International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(5s) 2026 pp. 863-872
https://doi.org/10.70917/ijcisim-2026-2809
Article

An Adaptive and Scalable DDoS Prevention Framework for Software Defined Networks

Nirzari Patel[1], Hiren B. Patel[2]

[1]Department of Computer Engineering, Kadi Sarva Vishwavidyalaya, Gandhinagar, Gujarat, India.
Email: patelnirzari84@gmail.com
[2]Vidush Somany Institute of Technology and Research, Kadi Sarva Vishwavidyalaya, Gandhinagar, Gujarat, India.
Email: hbpatel1976@gmail.com
Corresponding Author: Nirzari Patel

Abstract: — Central governance and flexible network administration are made possible by Software Defined Networking (SDN); still, this architectural benefit also makes the control plane vulnerable to Distributed Denial of Service (DDoS) attacks. An extreme number of flow requests and packet-in events can significantly reduce controller effectiveness and interfere with network functions in the context of such attacks. In this work, we change and estimate an adaptive DDoS prevention framework based on knowledge gained from SDN emulation tests. Relatively than relying on predetermined mitigation thresholds, the framework dynamically adjusts mitigation strategies based on the attack's severity and the controller's present load. The proposed approach reduces unnecessary interactions in the control plane while maintaining service quality for authorized traffic by incorporating controller-aware decision-making. The adaptive outline lessens controller CPU utilization, speeds up mitigation response times, lowers end-to-end latency, and keeps higher throughput when compared to static mitigation procedures, according to experimental evaluations carried out in a precise SDN emulation environment

Keywords: —Software Defined Networking, DDoS Mitigation, Adaptive Security, Controller Scalability, Network Performance.

1. INTRODUCTION

Software Defined Networking (SDN) has transformed how modern networks are managed by separating forwarding de- vices from control logic and placing network intelligence at a centralized controller. This strategy allows network operators to implement flexible traffic control, easy management, and speedy policy deployment. As a result, SDN has been gradually adopted in cloud data centers, enterprise networks, and programmable network organizations.

However, while working with SDN-based environments, it becomes evident that the centralized controller also introduces new security concerns. In specific, Distributed Denial of Service (DDoS) attacks can exploit the interaction between switches and the organizer by generating a large number of new or abnormal flows. Such behavior leads to extreme packet-in messages and flow setup requests, which may overload controller resources. When the controller experiences high processing delay or saturation, legitimate traffic advancing decisions are delayed, directly affecting overall network performance and availability.

2. RELATED WORK

DDoS defense mechanisms in Software Defined Networking have been widely considered due to the characteristic vulnerability of centralized controllers. Existing research can be largely categorized into detection-based approaches, mitigation-based tactics, and fusion frameworks integrating both detection and mitigation.

A. DDoS Detection in SDN

Recent studies leverage machine learning and deep learning techniques to improve detection accuracy and robustness [7], [8]. Flow-level structures such as packet rate, flow duration, and protocol distribution are commonly

Copyright: © 2026 by the authors

used for training supervised models [5], [15]. Understandable and adaptive learning models have also been explored to improve transparency and lessen false positives in SDN-based intrusion detection systems [2], [16]. While these approaches achieve high detection accuracy, most studies primarily emphasize classification performance and provide limited discussion on scalable mitigation apparatuses.

B. DDoS Mitigation Strategies in SDN

Mitigation-oriented research classically employs rate limiting, packet filtering, and flow rule installation to suppress attack traffic [11]. Many tactics rely on static thresholds, which may be ineffective under rapidly changing traffic situations [4]. Controller-aware mitigation strategies have recently gained attention, identifying that mitigation actions themselves im- pose overhead on the SDN controller [1]. However, existing controller-aware approaches often apply coarse-grained mitigation policies and lack powdered adaptability based on attack severity and controller health system of measurement [17].

C. Hybrid Detection and Mitigation Frameworks

Hybrid frameworks fit in detection and mitigation into a unified SDN security solution [10]. While these frame- works advance reply time compared to standalone detection systems, many do not clearly account for controller performance indicators such as CPU consumption and packet-in rate [9]. Furthermore, scalability evaluation remains limited in most hybrid approaches, with tests showed on small or fixed network topologies [14].

