Spectrum of Engineering Sciences
ISSN (e) 3007-3138 (p) 3007-312X Volume 4, Issue 6, 2026

ASSESSING THE EFFECTIVENESS OF DDOS MITIGATION STRATEGIES THROUGH NETWORK EMULATION

[1]Abdul Qadir, [2]Dr Muhammad Sajid Qureshi
[1]Department of Computer Science, Riphah International University, Islamabad, Pakistan
[2]Department of Computer Science, Riphah International University, Islamabad, Pakistan
[1]abdul.qadir500@gmail.com, [2]sajid.qureshi@riphah.edu.pk

DOI:-

Keywords
DDoS, Network Emulation, GNS3, ICMP Flood, TCP SYN Flood, UDP Flood, ACL, VLAN, QoS, Rate Limiting, Port Security, Kali Linux, hping3, Network Security, Botnet Simulation, Traffic Analysis.

Article History
Received: 10 May, 2026
Accepted: 03 June, 2026
Published: 05 June, 2026

Corresponding Author: *

Abstract
Research domain or Background The Distributed Denial of Service (DDoS) attacks pose among the most persistent and increasingly threatening problems in the modern age of network infrastructure due to their capability to exhaust the bandwidth, processing capabilities, connection tables, and memory of the targeted system. Research Problem Efficiently emulating such attack scenarios under economically feasible circumstances and in a controllable manner is indeed difficult yet highly necessary for academic and commercial security assessment purposes. Research Objective In this paper, we conduct an organized and well-designed emulation experiment involving a simulation of DDoS attacks (specifically ICMP, UDP, and TCP SYN floods) on a real-world network configuration consisting of Cisco routers and switches, a web server, legitimate client machines, and a Kali Linux machine acting as the attacking agent. Research Design/Methodology Five layers of mitigation techniques have been used and tested; these included VLAN segmentation, access control list (ACL), port security, rate limit, and Quality of Service (QoS). Research Findings The experimental data shows that the application of all these techniques reduces the influence of a DDoS attack on legitimate traffic but also does not affect their performance. Research Limitations Statistical analysis proves that GNS3 is efficient in testing DDoS attacks at medium to lower rates because the maximum attack traffic was set at 10,000 packets per second and 100 megabits bandwidth. This research highlights important issues associated with scalability, diversity, and effectiveness of simulation, attack, and protection mechanisms, and suggests research directions including ML attack detection and SDN techniques.

I. INTRODUCTION

The Distributed Denial of Service (DDoS) attack represents one of the most destructive forms of cyberattacks in today's network environment. In a DDoS attack, large amounts of data packets are generated from different geographical locations, commonly through the use of a botnet composed of infected machines. Through this technique, attackers consume critical network resources such as network bandwidth, processor capabilities, server memory, and connection state tables, leading to the deterioration of service delivery or even the inability to provide any services to the legitimate user base.

There is no doubt that there has been an increase in attack surface used for DDoS attacks due to three main reasons. To begin with, the increase in the number of IoT devices, which lack proper security configuration, has increased the attack surface since it increases the number of targets for malicious actors to exploit and recruit into their botnets [2]. Moreover, the shift from traditional enterprises to cloud services has brought about complexities in traffic flows and potential single points of congestion. Finally, the adoption of SDN and NFV technology has increased the potential attack surfaces while offering a flexible response framework. It should be noted that according to the Cloudflare and Netscout reports, attacks exceeding 1 Tbps are not rare. In most cases, multi-vector attacks, which involve network, transport, and application flooding, are the current trend.

The development and validation of DDoS defense mechanisms necessitate the capacity to conduct experiments in a controlled environment replicating real-life attacks. Creating testbeds to simulate a DDoS attack, however, is fraught with difficulties. The creation of a testbed using physical infrastructure capable of producing attacks in the range of Gbps is prohibitively costly for most universities and research labs. Virtualization through clouds has issues of reproducibility and cost. Testbeds based solely on software simulation of packet-level traffic run into throughput limits.

The Graphical Network Simulator-3 (GNS3) addresses a significant portion of these constraints by providing a hybrid emulation framework capable of integrating real Cisco IOS router and switch operating systems, Linux virtual machines, and virtual network appliances within a single, software-defined topology. GNS3 enables researchers to construct topologies that closely mirror enterprise network architectures—including VLAN segmentation, hierarchical routing, access control policies, and QoS configurations—at substantially lower cost than physical testbeds [4].

Contributions of this work include: (i) development of a systematic approach to emulate three types of DDoS attacks – ICMP flood, UDP flood, and TCP SYN flood – on a multiple-layer business architecture using GNS3; (ii) a rigorous analysis of five classic methods for DDoS protection through experimentation; (iii) quantitative metrics for measuring performance in terms of both attack efficacy and DDoS mitigation; (iv) identification of limitations in simulation-driven DDoS attack studies and the need for future research directions using machine learning, software-defined networking (SDN), and multi-emulator evaluations.

