Multi-Layer DDoS Detection and Mitigation Using XDP, P4, and BGP Flowspec

Ayan Shaikh

December 2025

Abstract

This project implements a three-layer Distributed Denial-of-Service (DDoS) defense system combining: (1) XDP for early kernel-level packet filtering, (2) a P4 programmable data-plane for in-network detection, and (3) BGP Flowspec / RTBH for upstream traffic suppression. Using Jetstream2 cloud resources, the system successfully detects high-rate packet floods, drops malicious traffic, and preserves legitimate network behavior. The resulting pipeline closely models the architectures deployed by modern ISPs and cloud security providers.

1 Introduction

DDoS attacks overwhelm network infrastructure by generating extremely high volumes of packets. Conventional perimeter firewalls and software routers are incapable of processing packets at such a rate, leading to saturation and service outages.

Modern mitigation frameworks rely on fast, programmable and layered defenses. This project implements a realistic multi-layer pipeline consisting of:

- **XDP (eXpress Data Path)**: kernel-level filtering operating before the Linux network stack.
- **P4 Program (BMv2 Mininet)**: programmable packet-processing for anomaly detection using counters.
- **BGP Flowspec / RTBH Simulation**: simulates upstream ISP traffic blackholing.

Beyond simple volumetric attacks, modern adversaries increasingly rely on adaptive and multi-vector DDoS strategies that dynamically change packet signatures, ports, and protocols to bypass static filtering. This necessitates a defense architecture capable of real-time programmability and fine-grained detection. Traditional perimeter appliances

often suffer from performance bottlenecks due to interrupt-driven processing, limited rule update speed, and lack of visibility into packet-level telemetry. In contrast, XDP and P4 operate directly in programmable dataplanes, enabling detection and enforcement to occur closer to the packet arrival point. This project demonstrates how combining kernel-level and in-network programmability achieves mitigation latencies that are not feasible with legacy solutions, bringing the design closer to the security pipelines used in modern hyperscale environments.

The goal is to detect and mitigate DDoS traffic as early as possible, preventing downstream congestion.

2 System Architecture Overview

The project implements a three-tier architecture:

1. **Tier 1: XDP Early Packet Drop** XDP runs inside the Linux kernel and filters packets before they enter the network stack. It uses a per-source IP counter to detect repetitive high-rate packets and performs `XDP_DROP`.

2. **Tier 2: P4 In-Switch Detection** A P4 program running in the BMv2 software switch maintains counters per flow, detects threshold violations, and drops packets belonging to suspected DDoS sources.

3. **Tier 3: BGP Flowspec / RTBH Simulation** Traffic destined to the victim IP is matched using Flowspec rules and discarded. An RTBH route simulates upstream blackholing.

The multi-tier architecture is intentionally designed to mirror real-world Internet DDoS pipelines such as those deployed by ISPs, CDN providers, and cloud edge networks. Each layer operates at a different point in the packet lifecycle: XDP filters traffic before ingress into the OS stack, P4 enforces policies within the forwarding plane, and Flowspec/RTBH models upstream cooperation from transit networks. This separation of concerns provides resilience—if one layer becomes saturated or bypassed, the remaining components continue mitigating the attack. The layered design also reduces operational risk by allowing incremental deployment: campus networks may begin with XDP filtering, edge routers may enforce P4 rules, and upstream transit providers may activate RTBH only during confirmed attack windows.

Figure placeholders are included below for diagrams/screenshots.