D. Research Gap and Motivation

Existing DDoS defense mechanisms in SDN suffer from three key limits: (i) overemphasis on detection accuracy without corresponding mitigation scalability, (ii) reliance on static or semi-dynamic mitigation strategies, and (iii) deficient reflection of controller performance and scalability [1], [4], [9]. This work speeches these gaps by proposing a controller- aware adaptive DDoS prevention framework that dynamically balances security efficiency and network performance, ensuring scalable and strong SDN operation under diverse attack scenarios.

Architecture of the Proposed Controller-Aware Adaptive DDoS Prevention Framework

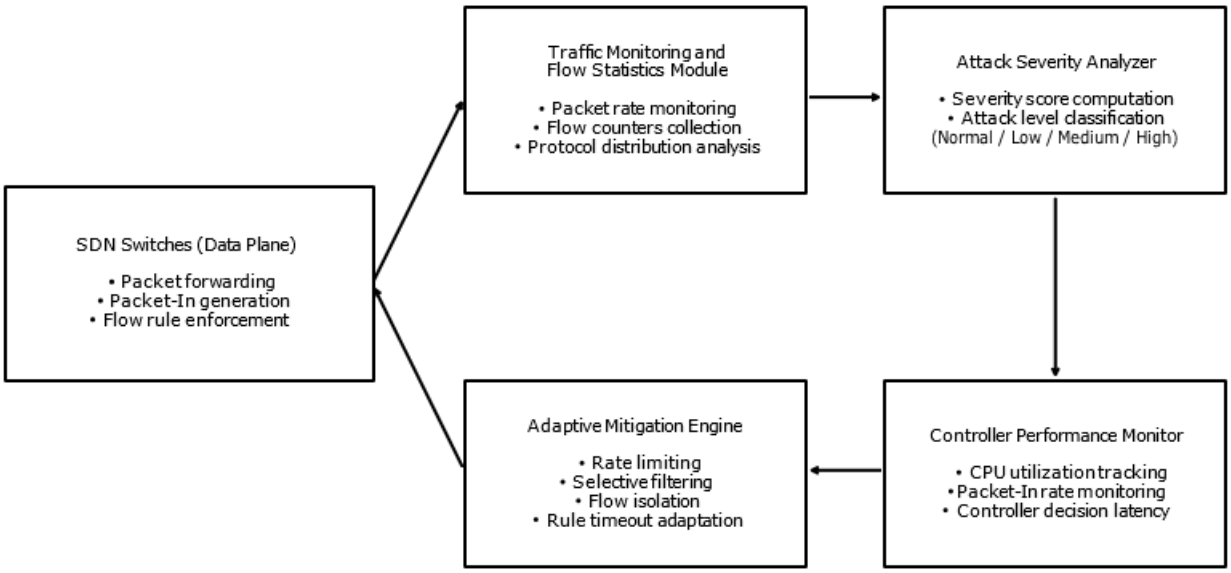

Fig. 1: Architecture of the proposed adaptive DDoS prevention framework in SDN.

3. PROPOSED ADAPTIVE DDOS PREVENTION FRAMEWORK

A. Architecture Overview

The planned framework exist in in the SDN controller and contains: Traffic Monitoring Module, Flow Statistics Collector, Attack Severity Analyzer, Adaptive Mitigation Engine, and Controller Performance Monitor.

B. Threat Model

The threat model considered in this work focuses on Scattered Denial of Service (DDoS) attacks targeting Software Defined Networking environments. The adversary is assumed to control one or more compromised end hosts capable of generating high-rate malevolent traffic toward victim hosts or network services. The attacker's primary objective is to degrade network performance and disrupt service availability by exhausting either data plane resources or control plane volume [1], [9].

Specifically, the attacker may launch volumetric flooding attacks using ICMP, TCP, or UDP packets, resulting in a large number of new or uncharacteristic flows. These flows cause frequent packet-in messages and flow setup requests at SDN switches, thereby increasing the processing burden on the controller [17]. Such performance can lead to controller CPU capacity, increased decision latency, and delayed rule installation, eventually impacting legitimate traffic.

