future internet

MDPI

Article

Location-Aware SDN-IDPS Framework for Real-Time DoS Mitigation in Vehicular Networks

Aung Aung, Kuljaree Tantayakul * and Adisak Intana

College of Computing, Prince of Songkla University, Phuket 83120, Thailand; 6730621003@psu.ac.th (A.A.); adisak.i@phuket.psu.ac.th (A.I.)

* Correspondence: kuljaree.t@phuket.psu.ac.th

Abstract

Integrating Software-Defined Networking (SDN) to enhance mobility management in Vehicular Ad Hoc Networks (VANETs) comes with an additional critical risk. Because centralized controllers are single points of failure, they create the risk that the network will be subject to denial-of-service (DoS) attacks during handovers. Most Intrusion Detection and Prevention systems (IDPSs) do not adequately address these risks because they are topology-blind and have excessive processing layers. This article presents a novel Location-Aware SDN-IDPS Framework that employs a hierarchical defense approach to protect vehicular networks against volumetric attacks. This two-plane system operates with the first tier, which uses dynamic host-location mappings to drop spoofed traffic at the switch level (data plane). In contrast, the second tier analyzes confirmed traffic through a Suricata-based engine to identify and respond to complex flood attack patterns. The experimental results from the Mininet-WiFi testbed show that the system provides a significant improvement over the unprotected state, with controller CPU utilization reduced by up to 18 times (from 9.0% to below 0.5%). In addition, the system provides a 2.3 s guaranteed recovery time, service continuity, successful microsecond-level mitigation time, and a packet delivery ratio (PDR) of 99.73% for legitimate safety messages. In control-plane stress testing, the proposed location-aware logic improved throughput stability by approximately 76.26% compared to the baseline. These findings confirm that offloading anti-spoofing logic to the network edge significantly enhances resilience without compromising performance in safety-critical vehicular environments.

Keywords: Software-Defined Networking (SDN); Intrusion Detection and Prevention System (IDPS); SDN mobility management; vehicular networks

Check for updates

Academic Editors: Faycal Bouhafs and Sachin Sharma

Received: 11 January 2026
Revised: 24 January 2026
Accepted: 3 February 2026
Published: 6 February 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

The integration of SDN with VANETs has become important for Intelligent Transportation Systems (ITSs) by separating the control plane from the data plane, which allows centralized mobility decisions to be handled and enables seamless handovers and optimized routing in highly dynamic vehicular environments [1,2]. Therefore, this combination leads to critical security vulnerabilities, and the controller becomes a single point of failure if the control plane is overwhelmed. As a result, the entire mobility management service can be severely affected or completely interrupted, making the controller an attractive target for denial-of-service (DoS) attacks [3,4].

There are increased security risks in vehicle networks during mobility handovers. When vehicles are crossing between various access points, the network controller has to

handle a colossal number of control messages (e.g., OFP_PACKET_IN) to update forwarding rules. Malicious actors often exploit this critical window to launch volumetric attacks, such as ICMP Ping Floods or IP-spoofed TCP SYN floods, to deliberately flood and crash the controller's processing capacity [5,6]. While traditional Intrusion Detection and Prevention Systems (IDPSs) like Suricata v8.0.0 are effective at signature-based detection, they are inherently "topology-blind" in mobile environments. Although these systems perform well at analyzing traffic content, they lack real-time awareness of physical network shifts, such as a vehicle moving across different switch ports. This blind spot often leads them to fail to recognize a valid changeover signal and a location-spoofing attack. Moreover, heavy deep packet inspection (DPI) is often accompanied by excessive latency, and it is not always easy to ensure a smooth connection without affecting network performance [7,8].

The existing literature tends to separate these areas, focusing either on static SDN security or on the efficiency of independent IDPS systems [9,10]. This fragmentation leaves us with a significant gap, which we can address with an active, mobility-aware defense mechanism that defends vehicles as they move. To address these limitations, we implement an integrated security system that combines SDN's topology awareness with the analytical capabilities of an IDPS.

To address this challenge, we propose a Location-Aware SDN-IDPS. Unlike conventional approaches that rely solely on heavy traffic inspection, our framework implements a hierarchical defense strategy. The primary contribution of this work is the development of a mechanism that uses dynamic host-location mappings to cross-reference the physical ingress port of each incoming packet. This approach shifts the security model's paradigm from a reactive search for malicious content to a proactive topological verification system. By bypassing a dual-layer defense, we can prioritize blocking spoofed traffic at the switch level, reserving resource-intensive Suricata analysis only for verified connections, and greatly reduce the overall system load.

We used Mininet-WiFi to simulate a realistic vehicular mobility environment to test our framework. The experimental analysis focuses on the effectiveness of the hierarchical defense against the volumetric attacks of the large scale. The practical effect of our design could be observed by using such indicators as the packet delivery ratio (PDR) for legitimate users, SDN controller resource consumption (CPU usage), Total Mitigation Time (TMT), recovery times (RTs) and the Scalability Control Plane Stress Test. Finally, the findings demonstrate that it is necessary to shift the burden of preventing spoofed traffic to the data plane while maintaining the critical role of topology awareness in securing safety-critical vehicular applications.