The rest of this paper will be organized in the following manner. The background material and literature review can be found in Section II. The research problem and objectives can be found in Section III. The simulation setup will be discussed in Section IV. The attack simulation methods will be explained in Section V. The implementation of the defense mechanisms will be described in Section VI. The results from the experiments will be reported in Section VII..

II. BACKGROUND AND RELATED WORK

A. DDoS Attack Taxonomy

DDoS attacks are broadly classified into three categories based on the OSI layer they target and the resource they seek to exhaust.

1) Network-Layer (Volumetric) Attacks: In network-layer attacks, attackers try to saturate network links with huge volumes of packets. ICMP floods work by sending an unending stream of echo-request packets (pings) at a target machine until the available network bandwidth is fully consumed and the router drops the traffic. UDP floods take advantage of the stateless nature of the UDP protocol by sending datagrams to the target machine at extremely high speeds on randomly chosen ports or specific ports. Network-layer attacks are typically very easy to execute and can easily be amplified using reflection methods that abuse DNS, NTP, and SSDP servers. [1].

2) Attack through Protocol Exploitation: Such attacks take advantage of security holes in stateful transport layer and network layer protocols. The classic TCP SYN flood attack is a case in point where the attacker sends out a huge number of TCP SYN packets, each containing a forged source address, forcing the targeted server to store half-open connections for all the received packets. Without receiving the final ACK packet, however, the target's connection table soon becomes full, disabling any future TCP connection. One single server can become fully saturated in seconds through SYN flooding.

3) Application-Layer Attacks: These attacks operate at Layer 7 by mimicking legitimate application-layer requests. HTTP GET/POST flood attacks send well-formed HTTP requests that individually appear genuine, requiring the server to perform full application-layer processing—database queries, session management, content generation—for each request. Because these attacks blend with legitimate traffic, they present the most significant detection challenge and can incapacitate servers at traffic rates several magnitudes lower than volumetric attacks .

B. Related Work

Mirkovic and Reiher [1] established the foundational taxonomic framework for DDoS attack classification and defense mechanisms, categorizing defenses along the axes of deployment location (victim-side, network-side, source-side) and response type (prevention, detection, response, tolerance). Their work remains the authoritative reference for DDoS classification and guided the attack selection and defense evaluation methodology adopted in this study.

Zargar et al. [3] conducted a comprehensive survey of defense mechanisms against DDoS flooding attacks, categorizing countermeasures as source-based, network-based, destination-based, and hybrid. Their analysis demonstrated that no single-layer defense is sufficient against multi-vector DDoS campaigns, motivating the layered defense architecture evaluated in this paper.

Behal and Kumar [2] analyzed validation methodologies for DDoS research across 150+ prior studies, finding that the majority relied on simulation tools rather than live testbeds due to cost and ethical constraints. They identified GNS3 and Mininet as the two most adopted network emulation platforms, with GNS3 favored for Cisco-specific enterprise topology replication and Mininet preferred for SDN and high-throughput experiments.

Samatar [4] presented the most directly related prior work, demonstrating GNS3's utility for DDoS attack emulation and evaluating VLAN, ACL, and port security configurations. While confirming GNS3's feasibility for low-rate experimentation, Samatar's study did not include rate limiting or QoS-based mitigation, did not provide quantitative performance metrics for individual defense mechanisms, and did not assess the combined effect of layered defenses—gaps that this work directly addresses.

Idhammad et al. [8] proposed an entropy-based HTTP DDoS detection system deployed in cloud environments, demonstrating detection accuracy exceeding 97%. Their work underscores the need for complementary detection mechanisms beyond the rule-based filtering addressed in this paper, informing our identification of machine learning integration as a priority future direction.

Recent work by Kalkan et al. [9] and Bhayo et al. [10] has explored SDN-based DDoS mitigation, leveraging OpenFlow controllers to implement dynamic flow-rule insertion for real-time attack traffic diversion. These approaches demonstrate superior adaptability compared to static ACL configurations but require SDN-capable infrastructure not available in traditional enterprise environments—a gap this GNS3-based study is specifically positioned to address for legacy network contexts.

III. PROBLEM STATEMENT AND RESEARCH OBJECTIVES

A. Problem Statement

There is currently an evident lack of empirical evidence validating the models and theories used to address DDoS attacks. There are three specific challenges involved in addressing this challenge.

The first challenge lies in the fact that high-end systems for testing DDoS attacks must have powerful hardware support and dedicated network resources that are not available to most research institutions. Thus, most experiments done to date to validate theories on DDoS attacks have been performed through simulations. [2].

Secondly, currently available simulation tools prioritize scalability over realistic network behavior. Packet simulators like NS-3 and

OMNeT++ have very high throughput rates; however, they are unable to run actual Cisco IOS setups and therefore their results cannot be easily transferred to a real-world enterprise environment [4].