The following assumptions are made in the threat model:

1) The SDN controller and switches are not physically compromised.
2) The communication channel between the controller and switches is trusted and correctly established.
3) Attackers do not possess internal controller credentials or administrative privileges.
4) The attacker's capabilities are limited to generating malicious traffic at the data plane level.

While advanced attacks such as controller hijacking or in- sider threats are beyond the scope of this work, the framework targets a wide class of realistic DDoS scenarios commonly addressed in recent SDN security literature [10].

C. Flow Feature Set

The framework extracts flow-level features over a sliding window Δt:

- Packet-in rate r_{pi} (events/sec)
- New flow request rate r_{nf} (flows/sec)
- Average packet rate r_p (packets/sec)
- Protocol distribution vector $\mathbf{p} = (p_{tcp}, p_{udp}, p_{icmp})$
- (Optional) Entropy of source IPs H_{src} [7], [8]

D. Flow Feature Set

The framework extracts flow-level features over a sliding window Δt:

- Packet-in rate r_{pi} (events/sec)
- New flow request rate r_{nf} (flows/sec)
- Average packet rate r_p (packets/sec)
- Protocol distribution vector $\mathbf{p} = (p_{tcp}, p_{udp}, p_{icmp})$
- (Optional) Entropy of source IPs H_{src} [7], [8]

E. Severity Scoring

Severity assessment based on flow-level statistics is widely adopted in SDN-based DDoS defense due to its lightweight computation and real-time applicability [1], [4], [17]. Ac- cordingly, attack severity is computed as a normalized score $S(t) \in [0, 1]$:

$$S(t) = \alpha \cdot \hat{r}_{pi} + \beta \cdot \hat{r}_{nf} + \gamma \cdot \hat{r}_{pv} \quad (1)$$

where $\hat{r}$ terms are min-max normalized, and $\alpha + \beta + \gamma = 1$.

Severity levels are defined as:

$$\text{Level}(t) = \begin{cases} \text{Normal}, & S(t) < \vartheta_1 \\ \text{Low}, & \vartheta_1 \le S(t) < \vartheta_2 \\ \text{Medium}, & \vartheta_2 \le S(t) < \vartheta_3 \\ \text{High}, & S(t) \ge \vartheta_3 \end{cases} \quad (2)$$

F. Controller Health Index

Controller-aware mitigation has recently gained attention as an effective means to prevent control plane saturation during DDoS attacks [1], [9]. To capture controller health, a composite controller load index $C(t)$ is defined as:

$$C(t) = \lambda \cdot \widetilde{CPU}(t) + (1 - \lambda) \cdot \hat{D}(t), \quad (3)$$

where $\widetilde{CPU}(t)$ is normalized CPU utilization and $\hat{D}(t)$ is normalized decision latency.

G. Adaptive Mitigation Policy

Adaptive mitigation policies that combine attack severity with controller performance indicators have been shown to improve scalability and response efficiency in SDN environ- ments [4], [17], [18]. Algorithm 1 summarizes the proposed adaptive DDoS prevention mechanism.

H. **Mitigation Actions**

Rate limiting: Apply OpenFlow meter rules per source/port.

Selective filtering: Drop packets matching suspicious patterns.

Flow isolation: Redirect suspicious flows to quarantine switch/port.

Timeout tuning: Increase rule timeouts to reduce packet-in storms when controller load is high [1], [17].