The remainder of this article is organized as follows. Section 2 reviews related work on SDN security and existing IDPS solutions in vehicular networks. Section 3 details the architecture of the proposed Location-Aware SDN-IDPS Framework and its operational logic. Section 4 describes the experimental setup, topology, and attack scenarios. Section 5 presents the performance analysis and discussion of the experimental results. Finally, Section 6 concludes the article and outlines directions for future research.

2. Related Work

The security of Software-Defined Networking (SDN) has been extensively studied, with particular focus on integrating Intrusion Detection and Prevention Systems (IDPSs) to mitigate denial-of-service (DoS) attacks. This section categorizes the existing literature into three primary domains: Static SDN Defense Mechanisms, Advanced Detection using AI/ML, and Security Analysis in Vehicular/Mobile Networks. A comparative summary is provided in Table 1.

Table 1. Comparison of related works.

Refs	Prevention Techniques	Detection Techniques	Attack Types	Environment	Key Performance Metric	Mobility Support
Fares et al. [7]	Traffic redirection (Honeynet)	Snort-based detection	DoS from internal devices	Ryu + Mininet	Detection time 3.47 s, response delay 30.44 s	No (Static)
Agapiou et al. [8]	IDS/IPS-based prevention	Snort IDS/IPS	TCP SYN flood	ONOS + Snort	CPU/mem load, resilience	No (Static)
Jamal et al. [10]	None	Manual log inspection	Fast and Slow DDoS	GNS3 + ODL	15% downtime, resource exhaustion	No (Static)
Cao, R. [11]	Threshold and REST API	Static, Dynamic detection (sFlow-RT)	ICMP flood	Ryu + sFlow-RT	Flexibility vs. Performance	No (Static)
Tseng et al. [12]	OpenFlow-based blocking	Suricata IDS + ML (DT)	IPv6 DoS/Telnet	SDN/NFV IPv6	Improved mitigation accuracy	No (Static)
Mazher et al. [13]	OpenFlow-based modification	R-IDPS (ML-SVM)	ICMP, TCP SYN	SDN IoT testbed	97–99% accuracy, low overhead	No (Static)
Tymoshchuk et al. [14]	None	Deep Learning (CNN)	Flood (TCP, UDP, HTTP, ICMP)	Ryu + Mininet	99% accuracy, low overhead	No (Static)
Clinton et al. [15]	OpenFlow-based prevention	Neural Network (24-106-5)	Flood (SYN, ACK, HTTP, UDP)	KVM Hypervisor	99.35% accuracy, low overhead	No (Virtual)
Poonia et al. [16]	Flow table updates	Entropy-based + Random Forest	ICMP flood, SYN flood	Ryu + Mininet	CPU utilization	No (Static)
Bin Zamrai et al. [17]	None	Performance monitoring	SYN flood	SD-IoV	Small packets cause disruption	Yes (Analysis)
Al-Dhuraibi et al. [18]	None	Traffic behavior	DDoS in VANETs	SDN VANETs	Packet delivery ratio degraded	Yes (Analysis)
Jaiswal, A. [19]	None	RYU SimpleMonitor	DoS Traffic (Ping)	Ryu + Mininet	Tx/Rx Packet Analysis	No (Static)
Proposed Approach	OpenFlow-based mitigation	Suricata + Location Aware	Flood (ICMP/TCP) Spoofed	Mininet-WiFi	CPU < 0.5%, PDR 99.73%, 2.3 s Recovery, Stability 76.26%	Yes (Active)

2.1. Traditional IDPS and Static SDN Defense

Initial studies were aimed at integrating simple IDPS tools with SDN controllers in order to secure fixed network configurations. For example, Scarfone et al. [20] were the first to define the architectural considerations for signature-based and anomaly-based detection, setting the field's requirements in their NIST Special Publication. Based on those ideas, Fares et al. [7] created a complex framework that used Snort with a Honeynet. In spite of the fact that they could effectively reroute malicious traffic, there was a mitigation delay of 30.44 s, which significantly degrades system performance. The heavy reliance on controller polling caused this lag, which eventually rendered the approach impossible and made it unsuitable for real-time vehicular applications.

Similarly, authors such as Agapiou et al. [8] and Cao et al. [11] investigated the SDN ability to resilient to the stress of TCP SYN and ICMP floods. Although Agapiou et al. [8] and their team discovered that the Snort IDS was capable of identifying such attacks, and they also observed that the large volumes of traffic brought down the CPU and bandwidth of the controller. Cao et al. [11] attempted to address this by comparing

dynamic sFlow-RT mechanisms to static thresholds, finding that while dynamic methods offered flexibility, they did not provide a specific context to mobile handovers. In addition, although Tseng et al. [12] integrated Suricata with OpenFlow-based blocking to address IPv6 Telnet attacks, their study remained within the static NFV framework, and mobility issues were not extensively covered.