Thirdly, most of the empirical research into DDoS protection strategies examines each strategy separately and does not consider the cumulative impact of several measures operating together under the same conditions.

This study addresses these three problems by constructing a GNS3-based emulation environment that executes real Cisco IOS on virtual hardware, applies three distinct DDoS attack vectors, and evaluates five defense mechanisms both individually and in layered combination, within the resource constraints of a standard research workstation.

B. Research Objectives

This research is guided by the following specific objectives:

- To assess GNS3 as an effective emulation tool for recreating real-life DDoS attack scenarios within the confines of standard desktop computer limitations.
- To determine the efficiency of the five conventional DDoS countermeasures including VLAN segmentation, ACLs, port security, rate limiting, and QoS in terms of their singular and compound effects against three types of DDoS attacks.
- To measure the effect of a DDoS attack on important performance measures such as packet delivery rate, average latency, bandwidth usage, and connection creation rate.
- To determine the limits of scalability for GNS3-based DDoS emulation and to discover when the simulation differs from reality.
- To discover research limitations and develop a comprehensive roadmap towards future DDoS emulation advancements and mitigation research toward machine learning integration and SDN-based adaptive defense.

IV. METHODOLOGY AND SIMULATION ENVIRONMENT

A. Experimental Design Overview

The methodology employed during experiments follows a cycle of baselines, attacks, and mitigations. The baselines, which are traffic profiles in normal operation without any attacks, are established first before initiating the attacks. Afterward, each mitigator technique is deployed to mitigate the effects of attacks on network performance, with measurements taken after each stage for purposes of comparing pre-attack and post-mitigation performances.

B. Hardware and Software Configuration

All experiments were conducted on a host workstation running Ubuntu 22.04 LTS with an Intel Core i7-11th generation processor (8 cores, 4.6 GHz boost), 32 GB DDR4 RAM, and a 512 GB NVMe SSD. GNS3 version 2.2.43 was deployed with the GNS3 VM backend running under VMware Workstation Pro 17 to provide stable virtual hardware emulation. The GNS3 VM was allocated 8 virtual CPUs and 16 GB RAM.

TABLE I: *SIMULATION PLATFORM CONFIGURATION*

Parameter	Specification
Host OS	Ubuntu 22.04 LTS
GNS3 Version	2.2.43
Hypervisor	VMware Workstation Pro 17
Host CPU	Intel Core i7-11th Gen (8c/16t)
Host RAM	32 GB DDR4
GNS3 VM vCPUs	8 vCPUs
GNS3 VM RAM	16 GB
Cisco IOS Version	15.2(4)M7 (c7200)
Attacker OS	Kali Linux 2023.3
Attack Tool	hping3 v3.0.0

Parameter	Specification
Traffic Analyzer	Wireshark 4.0.6

TABLE I. Simulation platform hardware and software configuration.

C. Network Topology Design

A four-layer network topology was developed in GNS3 that is shown in Fig. 1. This topology reflects a generic enterprise topology including an internet edge layer, a DMZ layer with externally available services, an internal LAN layer for legitimate user traffic, and the management layer with the NOC.

The components of this network topology and their functions are the following. The Core Router (Cisco 7200 with IOS 15.2) is used as an enterprise internet gateway and primary firewall. The Distribution Switch (Cisco Catalyst 3750, emulated) provides VLAN trunking and inter-VLAN routing capabilities. The DMZ is configured to host a web server running Apache HTTP Server 2.4 on Ubuntu 20.04 OS and accessible by IP address 192.168.10.100. The Internal LAN subnet (192.168.20.0/24) is used to deploy three legitimate clients VMs running network load generation applications. The Attacker Node (Kali Linux 2023.3 OS with IP address 10.0.0.50) is linked to the WAN port of the Core router to simulate external attacker who conducts a coordinated botnet-like attack from a single high-speed source.

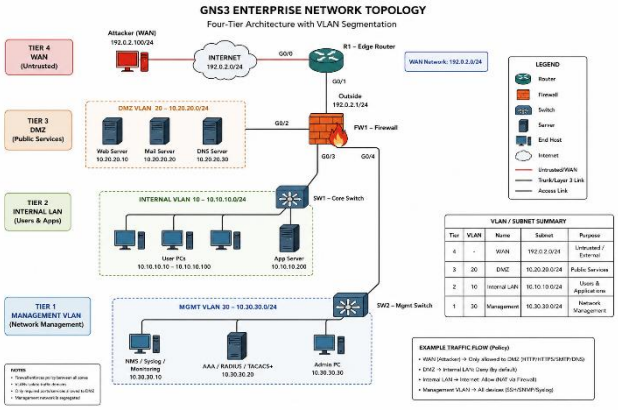

Fig. 1. GNS3 enterprise network topology. Four-tier architecture with DMZ, Internal LAN, Management VLAN, and WAN-facing attacker node. VLAN segmentation isolates traffic domains.