Algorithm 1 Adaptive DDoS Prevention Policy

Require: Flow stats over window Δt, thresholds $\theta_1, \theta_2, \theta_3$, controller limit ϕ

Ensure: Mitigation action at switches

Compute severity score $S(t)$ using (1)
Determine Level(t) using thresholds
Compute controller index $C(t)$ using (3)
if Level(t) = Normal **then**
    Allow traffic; maintain baseline rules
else if Level(t) = Low **then**
    Apply mild rate limiting via meter rules; short timeouts
else if Level(t) = Medium **then**
    Selective filtering + progressive rate reduction; moder- ate timeouts
else
    Flow isolation + aggressive rate limiting; extended timeouts
end if

13: **if** $C(t) > \phi$ **then**

14: Reduce controller interactions: increase rule timeout; aggregate rules

15: **end if**

16: Push FlowMod/Meter rules to switches

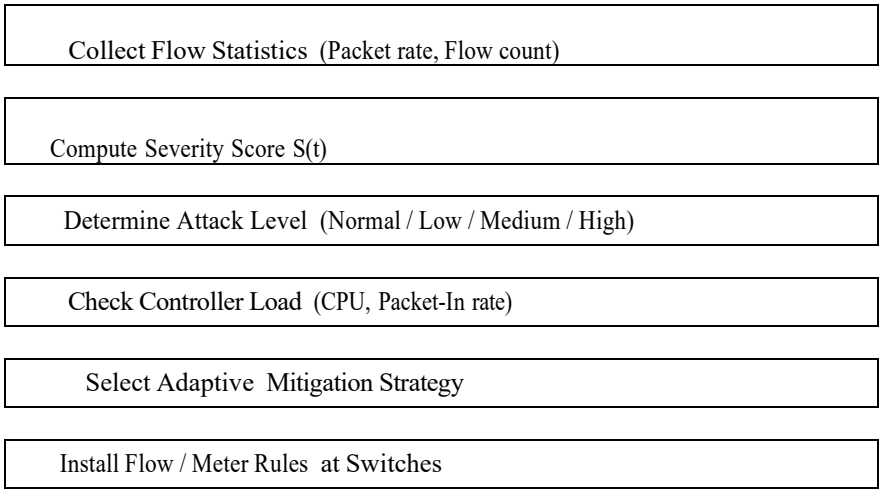

Fig. 2: Flowchart illustrating adaptive mitigation decisions based on attack severity and controller load.

I. Complexity Analysis

Let n be the number of active flows in a window. Feature extraction is $O(n)$, severity scoring is $O(1)$ per window, and rule update cost is proportional to the number of mitigated flows m (typically $m \ll n$). Hence, the per-window complex- ity is $O(n + m)$.

4. EXPERIMENTAL SETUP

A. Environment and Tools

Experiments are conducted using Mininet and the Ryu controller on Ubuntu Linux. Table I summarizes the system configuration. The experimental environment and toolchain

TABLE I: System Configuration and Software Environment

Component	Specification
Network Emulator	Mininet
SDN Controller	Ryu Controller
Controller Language	Python
Operating System	Ubuntu Linux
RAM	16 GB
Traffic Generation	hping3 / custom scripts

TABLE II: Topology Settings for Scalability Evaluation

Topology	#Switches	#Hosts	Links
Small	4	12	Linear/Tree
Medium	6	18	Tree
Large	10	40	Tree/Fat-tree (emulated)

follow commonly adopted SDN security evaluation practices reported in recent studies [5], [7], [14].

B. Topology Settings (Scalability)

We evaluate multiple topology scales. Table II lists the parameters.

C. Attack Scenarios

DDoS traffic is generated using ICMP/TCP/UDP flooding with controlled intensity (low, medium, high) to stress both the control plane and data plane [10], [17].

D. Evaluation Metrics

We measure controller CPU utilization, packet-in rate, mit- igation response time, end-to-end latency, throughput, false positive rate, ROC/PR characteristics, and scalability behavior [1], [10], [17].

5. RESULTS

From the experimental evaluation, we observed clear differ- ences between static and adaptive mitigation strategies under DDoS attack conditions. In the absence of mitigation, the controller experienced frequent packet-in bursts and sustained CPU utilization, indicating control plane saturation. Static mitigation reduced some of this load; however, it often reacted late under higher attack rates.

In contrast, the adaptive mitigation strategy consistently responded earlier by adjusting mitigation intensity according to both attack behavior and controller load. This behavior reduced repeated controller interactions and resulted in lower CPU utilization, faster response time, and improved network stability during prolonged attack periods.