2.2. Advanced Detection: Machine Learning and Blockchain

In a bid to overcome the limitations of signature-based detection, recent studies have shifted to integrating Machine Learning (ML) and Blockchain. Singh et al. [21] conducted a comprehensive survey of these techniques, demonstrating their ability to identify complex DDoS patterns in SDN-IoT networks. As an illustration, Mazher et al. [13] proposed a system, R-IDPS, a Support Vector Machine (SVM)-based device designed to operate in the IoT that achieved an exceptional accuracy of 97–99% and consumes fewer resources. Nevertheless, these high scores were accompanied by a notable decrease in detection performance once the system was strained by heavy attack traffic.

Recent studies have also taken Deep Learning as a significant area of interest. Neural networks were used by Tymoshchuk et al. [14] to classify both HTTP and UDP floods with an accuracy of 95.05%, whereas Clinton et al. [15] used a different approach in which the traffic data was transformed into an image by using Convolutional Neural Networks (CNNs), achieving an even greater classification accuracy of 99.35%. In another approach, Poonia et al. [16] explored a Blockchain-based method to verify legitimate IP addresses throughout a network topology. Although these sophisticated techniques offer impressive precision, they often demand heavy computational resources or long training times. This causes a compression in an edge-based vehicular controller, which requires a microsecond response time in order to operate safely.

2.3. Security in Vehicular and Mobile Networks (VANETs/IoV)

The combination of SDN and vehicular networks produces certain vulnerabilities in security, mostly due to the fact that the vehicles are always on the move. As Bin Zamrai et al. [17] indicated, network layers of VANETs are more susceptible, and they proposed dynamic updates to isolate the threats. In a subsequent study, the same authors investigated the impact of SYN floods on Software-Defined Internet of Vehicles (SD-IoV), and it was found that even a 1-byte packet is sufficient to cause a severe drop in the performance of the entire network.

Al-Dhuraibi et al. [18] reported VANET performance under DDoS stress, and it clearly shows that the ratio of delivered packets decreases significantly during an attack. A similar study by Jamal et al. [10] found that there is a difference between fast and slow DDoS attacks: slow-burning attacks, which consume controllers' resources silently and are hardly noticeable, can be even more dangerous. Jaiswal et al. [19] also studied this by analyzing traffic irregularities using the Ryu controller. In spite of these observations, there is still a significant gap between the studies [10,17–19] in terms of offering an active and automated defense that can survive as a vehicle traverses the network and which can detect the threat or analyze the destruction it causes, rather than offering an analysis of the destruction itself or identifying the threat itself.

2.4. The Research Gap

As summarized in Table 1, existing solutions exhibit a distinct trade-off. Static SDN defenses [7,8,10–13,19] offer mitigation but lack mobility awareness. ML-based approaches [12–16] offer high accuracy but incur computational costs that may delay handover processing. Meanwhile, vehicular security studies [17,18] often stop at impact analysis. Existing quantitative solutions suffer from severe performance penalties in mobile

environments. For instance, Fares et al. reported a mitigation delay of 30.44 s due to heavy controller polling [7], while Agapiou et al. observed that standard Snort deployments led to 100% CPU saturation under flood conditions [8].

Currently, there is no single framework that combines topological location verification with signature-based IDPS to mitigate the weaknesses of mobility handovers. Our article addresses the gap by proposing a Location-Aware SDN-IDPS that can offer real-time anti-spoofing and automatic remediation in dynamically varying VANET conditions.

3. Proposed Location-Aware SDN-IDPS Framework

In the following section, we propose the architecture and operational logic of the security framework. In most traditional IDPS configurations, these systems operate as standalone tools, whereas our architecture integrates intrusion detection into the SDN control loop. We designed it based on a hierarchical defense strategy, where a lightweight, location-based check filters out spoofed traffic just before it reaches the resource-intensive inspection engine.

3.1. System Architecture

The proposed architecture, as illustrated in Figure 1, comprises three interacting planes designed to enforce security policies dynamically.

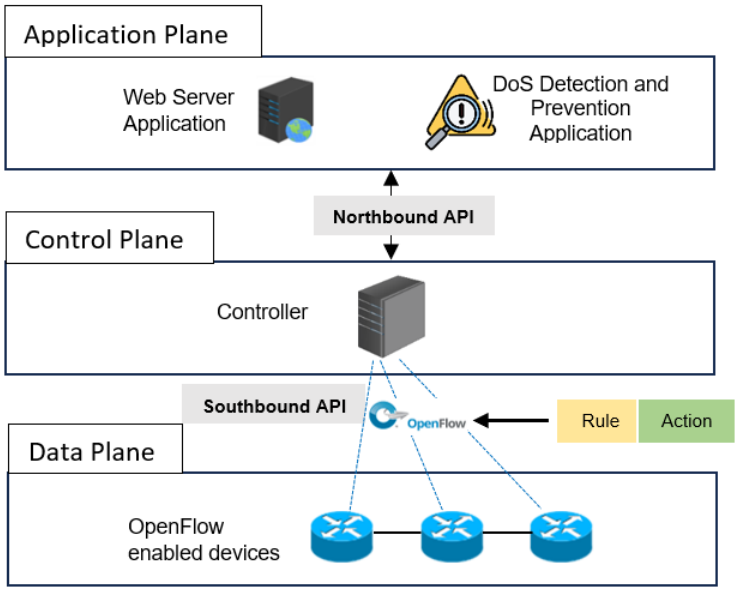


Figure 1. Location-Aware SDN-IDPS Architecture. The arrows represent the flow of security alerts via the Northbound API and OpenFlow rules via the Southbound API. Solid lines depict the logical interfaces connecting the Application Plane, Control Plane, and Data Plane.