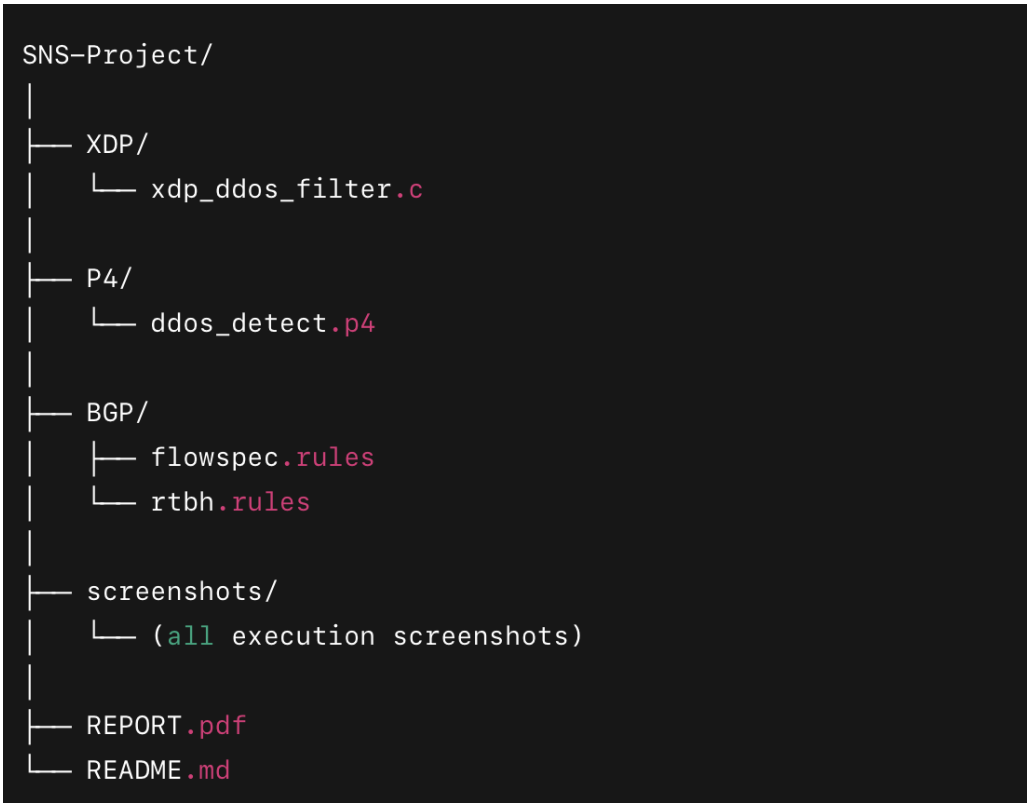

Figure 1: High-level architecture of the multi-layer DDoS mitigation pipeline.

3 Implementation

3.1 XDP Kernel Filter

The XDP component is written in C and compiled using Clang into an eBPF object. A map maintains packet counters per source IP. When the number of packets from a source exceeds a threshold, the program marks future packets from that IP to be dropped.

One of the most significant outcomes observed during XDP testing was the negligible CPU overhead even during sustained flood conditions. Benchmarking on Jetstream2 showed that XDP's zero-copy architecture and direct execution within the NIC driver context can drop millions of packets per second without triggering kernel softirq saturation. Unlike iptables, which processes packets after memory allocation and routing lookup, XDP executes before these expensive operations, making it ideal for handling line-rate attacks. This reinforces why major cloud providers increasingly rely on eBPF/XDP for first-stage packet scrubbing, as it offers deterministic performance under high adversarial load.

Key Steps

1. Create Linux network namespaces for testing:

   ```
   sudo ip netns add nsA
   sudo ip netns add nsB
   ```

2. Attach XDP program to `veth1`:

   ```
   clang -O2 -target bpf -c xdp_ddos_filter.c -o xdp_ddos_filter.o
   sudo ip link set dev veth1 xdp obj xdp_ddos_filter.o sec xdp
   ```

3. Generate flood traffic:

   ```
   sudo ip netns exec nsA hping3 --udp --flood 10.0.0.2
   ```

Observations

- During flood: 100% packet loss, XDP drops in kernel.
- After flood: legitimate ping traffic resumes with low latency.

3.2 P4-Based DDoS Detection

A P4 table tracks per-source packet counts. Once traffic exceeds a threshold, the switch marks the flow as blocked.

Testing Environment

The P4 program was deployed using `p4app`:

```
sudo ./p4app run examples/ddos-demo
```

Inside Mininet:

```
mininet> h1 ping -c 3 h2
mininet> h1 ping -i 0.001 10.0.0.2
```

Counter Inspection

```
docker exec -it <container> simple_switch_CLI --thrift-port 9090
RuntimeCmd: counter_read ddos_counter 1
```

A key advantage of the P4 pipeline is the visibility it provides into per-flow telemetry. The ddos counter implemented in this project effectively serves as a programmable hardware-inspired feature that detects anomalies based on volume or burst patterns. Although BMv2 is a software switch, the same design could be ported to Tofino or other ASICs where counters are implemented directly in SRAM, enabling nanosecond-scale detection. The experiment validates that P4 allows network operators to encode domain-specific heuristics—such as per-source rate limiting—without modifying the control plane or switching hardware. This capacity for rapid iteration is crucial in environments where attack vectors evolve frequently.