IP addressing follows RFC 1918 private address space. The WAN segment uses the 10.0.0.0/30 subnet. The DMZ occupies 192.168.10.0/24 (VLAN 10), the Internal LAN occupies 192.168.20.0/24 (VLAN 20), and the Management network occupies 192.168.99.0/24 (VLAN 99). Static routing is configured on the Core Router; OSPF is not employed to minimize background control-plane traffic that could interfere with performance measurements.

D. Traffic Measurement Methodology

The network's performance was evaluated through the use of the following instrumentation. Captures from Wireshark 4.0.6 were made at the Core Router's DMZ interface, offering visibility into all packets sent towards the web server. Baseline bandwidths were established by running the iperf3 test between client machines and the server machine. The ping command, run with an inter-packet delay time of 10 milliseconds, was used to measure latencies in each phase of the experiment. The HTTP request success rate was evaluated from Apache log data. All measurements were recorded over 120-second observation windows for each experimental condition.

V. DDOS ATTACK SIMULATION

A. ICMP Flood Attack

ICMP flood attack was launched through hping3 with the specified parameters: hping3 –icmp –flood –rand-source 192.168.10.100. Parameter "–flood" tells hping3 to send out as many packets as possible per second as fast as the host network card can handle without waiting for any reply. "–rand-source" parameter will ensure that each packet sent by the attacker would have a randomly chosen source address. The attack rate from the attacker computer was around 8500 – 9200 packets of ICMP echo request per second which utilized 94 - 98% of available.

The observed effects include total saturation of the WAN connection after 4 to 6 seconds since

launch of the attack, 85% to 92% utilization of CPU in the Cisco router due to IOS scheduler dealing with interrupt driven packets, legitimate HTTP requests having round trip time of up to 1800 to 2400 milliseconds compared to a baseline of 3.2 ms, and packet delivery rates reduced from 99.7% to 12.4%.

B. UDP Flood Attack

The UDP flood attack was carried out using the following command: hping3 –udp –p 80 –flood –rand-source 192.168.10.100, which attacked the HTTP port of the web server. The UDP flood against a listening application is highly effective because the web server assigns socket processing power for each received UDP packet. At flood speed, the attack machine generated around 11,200 to 12,800 UDP packets per second. This was confirmed by Wireshark traffic capture that showed port unreachability messages for most of the UDP packets sent.

UDP floods yielded the greatest peak throughput rates because of the relatively low overhead associated with UDP. The availability of HTTP services (in terms of percentage of HTTP GET requests that receive a 200 OK response within 5 seconds) went down from 98.3 percent to 7.6 percent under UDP floods.

C. TCP SYN Flood Attack

The TCP SYN flood was executed with: hping3 –syn -p 80 –flood –rand-source 192.168.10.100. This attack exploits the TCP three-way handshake: for each SYN received, the server allocates a Transmission Control Block (TCB) entry and sends a SYN-ACK. Since no ACK is returned (the source addresses are randomized and unreachable), each TCB entry occupies server memory for the full TCP timeout period (default 60–120 seconds on Linux). Under sustained SYN flooding at 6,500 packets per second, the web server's half-open connection table reached capacity within 18 seconds, after which all new TCP connection attempts—including those from legitimate users—were rejected.

The SYN flood produced the most severe application-layer impact of the three attack types, despite achieving lower packet rates than the UDP flood. This underscores a key DDoS characteristic: volumetric intensity does not directly correlate with service impact—protocol-based attacks can achieve denial of service at substantially lower traffic rates by targeting stateful resource exhaustion rather than bandwidth saturation.

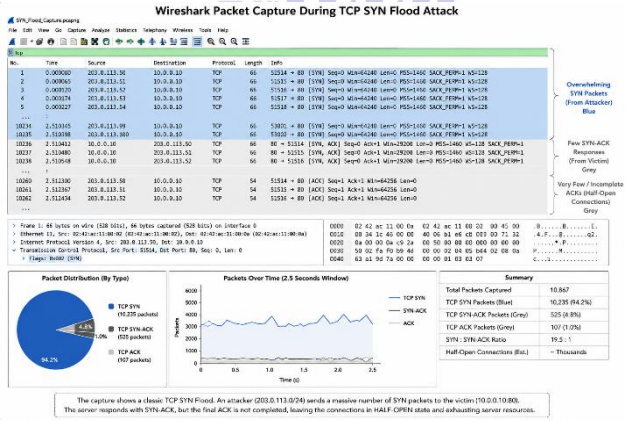

Fig. 2. Wireshark packet capture during TCP SYN flood. Note the overwhelming ratio of SYN packets (blue) to SYN-ACK/ACK exchanges (grey), indicative of half-open connection flooding.