- **Data Plane (Infrastructure Layer):** Our system is based on OpenFlow-enabled switches and wireless access points that control and redirect the traffic of vehicles in accordance with a certain set of flow tables. The data plane is used as the initial defense mechanism, and with high-priority "drop" rules, malicious activity is prevented at the network edge.
- **Control Plane (Orchestration Layer):** Hosted by the Ryu SDN controller, this layer maintains the global view of the network. It manages two critical modules:
    1. **Host-Location Manager:** Maintains a dynamic table of (Source IP, Switch Port) bindings to track vehicle mobility.
    2. **Security Enforcer:** Parses alerts from the Application Plane and installs OpenFlow drop rules via OFPT_FLOW_MOD messages to update switch flow tables.

- **Application Plane (Security Layer):** The IDPS engine of Suricata is hosted. It performs signature-based detection to check for abnormal traffic before forwarding. It scans only the traffic that has already passed the volumetric flood checks by the controller. If it detects any suspicious activity, it sends an early warning to the controller immediately via a southbound interface via syslog on UDP port 514.

3.2. Hierarchical Defense Logic

Since standard hierarchical methods overlay lightweight inspection with heavy inspection, our tier 1 defense is proactive topologically. The table of (Source IP, Switch Port) bindings, which is maintained in real-time allows the system to reject spoofed traffic at the switch level purely on physical metadata, without requiring any analysis of the packet contents. This changes the paradigm of "inspecting what is in the packet" (Reactive) to "the verification of the point at which the packet has entered" (Proactive). To address the latency and resource exhaustion challenges in vehicular networks, the framework operates in two distinct tiers:

3.2.1. Tier 1: Location-Aware Anti-Spoofing (Topology Verification)

This mechanism functions as the first line of defense, as illustrated in Figure 2. Since SDN controllers have a real-time view of the network topology, they can verify the physical source of traffic.

- **Host-Location Mapping:** When a vehicle attaches to the network at AP1, the controller records its identity and ingress point: Vehicle_IP → Ingress_Port_1.
- **Location Verification:** If an attacker attempts to spoof the vehicle's IP address from a different location (e.g., Ingress_Port_2), the controller detects a location mismatch.
- **Action:** The controller immediately installs a high-priority blocking rule on the switch for that specific flow. This process bypasses the IDPS entirely, preserving CPU resources.

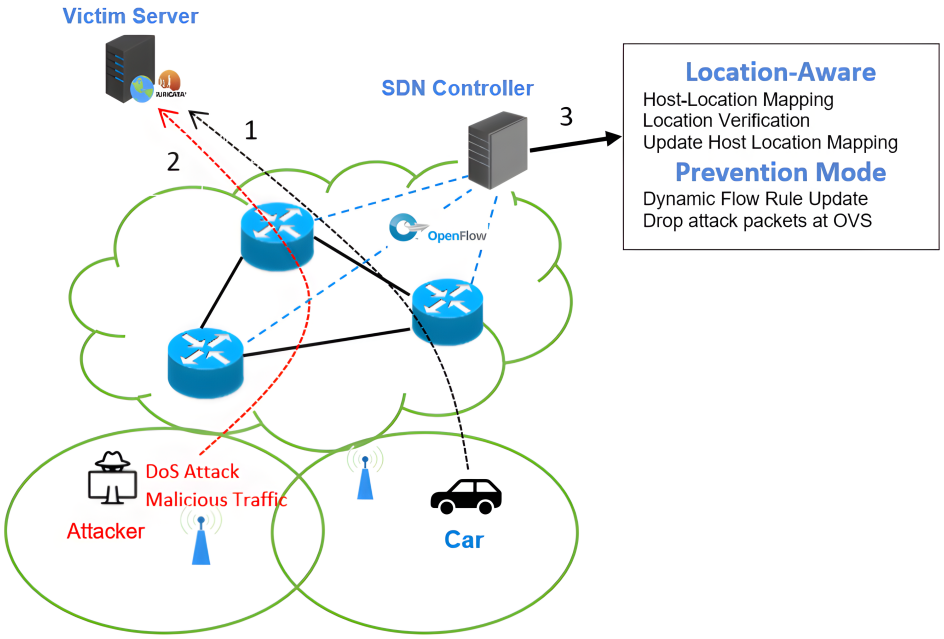


Figure 2. Workflow of Tier 1 Defense: Location-Aware Anti-Spoofing Verification Mechanism. Red dashed arrows indicate malicious traffic flows, while black dashed arrows represent legitimate vehicular traffic. The solid arrow depicts the alert signaling, controller, and victim server actions, and the blue dashed lines denote OpenFlow control channels. The numbers indicate the sequence of operations.