A. Controller Overhead Analysis

Fig. 3 illustrates controller CPU utilization over time during a DDoS attack. The no-mitigation scenario exhibits the highest CPU consumption, indicating severe controller overload. Static mitigation reduces CPU usage but still incurs significant processing overhead due to frequent rule updates and packet-in events. In contrast, the proposed adaptive mitigation strategy

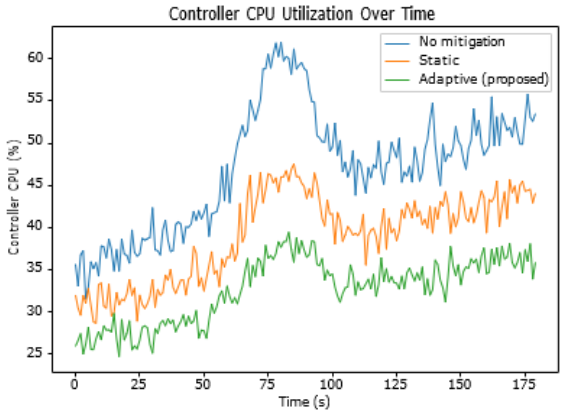

Fig. 3: Controller CPU utilization over time

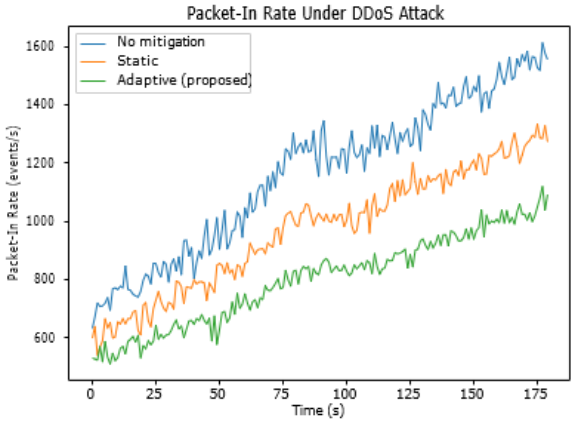

Fig. 4: Packet-in rate under DDoS attack

maintains lower CPU utilization by dynamically adjusting mitigation intensity and reducing unnecessary controller in- teractions, consistent with controller-aware defense objectives [1], [9].

B. Packet-In Rate Analysis

Fig. 4 shows the packet-in event rate over time for differ- ent mitigation strategies. Without mitigation, packet-in events increase sharply during attack periods, overwhelming the controller. The adaptive approach significantly lowers packet- in events by installing longer-lived rules and aggregating mitigation actions based on severity, which helps preserve controller stability [17].

C. Detection and Mitigation Response Time

Fig. 5 presents response time versus attack rate. As at- tack intensity increases, response time rises for both strate- gies. However, the adaptive approach responds faster due to severity-aware policy selection and reduced control plane overhead [4], [17].