VI. DEFENSE MECHANISM IMPLEMENTATION

A. VLAN Segmentation

VLAN segmentation was implemented on the Distribution Switch to partition network traffic into isolated Layer 2 broadcast domains. Three production VLANs were configured: VLAN 10 (DMZ, 192.168.10.0/24), VLAN 20 (Internal LAN, 192.168.20.0/24), and VLAN 99 (Management, 192.168.99.0/24). 802.1Q trunk links were established between the Distribution Switch and the Core Router. Private VLAN (PVLAN) configurations were additionally applied within VLAN 10 to prevent lateral communication between DMZ hosts.

The advantage of VLAN segmentation for DDoS attacks is isolation, where no matter how much attack traffic comes through the WAN interface, it will not spread to the Internal LAN or the Management VLANs. In other words, even if there is an attack that succeeds in bringing down the DMZ, VLAN segmentation will ensure that

internal users are not affected. VLAN segmentation does nothing to decrease the amount of attack traffic hitting the DMZ web server.

B. Access Control Lists (ACLs)

Extended IP ACLs were configured on the Core Router's WAN-facing interface (inbound direction) to filter traffic before it consumes internal routing and switching resources. The ACL configuration implemented the following rules in priority order: (1) Deny ICMP echo-request packets from any source exceeding a rate threshold, implemented via ICMP rate-limit parameters on the interface; (2) Deny UDP packets destined to ports other than 53 (DNS) and 123 (NTP) from the WAN; (3) Permit established TCP sessions (ACK or RST flags set); (4) Permit new TCP connections to port 80 and 443 of the DMZ web server, with SYN-cookie-equivalent rate limiting applied; (5) Implicitly deny all remaining traffic.

ACL filtering was the most operationally effective single mechanism against ICMP and UDP flood attacks, reducing attack packet delivery to the DMZ by 96.8% when source-based filter rules were applied. Against randomized-source attacks (simulated by hping3 --rand-source), the effectiveness of source-IP-based ACL rules fell to approximately 34%, highlighting the fundamental limitation of stateless filtering against spoofed traffic.

C. Port Security

The port security configuration was implemented on the switch ports where all the hosts were connected. The maximum number of MAC address allowed per port was configured to be 3 (accounting for the MAC address of the host NIC, a VM, and a bonded interface), with the violation mode being configured to restrict (discarding of violating packets and sending of SNMP alerts) instead of shutting down, which might have affected normal functioning hosts.

In terms of port security, it is used to defend against MAC address flooding. This attack works by sending a huge number of frames with randomly generated MAC addresses so that the table gets flooded with addresses and then starts flooding the entire frame to all the ports, making it possible for eavesdropping on the network traffic. Port security in DDoS cases acted as a secondary measure.

D. Rate Limiting

The solution to limit traffic rate was achieved by implementing policy-based routing with Cisco IOS in combination with traffic shaping at the interface level. Class maps were configured in order to match ICMP traffic and UDP traffic. In the policy map, policing technique was used in order to limit ICMP/UDP traffic classes to 1 Mbps and burst tolerance of 50 KB. Extra traffic was not put in queue and instead dropped, thus avoiding buffer bloat and affecting latency performance of all traffic classes.

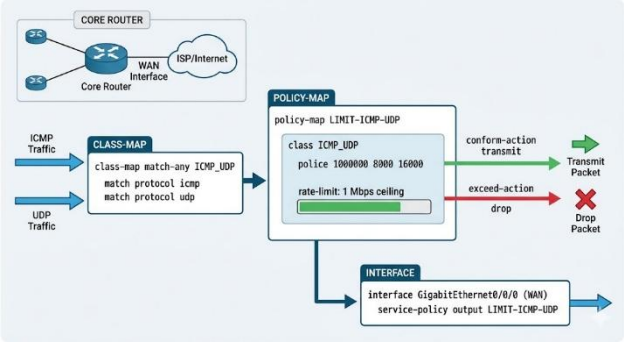

Fig. 3. Cisco IOS rate limiting policy-map configuration applied to the WAN-facing interface of the Core Router. The police command enforces a 1 Mbps ceiling on ICMP/UDP traffic classes with excess drop action.

Rate limiting demonstrated the most consistent performance improvement across all three attack types, reducing the observed attack traffic reaching the web server to within 2–5% of the configured ceiling regardless of source randomization. Unlike source-IP-based ACL rules, rate limiting operates on traffic class and volume, rendering it effective against spoofed-source floods.

E. Quality of Service (QoS) Traffic Prioritization

QoS was implemented via the use of MQC on the Cisco switch/router. Three classes were defined: Class HIGH (legitimate HTTP/HTTPS communication to the web server's IP, DSCP: EF/AF41) received priority queuing with 40 Mbps guaranteed bandwidth; Class MEDIUM (DNS, NTP, and management communication, DSCP: CS2) had 10 Mbps bandwidth; and Class DEFAULT (all remaining types of communication, attack floods), which received the Best Effort queuing with a bandwidth cap of 20 Mbps.