3.2.2. Tier 2: Signature-Based Anomaly Detection (Suricata IDPS)

When a legitimate vehicle performs a handover and attaches to an AP2, the SDN controller updates the host-location mapping accordingly. Attack traffic that passes the

topological check (i.e., traffic originating from the correct port or legitimate network path) is forwarded to the Suricata engine, as shown in Figure 3.

- **Detection:** Suricata analyzes traffic patterns against defined rule sets (e.g., ICMP rate limits and TCP SYN flags). In order to achieve good performance when performing deep packet inspection, it uses optimized pattern-matching algorithms, such as variants of the Aho–Corasick algorithm, in such a way that packet analysis remains efficient and does not become a bottleneck. According to Gagniuc et al. [22], this design can be used to reliably and consistently inspect validated traffic flows.
- **Mitigation:** If a signature match occurs (e.g., sid:1000001 for ICMP Flood), Suricata triggers an alert to the SDN controller to block the closest AP to the source.

Suricata uses the Aho–Corasick algorithm for high-speed internal pattern matching, which allows for multi-pattern payload inspection. When matched, a log is sent to rsyslog, which streams a UDP payload to the SDN controller. The controller builds the dynamic OpenFlow rule based on the parsed payload.

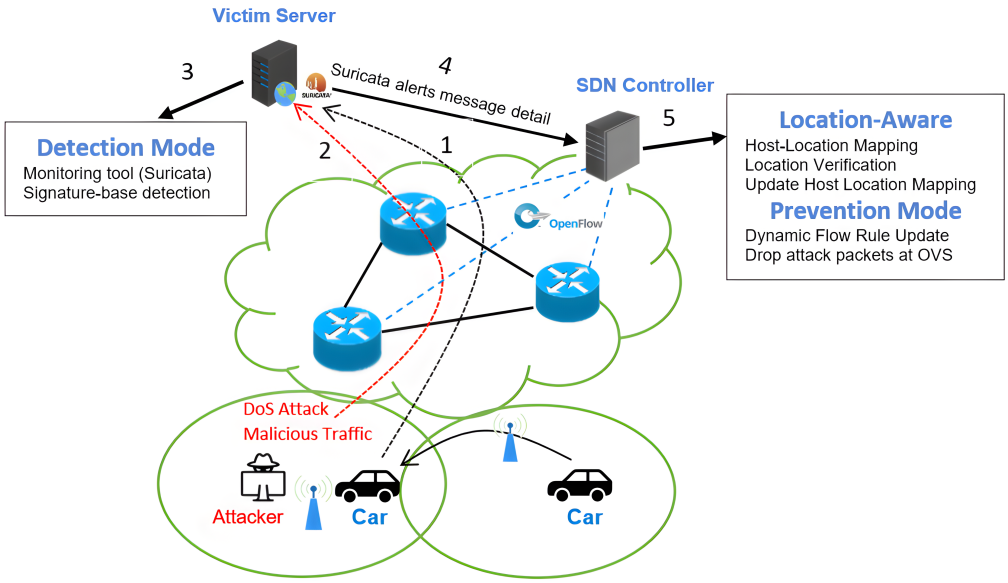


Figure 3. Workflowof Tier 2 Defense: Signature-based Anomaly Detection and Mitigation via Suricata. Red dashed arrows indicate malicious traffic flows, while black dashed arrows represent legitimate vehicular traffic. The solid arrow depicts the alert signaling, controller, and victim server actions, and the blue dashed lines denote OpenFlow control channels. The numbers indicate the sequence of operations.

3.3. Operational Signaling and Decision Logic

The detailed interaction between network entities and the algorithmic decision process is critical for understanding the system's real-time capabilities.

3.3.1. Signaling Workflow

The temporal sequence of message exchanges is depicted in the Signaling Diagram as illustrated in Figure 4.

1. **Initial Handshake:** Legitimate vehicles (e.g., Car1_VATI) attach to the network, triggering OFPT_PACKET_IN messages that allow the controller to establish valid host-location mappings.
2. **Spoofing Attempt (Tier 1):** In the event an attacker is attempting to spoof the identity of Car1 over a different access point (AP2), then the ingress switch forwards the unknown packet to the controller to examine it. The controller will block the packet at the switch level immediately after detecting the port mismatch by sending an OFPT_FLOW_MOD message.

3. **Handover Updates:** When a vehicle moves to a new AP, the switch transmits an OFPT_PACKET_IN message to the controller. The Host-Location Manager identifies the port change and revises the mapping table. It deletes the old entry to avoid replay attacks from the vehicle's former location.
4. **Tier Transition:** Tier 1 acts as a gatekeeper. Packets satisfying the valid (Source IP, Switch Port) tuple are sent as usual. An exact copy of this verified traffic is copied to the Application Plane by changing the packet to tier 2 and analyzing its signature.
5. **Volumetric Attack (Tier 2):** In cases where an attacker manages to bypass the location check (e.g., attacking from a valid port), the traffic will reach the victim server; however, the Suricata engine will then detect the flood pattern and initiate an alert to the controller via UDP port 514 to start mitigation.
6. **Final Mitigation:** The controller parses the alert and installs a permanent drop rule (Priority = 65535) on the edge switch, effectively isolating the attacker.