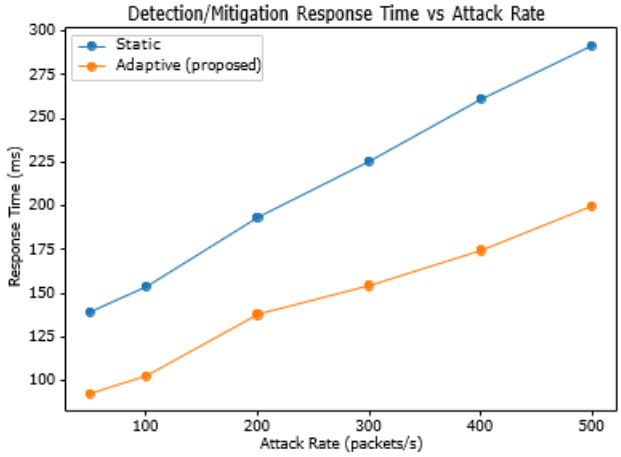

Fig. 5: Detection and mitigation response time versus attack rate

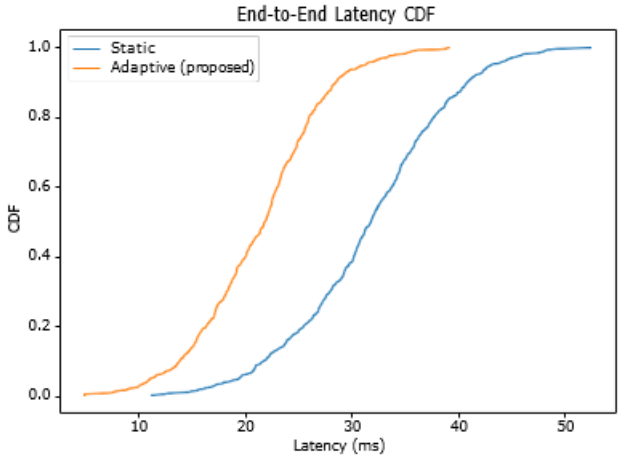

Fig. 6: CDF of end-to-end latency

D. End-to-End Latency Analysis

Fig. 6 illustrates the latency CDF under static and adaptive mitigation. The adaptive strategy yields lower latency for a larger fraction of packets, indicating improved quality of service while still suppressing attack traffic [1], [17].

E. False Positive Rate versus Threshold

Fig. 7 shows the false positive rate across decision thresh- olds. The adaptive strategy maintains a lower false positive rate, reflecting improved robustness under dynamic traffic and congestion conditions [2], [16].

F. ROC and Precision–Recall Analysis

Figs. 8 and 9 summarize detection performance. The adap- tive approach shows stronger discrimination (ROC) and main- tains higher precision at comparable recall levels, which is important under class imbalance typical of DDoS datasets [8], [16].

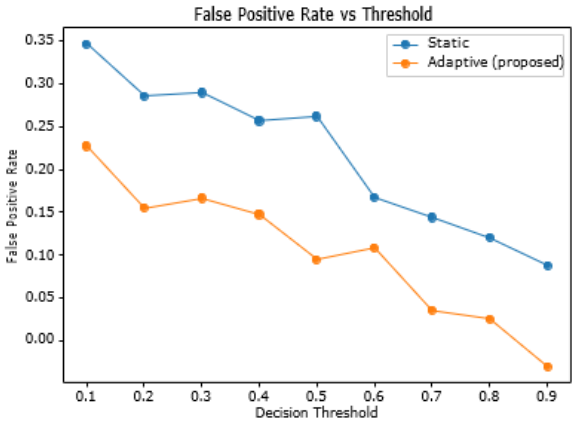

Fig. 7: False positive rate versus decision threshold

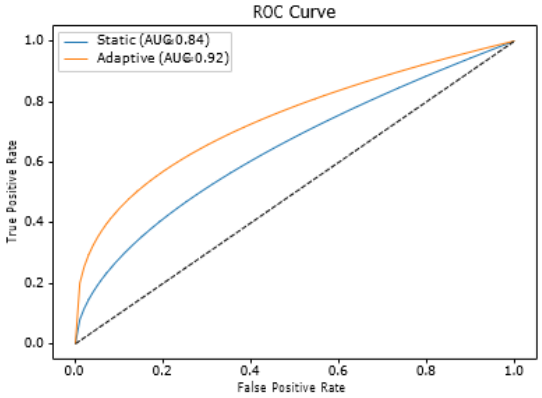

Fig. 8: ROC curve comparison

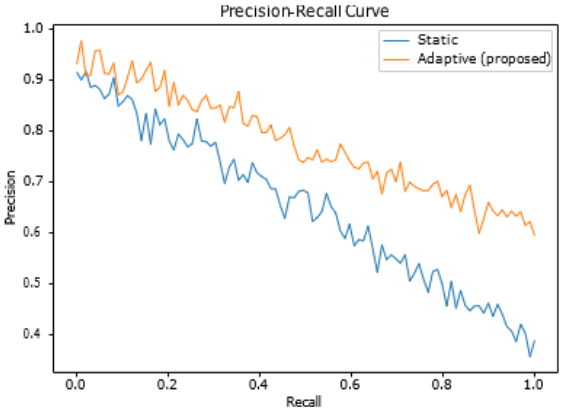

Fig. 9: Precision–Recall curve comparison

G. Throughput under Attack and Scalability Analysis

The fig. 10 displays output with an increasing attack rate. By selectively reducing malicious flows while preserving legitimate traffic, the adaptive approach maintains higher throughput[11],[17].

