With counter values, the SNF determines whether the number of SYN packets in a flow (identified by its source IP address) is significantly higher than the number of ACK packets (i.e., exceeding a threshold T, line 3). If so, the flow may be a DDoS flow with a high probability, such that the SNF drops its future packets (lines 4-5). Moreover, if the number of SYN packets equals that of ACK packets, the flow is validated as an innocent flow (lines 6-7). Otherwise, the security of the flow is unknown so the SNF sends the flow to an SYN proxy for further monitoring (lines 8-9).

Step#2: SNF deployment on programmable switches. Administrators input their programs to a *network orchestrator* [8] that deploys the SNFs defined in input programs on



```
1  syn_count = count(pkt.tcp.flag == SYN, [ip.src], 1)
2  ack_count = count(pkt.tcp.flag == ACK, [ip.src], 1)
3  if syn_count([pkt.ip.src]) - ack_count([pkt.ip.dst]) > T:
4      drop
5  else if syn_count([pkt.ip.src]) == ack_count([pkt.ip.dst]):
6      pass
7  else
8      sproxy
```

Fig. 2: *Workflow of mitigating DDoS attacks with programmable switches.*

programmable switches. The orchestrator performs a three-phase procedure. First, it transforms input programs into intermediate representations (IRs), which are conducive for further deployment. These IRs are graphs, where nodes are basic SNF processing components while edges represent the execution dependencies among different nodes. For example, Figure 2 plots the IR of SYN flood defense, which has six nodes and five edges. Each node corresponds to a specific packet processing operation of SYN flood mitigation, e.g., the 3-rd node represents the operation of comparing the values of syn_count and ack_count to detect malicious flows. Each edge denotes the execution dependency among two operations, e.g., the edge connecting the 3-rd node and the 4-th node indicates that the 4-th node will be executed if and only if the result of the 3-rd node exceeds the detection threshold T.

Second, the orchestrator maps each IR to the ASIC pipeline of a programmable switch. Figure 2 plots an example of the ASIC pipeline of a programmable switch. The pipeline comprises a parser, which is responsible for parsing packet headers, four pipeline stages, each of which offers both memory resources and computational resources (i.e., a number of ALUs) for executing SNF operations, and queues for traffic scheduling. To map an IR to a pipeline, the orchestrator typically formulates an integer linear programming (ILP) problem. The problem maps every SNF operation to a specific pipeline stage while minimizing the number of stages occupied by SNF operations to retain resource efficiency. Also, there are mainly two constraints in this problem. The first constraint is the *resource limitation*, i.e., the resource consumption of SNF operations placed on a stage should not exceed the resource capacity of the stage. Another constraint is to preserve *execution dependencies*, i.e., interdependent operations cannot be placed on the same stage. By solving the ILP problem with commodity ILP solvers, the orchestrator obtains the optimal mapping between the IR and the target ASIC pipeline. For example, as shown in Figure 2, the SNF operations are mapped to pipeline stages while the pipeline will perform these operations in order.

Third, the orchestrator inputs the IR and the mapping acquired in the second phase to the device-specific switch compiler to generate switch configurations. It installs these configurations on programmable switches and completes its deployment of SNFs on programmable switches.

Step#3: Runtime control. At runtime, administrators need to control both programmable switches and other data plane devices (e.g., routers) to jointly mitigate DDoS attacks. Such runtime control is three-fold. First, administrators collect the latest traffic statistics (e.g., per-flow frequency) from switches and detect DDoS attacks by analyzing the collected statistics.

Second, they populate their rules or thresholds to programmable switches to activate SNFs to handle DDoS attacks. Take SYN flood defense as an example. Administrators should configure the ASIC pipeline of the switch that performs SYN flood defense with the value of the threshold T, which is used to detect malicious SYN packets (line 3, Figure 2). In detail, they can invoke the APIs offered by the operating system (OS) of the switch to populate their threshold value.