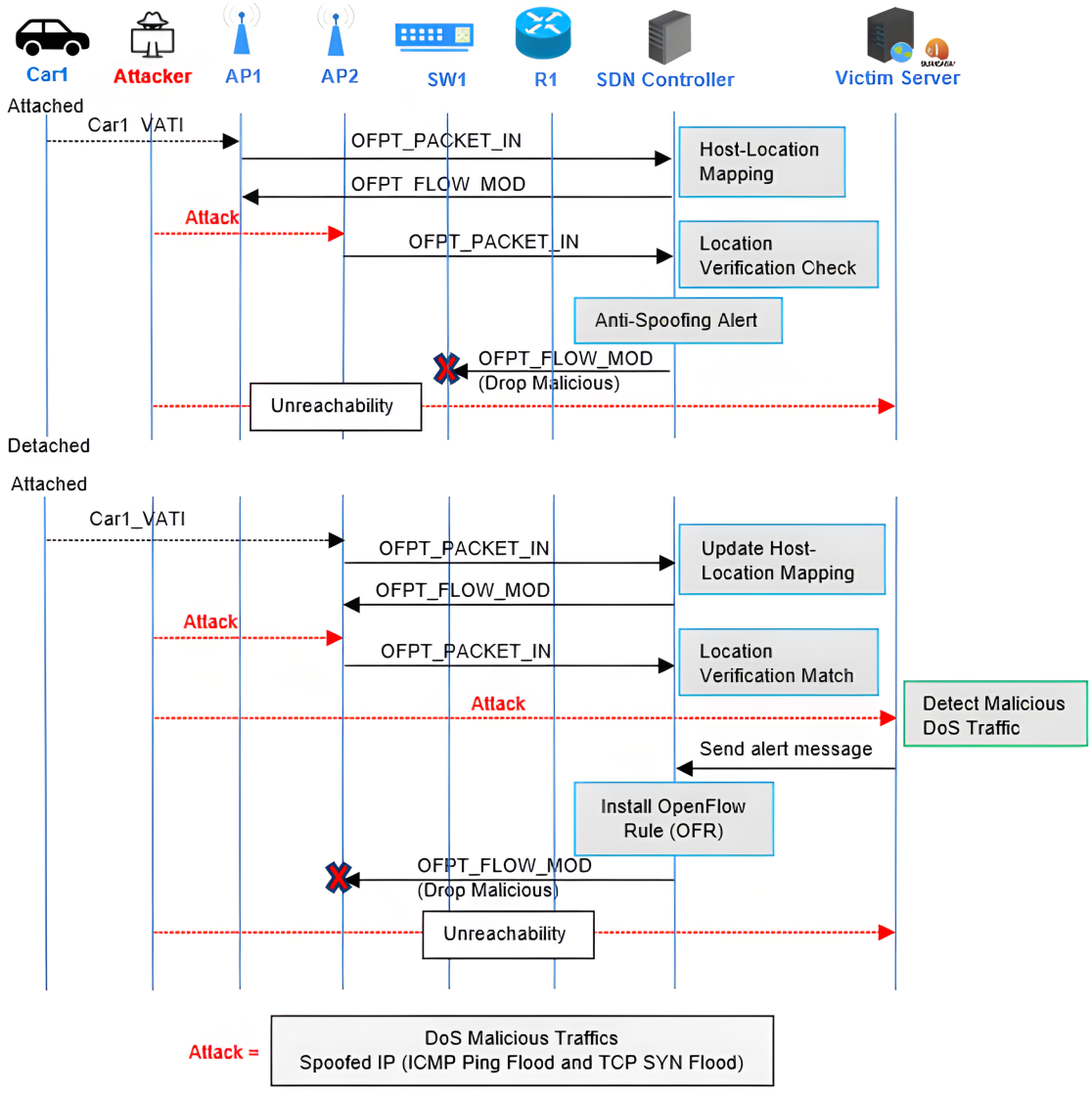


Figure 4. SDN-IDPS signaling diagram detailing the message exchange for tier 1 and tier 2 defenses.

3.3.2. Algorithmic Decision Process

The complete logic flow, from packet arrival to mitigation, is summarized in the Process Flowchart as illustrated in Figure 5. This flowchart illustrates the decision nodes where the system distinguishes between:

- **Bypass:** Attack traffic matching valid host-location mappings.
- **Location Mismatch:** Traffic triggering the tier 1 immediate drop.
- **Suspicious Traffic:** Verified traffic that subsequently triggers a tier 2 Suricata alert.

By integrating these decision paths, the framework ensures that no single point of failure exists; if one defense layer is bypassed, the subsequent layer remains active.