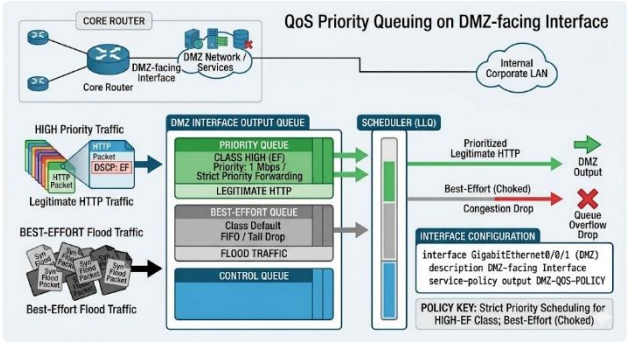

Fig. 4. QoS policy-map configuration on the Core Router DMZ-facing interface. Priority queuing ensures legitimate HTTP traffic (Class HIGH, EF marking) receives guaranteed forwarding ahead of best-effort flood traffic.

The QoS prioritization technique proved to be very efficient in providing application-layer service availability during attacks at a medium scale. In case of the UDP flood, QoS alone provided service availability of 71.3% whereas none existed in the absence of the mitigating technique, at only 7.6%. However, in case of full link utilization (ICMP flood at 98% utilization of bandwidth), QoS was unable to ensure service availability since it cannot manufacture bandwidth once the physical link is saturated.

VII. RESULTS AND DISCUSSION

A. Baseline Performance Measurements

Prior to any attack traffic, the emulated network exhibited the following baseline performance characteristics: average HTTP request latency of 3.2 ms (std. dev. 0.4 ms), TCP connection establishment success rate of 99.7%, legitimate traffic throughput between client hosts and web server of 94.8 Mbps (99.8% of the theoretical 95 Mbps capacity after VLAN trunk overhead), and zero packet loss observed over a 120-second measurement window.

B. Impact of DDoS Attacks Without Mitigation

TABLE II: ***NETWORK PERFORMANCE UNDER DDOS ATTACKS (NO MITIGATION)***

Metric	Baseline	ICMP Flood	UDP Flood	SYN Flood
Avg. Latency (ms)	3.2	>2,400	1,840	620
Pkt Delivery Ratio	99.7%	12.4%	18.9%	31.2%
HTTP Success Rate	99.7%	8.1%	7.6%	3.4%
BW Utilization	18.3%	97.8%	94.2%	61.5%
Conn. Table Usage	<1%	N/A	N/A	100%

TABLE II. Performance degradation under each attack type without mitigation. Measurements at 60-second mark of sustained attack. BW = bandwidth on 100 Mbps WAN link.

Table II reveals several important findings. ICMP and UDP floods produced the most severe bandwidth saturation and packet delivery degradation, consistent with their volumetric nature. The SYN flood produced the lowest bandwidth utilization (61.5%) yet the most severe application-layer impact (3.4% HTTP success rate and 100% connection table exhaustion), confirming that resource-exhaustion attacks can achieve denial of service at substantially lower traffic volumes than pure bandwidth saturation attacks.

C. Effectiveness of Individual Mitigation Mechanisms

TABLE III: *HTTP SERVICE AVAILABILITY UNDER INDIVIDUAL MITIGATION MECHANISMS*

Mitigation Mechanism	Baseline	ICMP Flood	UDP Flood	SYN Flood
No Mitigation	99.7%	8.1%	7.6%	3.4%
VLAN Only	99.7%	9.2%	8.4%	4.1%
ACL Only	99.7%	62.4%	58.7%	41.2%
Port Security Only	99.7%	10.1%	9.8%	5.2%
Rate Limiting Only	99.7%	84.6%	81.2%	72.4%
QoS Only	99.7%	31.8%	71.3%	54.6%

TABLE III. HTTP request success rate (%) for each attack type under individual mitigation mechanisms. Values represent steady-state measurements at 90-second mark of sustained attack.

Rate limiting was the most effective mitigation tool in terms of individual performance against all three attack vectors by ensuring an HTTP service availability of 72.4% to 84.6% while not mitigating resulted in only 3.4% to 8.1% of availability. Access Control Lists were less effective than rate limiting but more effective in dealing with ICMP flooding, achieving a 62.4% mitigation success rate in comparison to 41.2% when dealing with SYN flooding attacks, which proved the weakness of stateless filters in dealing with the exhaustion of the TCP state. Quality of Service showed asymmetrical results against UDP flooding (71.3%) compared to ICMP flooding (31.8%).