Third, administrators install routing rules on data plane devices to correctly direct traffic through the SNFs running on programmable switches. Figure 2 plots an example of such control. In this example, administrators generate three rules for the ingress router. The first rule guides the router to steer HTTP flows to pass through the two switches at the top of the topology (marked in green), which perform HTTP flood detection. The second rule directs DNS packets to the switches in the middle (marked in red), enabling the SNF of DNS amplification mitigation. The third rule specifies that other flows should be routed to the switches at the bottom of the topology (marked in blue) in order to perform following the detection of SYN flood attacks. With these rules, various flows are directed to corresponding SNFs, which prevents Internet services from being affected.

In particular, in addition to manual SNF control, recent studies [8] also proposed security network orchestrators to provide the capability of automatic SNF control. More precisely, these orchestrators will automatically collect the latest traffic statistics from the data plane and make corresponding decisions on how to manage and control SNFs. By this means,

TABLE I: *Comparison between the solutions that empower DDoS attack mitigation with programmable switches.*

Solution	Easy-to-use	Detection	Defense	Runtime Control
P4SC [10]	✗	✗	✗	✗
LightNF [11]	✓	✗	✓	✗
Poseidon [5]	✓	✗	✓	✓
Jaqen [9]	✓	✓	✓	✓
INDDoS [12]	✗	✓	✗	✗

these orchestrators set administrators free from laborious manual efforts, thereby significantly easing network management.

In addition, when a switch was only geared to handle one type of DDoS attack (i.e., only one SNF was deployed) while administrators want to alter their intent to defend against other attacks, administrators can exploit two approaches in this case. (1) They can use runtime control to direct the attack traffic to other switches, which collaborate with the current switch to cooperatively mitigate various attacks. (2) In general, a single programmable switch can simultaneously run several SNFs. Thus, administrators can deploy multiple SNFs on the target switch in advance. They can flexibly activate their desired SNFs to react to attacks on demand, e.g., activating the DNS amplification defense to replace the SYN flood attack defense.

III. Solutions of Programmable Switch-Assisted DDoS Attack Mitigation

However, it is challenging to use programmable switches for DDoS mitigation because switches exhibit strict resource limitations [5, 9]. For example, the memory capacity of a single switch is typically less than 50 MB. These resource limitations should be carefully taken into consideration when deploying SNFs on programmable switches. Otherwise, the SNF deployment will fail.

To address the above challenge, several solutions have been proposed in recent years. These solutions follow the workflow in §II to defend against DDoS attacks with programmable switches. In this section, we present a survey of these solutions. Our survey consists of the following two parts.

First, we classify these solutions based on four types of properties, including (1) *easy-to-use* (i.e., the solution offers a high-level approach for administrators to describe SNFs), (2) *detection* (i.e., the solution is able to detect DDoS attacks), (3) *defense* (i.e., the solution is capable of defeating DDoS attacks), and (4) *runtime control* (i.e., the solution enables runtime control of SNFs and traffic routing). Table I presents our classification. It shows that some of these solutions only focus on DDoS attack defense, while others support both detection and defense. With this classification, administrators can determine their needs and choose one of existing solutions from their demanded category. In general, it is recommended to employ the solution that activates both detection and defense on programmable switches (e.g., Jaqen [9]) to reduce the cost of DDoS attack mitigation. Also, if administrators have already adopted some approaches to detect DDoS attacks in their networks, choosing the solution that only activates DDoS defense (e.g., Poseidon [5]) is enough.

Second, we elaborate on the benefits and drawbacks of each solution in what follows. This part helps to gain insight into the differences between existing solutions.