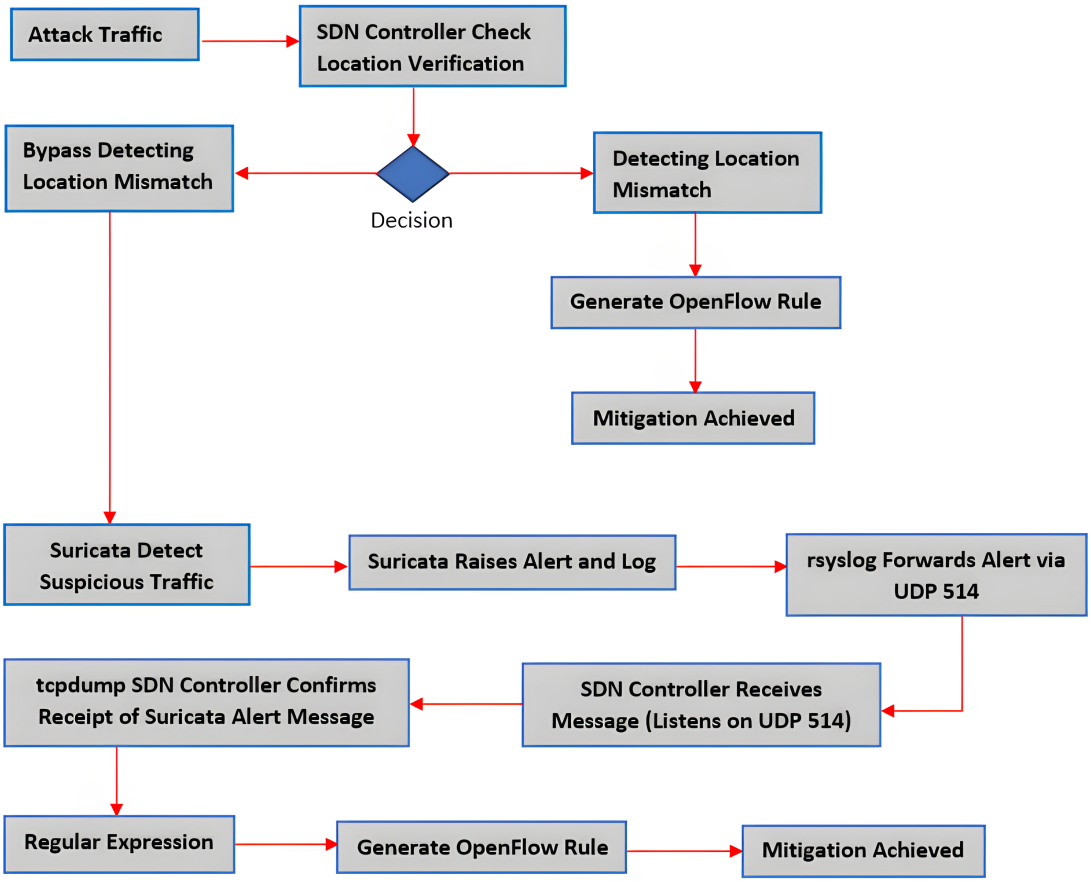


Figure 5. Detection and mitigation logic flowchart.

4. Experimental Setup and Methodology

To test our Location-Aware SDN-IDPS Framework, we constructed an elaborate emulation framework based on Mininet-WiFi. This section details the network topology, threat model, traffic generation methodologies, and the specific detection configurations used to evaluate the system's performance.

4.1. Simulation Environment

The experimental testbed was hosted on an Ubuntu 20.04 LTS virtual machine (Canonical Ltd., London, UK) allocated with 8 CPU cores and 8 GB of RAM to ensure sufficient processing power for high-speed packet forwarding. The host hardware operates on an Intel Core i7-14700 processor (Intel Corporation, Santa Clara, CA, USA). In the case of the network data plane, we used Mininet-WiFi v2.6.0 (Developed by Ramon Fontes, University of Campinas, Campinas, Brazil) as it has realistic wireless propagation and mobility capabilities. The Ryu Controller v4.30 (Nippon Telegraph and Telephone Corporation, Tokyo, Japan) controlled the entire control plane, and we deployed the Suricata IDPS engine v8.0.0 (Open Information Security Foundation, Indianapolis, IN, USA) on the victim server to enable signature-based detection. The complete hardware and software specifications are detailed in Table 2.

Table 2. Experimental environment specifications.

Component	Specification
Host Hardware	Intel Core i7-14700 (8 Cores), 16 GB RAM
Guest OS	Ubuntu 20.04.6 LTS
Network Emulator	Mininet-WiFi v2.6.0
SDN Controller	Ryu v4.30 (Python 3.8)
IDPS Engine	Suricata v8.0.0
Traffic Generation	Custom Python Script (Legitimate), hping3 (Attack)

4.2. Network Topology and Mobility Model

The experimental topology, illustrated in Figure 6, represents a typical Vehicle-to-Infrastructure (V2I) scenario. The network consists of

- **Infrastructure:** Two Wireless Access Points (AP1, AP2) connected to an OpenFlow switch (SW1) and an edge router (R1).
- **Vehicular Nodes:** Four connected cars (Car_1 to Car_4) equipped with wireless interfaces (10.0.0.3–10.0.0.6).
- **Attacker Node:** An internal malicious host (10.0.0.2) connected to the wireless network.
- **Victim Server:** A backend server (10.3.0.3) hosting a web application and the Suricata IDPS.