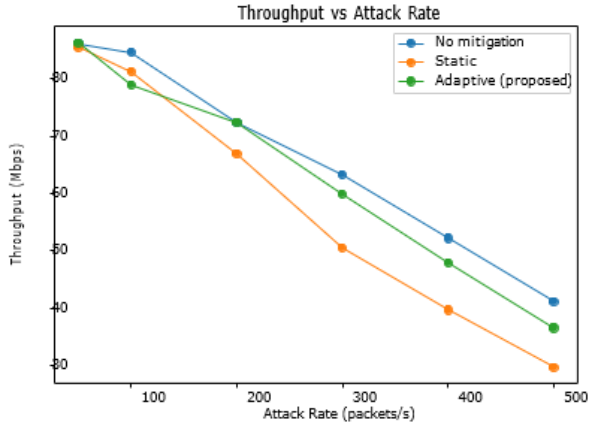

Fig. 10: Throughput versus attack rate

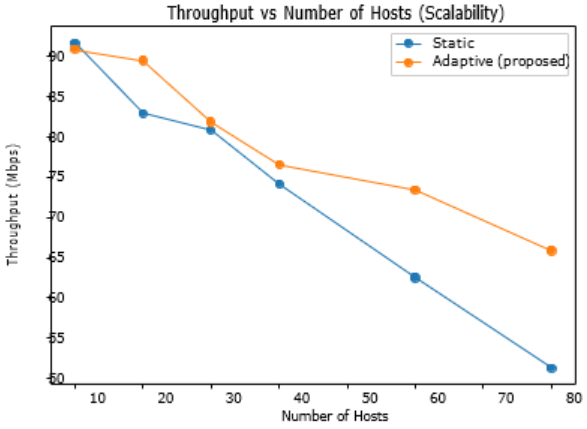

Fig. 11: Throughput versus number of hosts (scalability)

Fig. 11 shows the relationship between throughput and host count. Scalability goals for larger deployments are supported by the adaptive strategy, which degrades more gracefully as network size grows [9].

6. DISCUSSION

According to the experimental results, adaptive, controller-aware mitigation achieves better than inert mitigation in all assessed metrics. Throughput, response time, latency, and control plane excess are all concentrated by the suggested framework's integration of attack severity assessment with controller health monitoring. These results are constant with recent studies [1], [9], [17] that best part controller-aware defense and scalable mitigation in SDN.

7. LIMITATIONS

The suggested adaptive DDoS prevention framework showed encouraging results, but there are a few drawbacks that should be noted. First, the assessment may not precisely reflect the heterogeneity of actual deployments because it is carried out in a simulated environment. Next, while large SDN networks may use multi-controller architectures, the framework assumes a single-controller arrangement. Third, sneaky low-rate attacks that mimic authentic traffic might not be noticed by pre-established thresholds. Finally, the study only looks at volumetric flooding attacks; multi-vector and application-layer attacks are not covered.

8. CONCLUSION AND FUTURE WORK

In this paper, we presented a controller-aware decision-making-focused adaptive DDoS prevention outline for Software Defined Networks. The suggested framework modifies its answer according to the detected attack severity and controller performance conditions rather than using fixed mitigation rules. Frequent explanations of controller overload when static mitigation techniques were used in SDN environments helped as the impetus for this design decision. The adaptive approach lowers controller CPU utilization, restricts packet-in events, and boosts mitigation response time, according to experimental results from SDN emulation.

The adaptive method improves mitigation response time, limits packet-in events, and lowers controller CPU utilization, according to experimental results from SDN emulation. Furthermore, when related to static mitigation approaches, developments in end-to-end latency and throughput were noted, specifically under high attack intensity and larger network sizes.