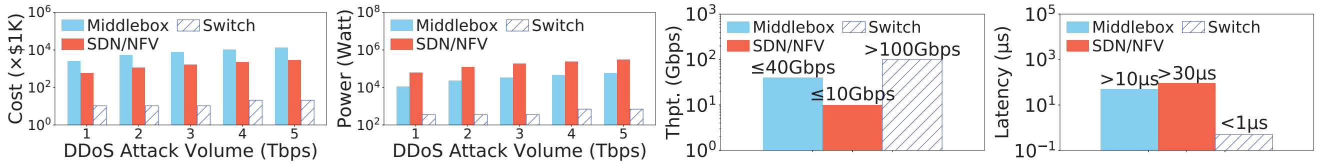
P4SC [10]. P4SC presents a preliminary system to implement SNFs on programmable switches. To use P4SC in DDoS attack mitigation, administrators first need to construct their requests of SNF chains. P4SC receives input requests and transforms these requests into IRs, which are directed acyclic graphs (DAGs). Next, P4SC merges these DAGs into a compound DAG. The reason is that there exist some duplicate SNFs among different DAGs. Thus, merging DAGs can reduce redundancy among DAGs, which reduces the resource consumption of SNF deployment. After that, P4SC transforms the compound DAG into a switch program and inputs this program to the switch compiler for SNF deployment. Moreover, since the management APIs offered by switches are low-level and device-specific, P4SC encapsulates these APIs to a set of primitives for administrators to populate SNF rules.

However, P4SC has several limitations due to its preliminary design. First, it does not provide a high-level approach for administrators to define their SNFs. In this context, administrators have to understand low-level details about switch architectures and restrictions when writing SNFs, leading to non-trivial development burdens. Second, P4SC does not consider the resource limitations imposed by programmable switches. Thus, its SNF deployment may be failed. Third, its runtime control does not support traffic routing. As a result, SNFs may miss some flows of interest, making it possible for malicious flows to bypass SNFs.

LightNF [11]. Similar to P4SC, LightNF focuses on the SNF implementation on the switch ASIC pipeline. But it surpasses P4SC with two characteristics. First, it shields device-specific details and enables administrators to write SNFs with high-level primitives. These primitives are intent-based, reducing development burdens. Second, it faithfully preserves switch resource limitations during SNF deployment. Thus, it addresses the challenge and avoids potential deployment failures.

However, LightNF does not offer a reasonable mechanism for runtime control. It is also unable to detect DDoS attacks.

Poseidon [5]. Compared to P4SC and LightNF, Poseidon presents a comprehensive system for DDoS attack mitigation. It offers an intuitive and modular language for administrators to write their SNFs in defense policies (e.g., SYN flood defense in Figure 2). Next, it receives defense policies from administrators as input and maps these policies onto the substrate network comprising programmable switches and commodity servers. Its mapping preserves switch resource limitations while maximizing the number of policies mapped to programmable switches to achieve the maximum performance. Moreover, administrators may change their defense policies to handle attack dynamics at runtime, indicating that some existing SNFs will be migrated to new locations or be replaced by new types of SNFs. In this context, the processing states (e.g., per-flow packet counts) recorded by SNFs may be lost, leading to inconsistent packet processing. To this end, Poseidon replicates the processing states of SNFs to commodity servers. When administrators modify their policies, Poseidon carefully directs traffic to pass through the correct sequence of the switches that run new SNFs and the servers that maintain the replicated processing states. In this way, the



(a) *Monetary cost.* (b) *Power consumption.* (c) *Throughput.* (d) *Per-packet processing latency.*

Fig. 3: *Comparison between the programmable switch-assisted solution and other solutions. (a)-(b) Cost of DDoS attack mitigation vs. DDoS attack volume. (c)-(d) Comparison of packet processing performance.*

runtime control of Poseidon ensures defense consistency even with attack dynamics and policy changes.

Poseidon assumes that DDoS attacks have been identified in advance. However, this is a questionable consumption due to two reasons. First, existing detection methods leverage packet sampling techniques to collect coarse-grained traffic statistics from the substrate network for attack detection. However, packet sampling misses lots of traffic statistics, leading to highly inaccurate detection. Second, existing detection methods require additional network resources such as dedicated monitoring servers to conduct offline analysis of collected statistics, which incurs non-trivial detection delay.