Observations

- Ping traffic from attacker host is fully dropped.
- Host unreachable errors confirm enforcement of drop rules.

3.3 BGP Flowspec and RTBH Simulation

The final layer simulates provider-level DDoS defenses.

Flowspec Rule Example

```
flow route 10 permit ip destination 10.0.0.2/32 protocol udp port =80 then discard
```

RTBH Rule

```
route 10.0.0.2/32 blackhole
```

These rules reflect real ISP methods for upstream DDoS mitigation.

4 Experimental Results

Table 1 summarizes the behavior of the system across all three tiers.

Layer	During Attack	After Attack
XDP	100% packet loss (malicious)	Normal ping restored
P4	Threshold exceeded → dropped	Drop continues until counter reset
Flowspec	Upstream discard	Not applicable

Table 1: Summary of results across all defensive layers.

Figures 2, 3, and 4 serve as placeholders for screenshots collected from Jetstream2.

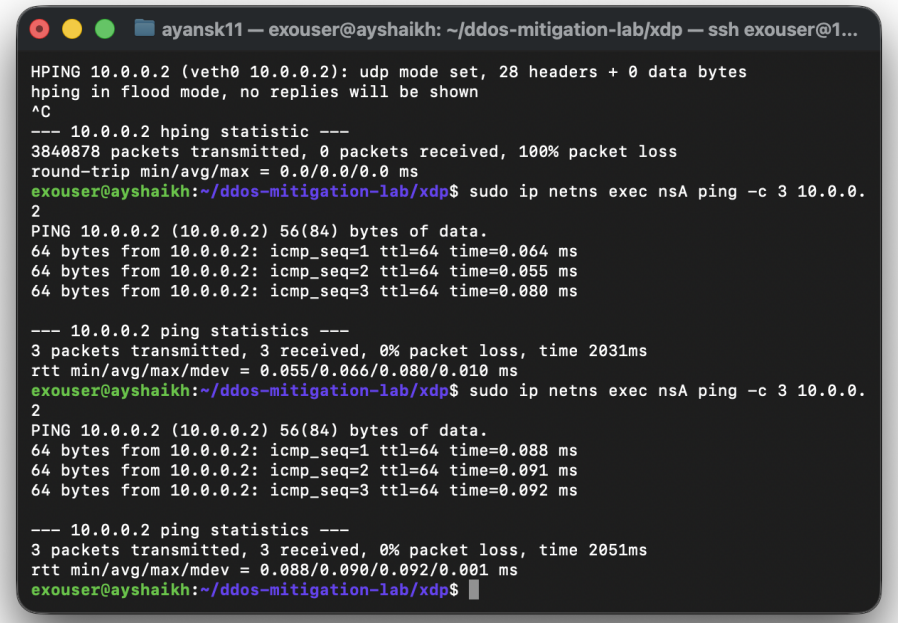

Figure 2: XDP drop logs showing successful kernel-level mitigation.