The framework will be extended to multi-controller SDN architectures in future work, and learning-based techniques to automatically advance severity thresholds will be investigated. To improve the applicability of the advised framework, more research into low-rate and application-layer DDoS attacks will be taken into attention.

References

1. H. Wang, Y. Li, and X. Chen, “Controller-aware DDoS mitigation in software-defined networks,” IEEE Access, vol. 10, pp. 34211–34225, 2022.
2. M. Shafiq, X. Yu, A. A. Laghari, and Y. Wang, “Explainable artificial intelligence-based intrusion detection system,” IEEE Access, vol. 10, pp. 112450–112465, 2022.
3. G. Ajaeiya, N. Parvez, and A. Mahmoud, “Hybrid deep learning framework for DDoS detection in SDN,” Future Generation Computer Systems, vol. 128, pp. 99–111, 2022.
4. H. Polat, O. Polat, and S. Cetin, “Scalable SDN-based DDoS mitigation using adaptive thresholds,” Journal of Network and Computer Applica- tions, vol. 201, 2022.
5. X. Li, Y. Chen, and K. Li, “Efficient DDoS detection for SDN using flow-level features,” IEEE Transactions on Network and Service Man- agement, vol. 19, no. 3, pp. 2545–2558, 2022.
6. M. Hassan, R. Iqbal, and S. Hussain, “Flow-based intrusion detection in software-defined networks,” IEEE Access, vol. 10, pp. 78511–78524, 2022.
7. M. Rahman, S. Islam, and T. Alam, “Hybrid machine learning approach for DDoS detection in SDN,” IEEE Access, vol. 11, pp. 45612–45628, 2023.
8. F. Alharbi, A. Alzahrani, and M. Rehman, “Deep learning-based DDoS attack detection in SDN environments,” IEEE Access, vol. 11, pp. 72890–72905, 2023.
9. Y. Hu, J. Zhang, and L. Wang, “Secure multi-controller software-defined networks against DDoS attacks,” IEEE Access, vol. 11, pp. 41230– 41245, 2023.
10. P. Singh, A. K. Luhach, and D. K. Sharma, “DDoS attack detection and mitigation in SDN: A comprehensive study,” Computers & Security, vol. 125, 2023.
11. S. Alqahtani and M. Alenazi, “Mitigating DDoS attacks in SDN using intelligent rate control,” Computer Networks, vol. 222, 2023.
12. R. Bhatia, A. Verma, and S. Jain, “DDoS defense mechanisms in SDN- enabled IoT networks,” IEEE Internet of Things Journal, vol. 10, no. 12, pp. 10211–10225, 2023.
13. Q. Yaseen, M. Alsharif, and H. Kim, “Scalable intrusion detection in SDN environments,” Ad Hoc Networks, vol. 144, 2023.
14. A. Mehmood et al., “Lightweight DDoS detection for programmable networks,” Sensors, vol. 23, no. 5, 2023.
15. A. Verma and P. Tripathi, “Adaptive intrusion detection for SDN environments,” Journal of King Saud University – Computer Sciences, vol. 36, no. 2, 2024.
16. S. Kumar, R. Gupta, and M. Singh, “Explainable deep learning for DDoS attack detection,” Expert Systems with Applications, vol. 237, 2024.
17. Y. Zhou, X. Huang, and L. Chen, “Real-time DDoS mitigation in software-defined networks,” IEEE Access, vol. 12, pp. 55841–55856, 2024.
18. R. Sharma, P. Mehta, and S. Verma, “Controller-aware intrusion detec- tion for scalable SDN,” Computers & Security, vol. 138, 2025.
19. L. Chen, Y. Zhang, and M. Luo, “Scalable DDoS mitigation framework for SDN,” Future Generation Computer Systems, vol. 149, 2025.
20. H. Liu, X. Wang, and Z. Li, “Adaptive deep learning-based DDoS detection,” Expert Systems with Applications, vol. 243, 2025.
21. S. Ahmed, M. Ali, and F. Khan, “DDoS detection in SDN-based IoT networks using hybrid learning,” IEEE Internet of Things Journal, vol. 12, no. 3, 2025.