Jaqen [9]. Liu *et al*. propose a system, namely Jaqen, to address the problem of detecting DDoS attacks in Poseidon. In addition to the normal workflow of mitigating DDoS attacks, Jaqen further employs some instances of universal sketches [13] on programmable switches. These sketches approximately summarize traffic statistics with low switch resource consumption and provably high accuracy. Jaqen periodically collects sketch values to the control plane, which accurately recovers complete traffic statistics in a timely manner. In this way, it enables timely and accurate detection of DDoS attacks. Moreover, Jaqen provides a network-wide resource manager that makes decisions in routing and traffic engineering in order to deal with attack dynamics.

INDDoS [12]. Unlike Jaqen that introduces innovative system design to tackle the problem of detecting DDoS attacks, INDDoS proposes to solve the problem algorithmically and theoretically. INDDoS provides two main contributions. First, by combining different types of sketches, it presents a resource-efficient and probabilistic data structure that estimates per-destination flow cardinality on programmable switches. Second, it runs the proposed data structure on the ASIC pipeline of programmable switches to directly identify which victims have experienced attacks in the data plane. Thus, INDDoS eliminates the need of involving the control plane for attack detection and thus achieves efficient attack detection.

INDDoS does not aim to provide DDoS attack defense or runtime control. Nevertheless, it can be integrated into other solutions like Poseidon to jointly provide a comprehensive system for DDoS attack mitigation.

IV. Case Study: Evaluating SYN Flood Defense

We present a concrete case study that mitigates SYN flood attacks to demonstrate the benefits of adopting programmable switches in DDoS attack mitigation. We implement the following three solutions that realize the SNF of SYN flood defense, including (1) the *middlebox-based solution* that simulates SYN flood defense on an NSFOCUS ADS 8000 middlebox [14], (2) the *SDN/NFV-based solution* that uses a state-of-the-art NFV framework, E2 [15], to realize SYN flood defense on a standard Linux box server with 36-core Intel(R) Xeon(R) Gold 6240C 2.60 GHz CPU and 128 GB RAM, and (3) the *switch-assisted solution* that invokes Poseidon [5] to implement the SNF on a 32×100 Gbps Tofino-based programmable switch. We first estimate the cost of using these solutions in SYN flood defense. According to the latest device prices and power consumption acquired from official websites, Figure 3(a)-(b) present the estimation of the monetary cost (in dollars) and the power consumption (in Watts) of using these solutions to defend against SYN flood attacks with different volumes. It indicates that compared to other solutions, the switch-assisted solution achieves orders-of-magnitude lower cost in SYN flood defense. This demonstrates the low cost of using programmable switches for DDoS attack mitigation.

Next, we evaluate the packet processing performance of these solutions. To do this, we build a real testbed, which is a linear topology comprising three devices. The device in the middle performs SYN flood defense, while the other two devices are two servers that run a traffic generator and a traffic receiver, respectively. The three devices are directly connected via 100 Gbps links. We manually generate a traffic workload comprising 64-byte packets at 100 Gbps. Then we direct the workload to go through the linear topology in an end-to-end manner. To quantify packet processing performance, we measure both throughput and per-packet processing latency of the SNF of SYN flood defense executed by the above solutions. As shown in Figure 3(c)-(d), the switch-assisted solution outperforms other solutions with significantly higher throughput (>100 Gbps) and orders-of-magnitude lower latency (<1 μs). The reason is that the programmable switch guarantees line-rate packet processing performance, making it feasible to achieve high-performance DDoS attack mitigation.

V. Open Issues

The promising trend of leveraging programmable switches for DDoS attack mitigation also introduces new open issues. We highlight two issues as follows.