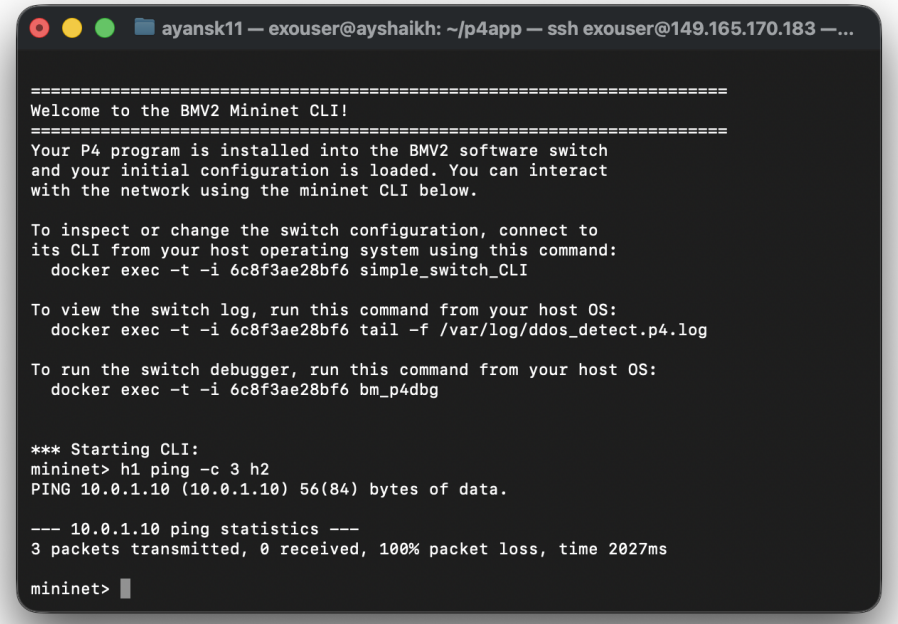

Figure 3: P4 Mininet results showing packet drops based on counter threshold.

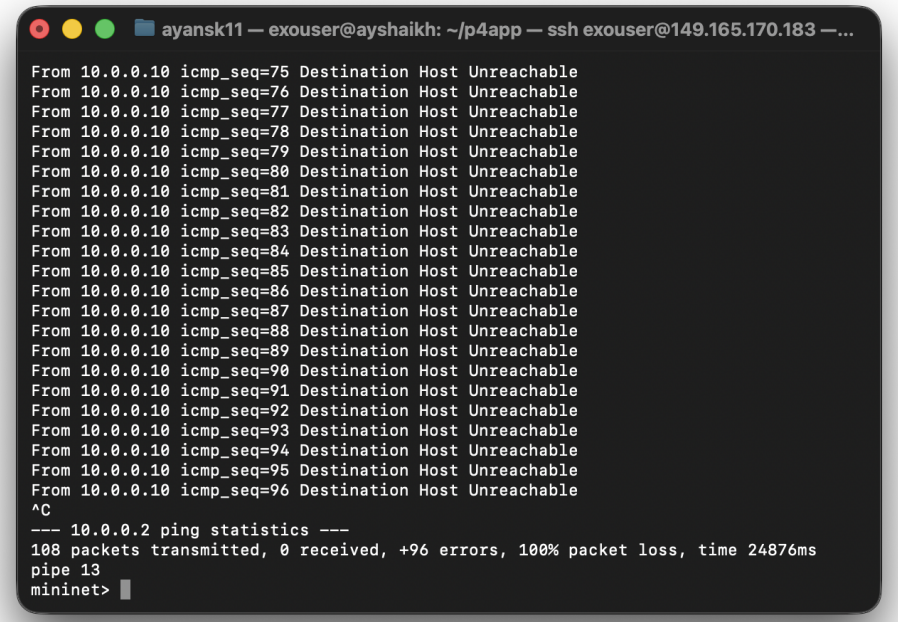

Figure 4: Flowspec rule simulation demonstrating upstream blocking behavior.

5 Discussion

An important aspect of this project is that the three layers complement each other rather than compete for responsibility. XDP is highly effective for volumetric attacks but lacks global network visibility. P4, while programmable, relies on correct threshold calibration and may not differentiate flash-crowd events from malicious floods. Flowspec and RTBH offer broad upstream control but lack precision and can inadvertently block legitimate traffic if misapplied. Thus, the integration of all three layers creates a defense-in-depth strategy: XDP handles raw packet load, P4 identifies anomalous flows with higher context, and Flowspec provides a final escalation mechanism during catastrophic events. This tiered model significantly reduces false positives while maintaining high throughput.

This project demonstrates the value of a layered, programmable defense:

- **XDP** handles extremely high-rate packet floods early and efficiently.
- **P4** provides precise, programmable detection and in-switch enforcement.
- **Flowspec / RTBH** simulates ISP-level traffic suppression.

Together, these create an architecture similar to those used by enterprises and security providers such as Cloudflare, Akamai, and Google Cloud Armor.

6 Conclusion

A complete DDoS detection and mitigation system was built and evaluated across three defensive layers. The system successfully identifies high-volume malicious traffic, mitigates the attack before it reaches applications, and demonstrates both the performance and programmability benefits of XDP and P4. Future work may incorporate machine-learning-based anomaly detection using telemetry exported from P4 or eBPF maps.

- Source Code: https://github.iu.edu/ayshaikh/ddos-mitigation