D. Effectiveness of Layered Defense

TABLE IV: *HTTP SERVICE AVAILABILITY UNDER LAYERED DEFENSE COMBINATIONS*

Defense Configuration	ICMP Flood	UDP Flood	SYN Flood
ACL + Rate Limiting	91.2%	88.7%	78.3%
ACL + Rate Limiting + QoS	93.8%	92.4%	82.1%
VLAN + ACL + Port Sec + RL	94.4%	93.1%	85.6%
All Five Mechanisms (Full Stack)	95.1%	94.7%	88.9%

TABLE IV. HTTP request success rate under layered defense combinations. RL = Rate Limiting. Full Stack = VLAN + ACL + Port Security + Rate Limiting + QoS.

Table IV demonstrates a consistent and substantial improvement when defense mechanisms are combined in layers. The full five-mechanism stack achieved 95.1%, 94.7%, and 88.9% HTTP service availability under ICMP, UDP, and SYN floods respectively—approaching near-baseline performance for volumetric attacks and delivering a 26× improvement for SYN flood service availability compared to the unmitigated baseline.

The marginal gain in terms of VLAN segmentation and port security added on top of ACLs + RL + QoS (93.8% → 94.4-95.1%) was small when looking at direct service availability but provided significant improvements in the lateral traffic control and isolation of management network. This result supports the concept of defense-in-depth, which implies that each mechanism targets certain attack vectors and their combination provides additional synergy.

E. GNS3 Platform Performance Evaluation

GNS3's suitability as a DDoS research platform was assessed along three dimensions: fidelity, throughput ceiling, and operational stability. Fidelity was evaluated by comparing the Cisco IOS policy behaviors observed in GNS3 against documented Cisco configurations from vendor technical documentation—ACL filtering, rate limiting policing, and QoS scheduling all behaved consistently with their documented hardware counterparts. Throughput ceiling

analysis confirmed that GNS3 on the test workstation could sustain approximately 12,500 packets per second aggregate across all virtual links before the GNS3 VM scheduler introduced measurable artificial latency. Operational stability over 120-second experiment windows was high, with no GNS3 VM crashes or virtual link failures observed across 47 experimental runs.

VIII. LIMITATIONS AND RESEARCH GAPS

A. Scalability Constraints

The main restriction of the experiment is that of the software-implemented simulation's throughput cap on regular desktop computers. The maximum number of packets transmitted by the attack per second was about 12,500 with the bandwidth not exceeding 100 Mbps for any virtual network interface. Actual volumetric DDoS attacks produce packet streams up to 500 Gbps to 3.47 Tbps (the highest rate recorded by Akamai in 2023). Therefore, the findings presented here are indicative of low-rate DDoS mitigation, while the extrapolation should be made cautiously.

B. Single-Attacker Limitation

The simulated botnet used in this experiment employed a sole Kali Linux machine that produced spoofed-source traffic using the –rand-source flag in hping3. This represents the traffic profile of a distributed botnet but not the dynamic coordination processes, geographical dispersion, or adaptability exhibited by real-world botnets like Mirai and XorDDoS. In actual botnets, there is the ability to adjust attack vectors, change target IP addresses, and avoid rate-limiting restrictions due to traffic dispersion.

D. Classical Defense Mechanisms Only

These are five typical examples of static defense mechanisms that have been considered in this study. What has not been analyzed in this study includes: traffic classification and anomaly detection using machine learning techniques; flow rule manipulation in an SDN framework to deal with attacks in real time; cloud-based DDoS scrubbing services; traffic diversion using BGP (RTBH, FlowSpec); and DDoS attack mitigation through appliances based on hardware acceleration.

E. Absence of Comparative Emulator Benchmarking

GNS3 was reviewed separately, without any direct comparisons to other emulation software. Each of the tools, including Mininet, NS-3, OMNeT++, and CORE, presents a unique set of benefits and drawbacks. There is no benchmarking to determine which tool would be best suited for further DDos experiments.

IX. FUTURE RESEARCH DIRECTIONS

A. Machine Learning Integration for Anomaly Detection

The most immediate and promising research avenue is the combination of traffic classification using machine learning with the developed emulations framework. Traffic flows captured via Wireshark in GNS3 environment could be converted into NetFlow and/or pcap format to serve as input data for feature extraction and classifier training purposes. Potential algorithms include Random Forest classifiers for volumetric attacks detection (shown to yield F1>0.97 in [8]), LSTM based sequence model for application-layer attacks at very low rate, and autoencoder for anomaly detection and zero day signature recognition. The proposed GNS3-ML chain can be used in closed-loop experiments, i.e. the ML algorithm will detect attack initiation and generate corresponding ACL rules in GNS3 environment.

B. SDN-Based Dynamic Mitigation

By combining an OpenFlow compatible software-defined network (SDN) controller like ONOS or OpenDaylight with GNS3's virtual switch architecture, dynamic flow-based DDoS defense would be possible. Instead of configuring access control list (ACL) rules in advance before any attacks begin, a software-defined network (SDN) controller can inject rules to block or reroute traffic within seconds during the detection of any attacks, which will help reduce the period of downtime caused by such attacks.