Support of complex SNFs. Existing programmable switches use a pipeline-based paradigm for packet processing. This pipeline-based paradigm partitions packet processing into several stages. Due to the concern of chip footprints and heat consumption, each pipeline stage can only perform a limited number of simple packet processing operations on each arrival packet within a small time budget. Such computational limitations forbid several SNF operations such as the encryption and decryption of layer-7 headers (e.g., HTTPS), which prevents many complex SNFs such as deep packet inspection from being deployed on programmable switches. Consequently, how to empower complex SNFs with the benefits of programmable switches remains unaddressed. One possible solution is to co-design commodity servers and programmable switches to jointly deploy these complex SNFs. More precisely, administrators can choose to partition the complex SNF into two parts, including an offloadable part comprising the SNF operations that can be deployed on programmable switches and an unoffloadable part consisting of other operations. They can deploy the offloadable part to programmable switches while implementing the unoffloadable part on servers. Such co-design can maximize the performance of complex SNFs while observing switch limitations.

Automatic resource allocation. Each SNF demands a portion of switch resources to process incoming traffic with high accuracy. For instance, a layer-3 Firewall needs to store a large number of rules in in-switch memory to accurately reject every malicious flow. However, given various forms of SNFs, it is time-consuming and laborious to estimate the exact amount of resources required by an arbitrary SNF to achieve a desired level of accuracy. Also, these estimates are closely relevant to incoming traffic rate. When incoming traffic rate is high, SNFs require more resources to handle massive packets arriving in a short period of time while achieving desired accuracy compared to the case of low incoming traffic rates. To date, existing solutions lack a reasonable mechanism to acquire accurate estimates of SNF resource consumption. As a result, SNFs may suffer from low accuracy due to insufficient resource allocation. One possible solution to the above problem may be to dynamically reallocate resources with respect to the current workloads of SNFs. But it requires careful measurement of SNF workloads and accurate estimates of allocated resources, making its design challenging.

VI. Conclusion

The emergence of programmable switches creates a fascinating paradigm for DDoS attack mitigation: deploying and managing SNFs on programmable switches enable low cost, high flexibility, and high performance. This article presents the workflow of using programmable switches to perform DDoS attack mitigation. It reviews recent solutions that realize this paradigm and compare their capabilities. It also presents some significant open issues that remain to be addressed.

Acknowledgement

This work is supported by the National Key R&D Program of China (2020YFB1804705), the Key R&D Program of Zhejiang Province (2021C01036,2020C01021), and the Fundamental Research Funds for the Central Universities (Zhejiang University NGICS Platform).

Reference

[1] Ddos attacks in q1 2020, accessed: October 23, 2021. [Online]. Available: https://securelist.com/ddos-attacks-in-q1-2020/96837/

[2] M. Jonker, A. Sperotto, R. van Rijswijk-Deij, R. Sadre, and A. Pras, "Measuring the adoption of ddos protection services," in *ACM IMC*, 2016, pp. 279–285.

[3] J. Sherry, S. Hasan, C. Scott, A. Krishnamurthy, S. Ratnasamy, and V. Sekar, "Making middleboxes someone else's problem: network processing as a cloud service," *ACM SIGCOMM Computer Communication Review*, vol. 42, no. 4, pp. 13–24, 2012.

[4] S. K. Fayaz, Y. Tobioka, V. Sekar, and M. Bailey, "Bohatei: Flexible and elastic ddos defense," in *USENIX Security*, 2015, pp. 817–832.

[5] M. Zhang, G. Li, S. Wang, C. Liu, A. Chen, H. Hu, G. Gu, Q. Li, M. Xu, and J. Wu, "Poseidon: Mitigating volumetric ddos attacks with programmable switches," in *NDSS*, 2020, pp. 823–835.

[6] F. Paolucci *et al.*, "Enhancing 5g sdn/nfv edge with p4 data plane programmability," *IEEE Network*, vol. 35, no. 3, pp. 154–160, 2021.

[7] Barefoot Tofino, accessed: October 21, 2021. [Online]. Available: https://www.barefootnetworks.com/technology/#tofino

[8] A. Zaalouk, R. Khondoker, R. Marx, and K. Bayarou, "Orchsec: An orchestrator-based architecture for enhancing network-security using network monitoring and sdn control functions," in *IEEE NOMS*, 2014, pp. 1–9.

[9] Z. Liu, H. Namkung, G. Nikolaidis, J. Lee, C. Kim, X. Jin, V. Braverman, M. Yu, and V. Sekar, "Jaqen: A high-performance switch-native approach for detecting and mitigating volumetric ddos attacks with programmable switches," in *USENIX Security*, 2021.

[10] X. Chen, D. Zhang, X. Wang, K. Zhu, and H. Zhou, "P4sc: Towards high-performance service function chain implementation on the p4-capable device," in *IEEE/IFIP IM*, 2019, pp. 1–9.

[11] X. Chen, Q. Huang, P. Wang, Z. Meng, H. Liu, Y. Chen, D. Zhang, H. Zhou, B. Zhou, and C. Wu, "Lightnf: Simplifying network function offloading in programmable networks," in *IEEE/ACM IWQOS*, 2021, pp. 1–10.

[12] D. Ding, M. Savi, F. Pederzolli, M. Campanella, and D. Siracusa, "In-network volumetric ddos victim identification using programmable commodity switches," *IEEE Transactions on Network and Service Management*, vol. 18, no. 2, pp. 1191–1202, 2021.

[13] Z. Liu, A. Manousis, G. Vorsanger, V. Sekar, and V. Braverman, "One sketch to rule them all: Rethinking network flow monitoring with univmon," in *ACM SIGCOMM*, 2016, pp. 101–114.

[14] Nsfocus ads 8000, accessed: October 23, 2021. [Online]. Available: https://nsfocusglobal.com/wp-content/uploads/2018/05/Anti-DDoS-Solution.pdf

[15] S. Palkar, C. Lan, S. Han, K. Jang, A. Panda, S. Ratnasamy, L. Rizzo, and S. Shenker, "E2: a framework for nfv applications," in *ACM SOSP*, 2015, pp. 121–136.

Xiang Chen received the B.Eng. and the M.Eng. degrees from Fuzhou University in 2019 and 2021, respectively. He is currently pursuing the Ph.D. degree with Zhejiang University, China. He received the Best Paper Award from IEEE/ACM IWQoS 2021. His researches focus on programmable networks.

Hongyan Liu received the B.Eng. degree from Fuzhou University, in 2020. He is currently pursuing the M.Eng. degree with the College of Computer Science, Zhejiang University. His research focuses on programmable networks.

Qun Huang is an Assistant Professor at Department of Computer Science and Technology, Peking University. He received his bachelor degree in Computer Science from Peking University in 2011. He got his Ph.D. degree in August 2015 in The Chinese University of Hong Kong. His research focuses on network measurement.

Dong Zhang received the Ph.D. degree from Zhejiang University, China, in 2010. He visited Alabama University, USA, as a visiting scholar from 2018 to 2019. He is currently an Professor at Fuzhou University, China. His research areas include software defined networking and Internet QoS.

Haifeng Zhou received the Ph.D. degree in computer science and technology from Zhejiang University in 2018. He is now an associate research fellow in College of Control Science and Engineering, Zhejiang University. His current research interests include Internet and security, intelligent networks and security systems.

Chunming Wu received the Ph.D. degree in computer science from Zhejiang University in 1995. He is currently a Professor with the College of Computer Science and Technology, Zhejiang University, China. His research fields include reconfigurable networks, proactive network defense and network security.

Qiang Yang received Ph.D. degree in Electronic Engineering and Computer Science from Queen Mary, University of London, U.K., in 2007. He is currently a Professor at College of Electrical Engineering, Zhejiang University. His research interests include smart energy systems and control. He is the Fellow of British Computer Society (BCS).