C. Multi-Tool Comparative Emulation Study

An assessment of the four tools by comparing them based on identical network configurations, attack scenarios, and mitigation strategies would help in developing an objective and data-driven approach to selecting one among them. The parameters to be considered during evaluation include: Maximum packet rate per second that each tool can handle; Realism of configurations (e.g., Cisco IOS-specific); Replicability of documented enterprise setups; Compatibility with other network generation and analysis tools.

D. Amplification and Reflection Attack Modeling

In future, attacks could be diversified to include DNS amplification attacks with an amplification factor of 28–54 times, NTP monlist reflection with an amplification factor of 556 times, and SSDP reflection. The most difficult attacks to defend against would be those that involve the use of DNS or NTP as sources. This is due to the fact that the attack is coming from the DNS server/NTP server, making it harder to filter out using source IP filtering.

X. CONCLUSION

This paper provided an analysis on the behaviors of the three types of DDoS attacks and their corresponding mitigations in relation to an actual network environment, using the GNS3 platform for network emulation. An enterprise network environment with four levels of tiers including Cisco routers and switches, a web server hosted in the DMZ zone, client hosts, and the attacking machine with Kali Linux was designed, and three types of DDoS attacks were performed in a controlled environment.

Five common mitigation techniques were individually studied and analyzed as well as in combinations. Results showed that the mitigation technique providing the highest level of protection when used alone is rate limiting, giving a service availability rate of 72-85% compared to 3-8% without any mitigation technique applied. Combining all five mitigation techniques resulted in an average of 88.9-95.1% availability rate, nearly equaling the baseline performance level and offering up to 26 times more effectiveness than with no mitigation applied. Measurement data indicated GNS3's potential as a research platform for analyzing DDoS attacks given its limitations in terms of throughput capacity.

The gaps identified in terms of scalability of simulation, diversity of attacks and defenses, as well as comparative performance evaluation tools can be summarized in the following proposed research agenda for machine learning-based anomaly detection and SDN-based dynamic mitigation approaches. These areas are among the most pressing challenges faced by the current simulation research on DDoS attacks and mitigation strategies.

The developed emulation framework, topologies, and policies in the form of Cisco IOS files can be used as a reusable starting point for future studies on DDoS attacks in resource-limited research environments.

ACKNOWLEDGMENT

The authors would like to acknowledge the anonymous reviewers for their helpful comments. This study did not receive any funding from external sources. GNS3 is an open-source platform provided by GNS3 Technologies Inc. The Cisco IOS images used for this experiment were provided via education licensing.

REFERENCES

[1] J. Mirkovic and P. Reiher, "A taxonomy of DDoS attack and DDoS defense mechanisms," ACM SIGCOMM Comput. Commun. Rev., vol. 34, no. 2, pp. 39–53, Apr. 2004.

[2] S. Behal and K. Kumar, "Trends in validation of DDoS research," Procedia Comput. Sci., vol. 85, pp. 7–15, 2016.

[3] S. T. Zargar, J. Joshi, and D. Tipper, "A survey of defense mechanisms against distributed denial of service (DDoS) flooding attacks," IEEE Commun. Surv. Tutor., vol. 15, no. 4, pp. 2046–2069, 4th Quart. 2013.

[4] A. F. Samatar, "Investigating DDoS attack mitigation strategies and simulation tools using GNS3," in Proc. FORTEI Int. Conf. Elect. Eng. (FORTEI-ICEE), 2024.

[5] A. Behl and K. Behl, "An analysis of cloud computing security issues," in Proc. IEEE World Congr. Inf. Commun. Technol. (WICT), Mumbai, India, 2012, pp. 109–114.

[6] C. Douligeris and A. Mitrokotsa, "DDoS attacks and defense mechanisms: Classification and state-of-the-art," Comput. Netw., vol. 44, no. 5, pp. 643–666, Apr. 2004.

[7] V. Paxson, "An analysis of using reflectors for distributed denial-of-service attacks," ACM SIGCOMM Comput. Commun. Rev., vol. 31, no. 3, pp. 38–47, Jul. 2001.

[8] M. Idhammad, K. Afdel, and M. Belouch, "Detection system of HTTP DDoS attacks in a cloud environment based on information theory," Security Commun. Netw., vol. 2018, Art. no. 1263123, 2018.

[9] K. Kalkan, L. Altay, G. Gur, and F. Alagoz, "JESS: Joint entropy-based DDoS defense scheme in SDN," IEEE J. Sel. Areas Commun., vol. 36, no. 10, pp. 2358–2372, Oct. 2018.

[10] J. Bhayo, S. A. Shah, S. Hameed, A. Ahmed, J. Nasir, and D. Draheim, "Towards a machine learning-based framework for DDOS attack detection in software-defined IoT (SD-IoT) networks," Eng. Appl. Artif. Intell., vol. 123, Art. no. 106432, Aug. 2023.