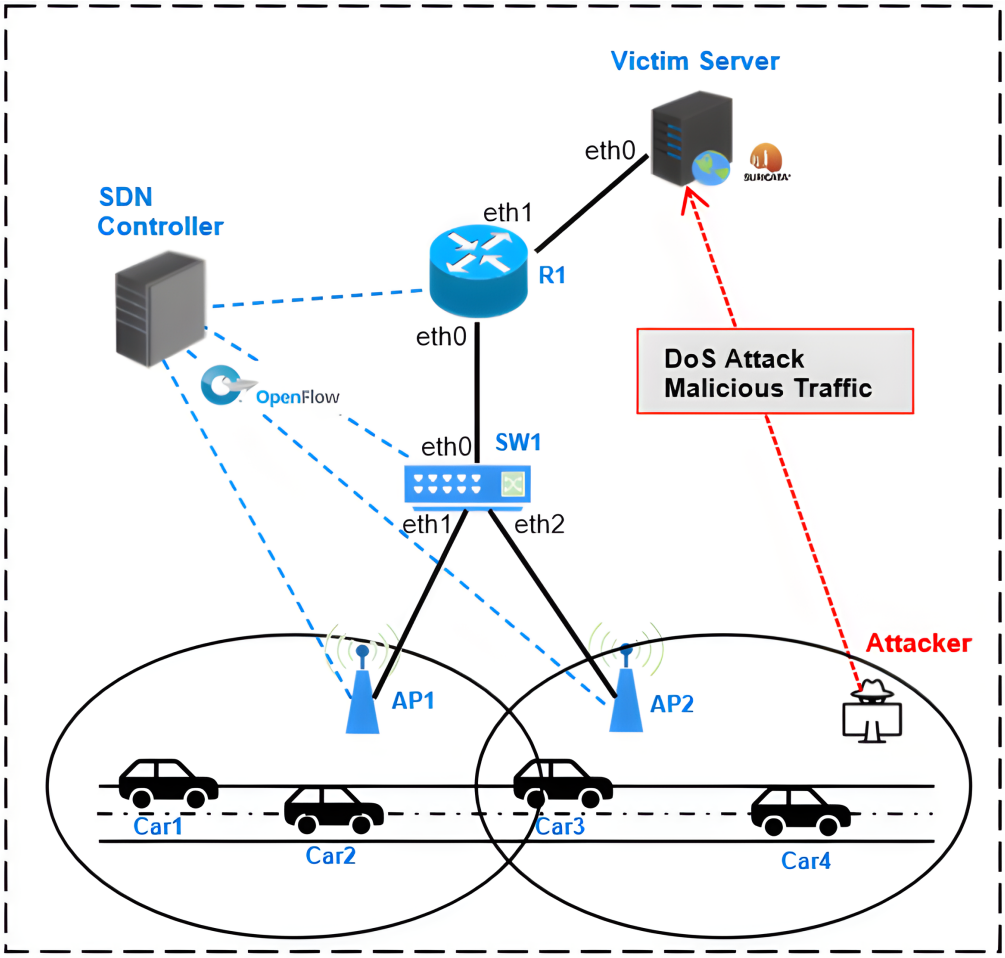


Figure 6. Experimental network topology in Mininet-WiFi. The black circles surrounding the Access Points (AP1 and AP2) represent the wireless communication range (coverage area) within which the vehicles connect to the network.

To simulate realistic urban traffic mobility, we modeled the cars using a linear mobility model with a constant speed of 39.38 km/h (10.94 m/s) to move along a 350 m trajectory. This speed was selected particularly depending on the real-time traffic statistics of Bangkok so that the experiment reflected authentic handover conditions between AP1 and AP2. Our linear path provided a deterministic, repeatable environment, enabling us to collect consistent, comparable performance measurements.

4.3. Threat Model

We assume an Internal Adversary located within the wireless network range. The attacker has the following capabilities:

1. **IP Spoofing:** The attacker can forge the source IP address of legitimate vehicles (10.0.0.3) to bypass simple Access Control Lists (ACLs).
2. **Volumetric Flooding:** The attacker can generate high-rate traffic (ICMP and TCP SYN) to exhaust network bandwidth and server resources.
3. **Mobility Awareness:** The attacker exploits the handover window—when host-location mappings are updating—to inject malicious traffic.
4. **Scalability to Distributed Attacks:** In spite of the fact that this study emulates one attacker node, the defense mechanism is per-flow based. When a distributed attack occurs, where a multiple nodes spoof legitimate vehicles at the same time, the controller will identify location mismatches at multiple ingress ports simultaneously and place independent drop rules on each attacking interface, and this will have the effect of isolating the distributed threat.

4.4. Traffic Generation Scenarios

To rigorously evaluate the system, we generated two distinct types of traffic running concurrently:

1. **Legitimate Background Traffic (VATI)**
   - **Purpose:** To measure the packet delivery ratio (PDR) of critical safety messages during an attack.
   - **Payload Explanation:** The Vehicle_ID (V_ID) uniquely identifies each registered vehicle, enabling the SDN controller to associate traffic with a legitimate entity rather than a spoofable IP address. The timestamp (Time) prevents replay attacks, the security key signature (Sig) authenticates the sender, and the hashed data (Data) ensures message integrity against tampering.
   - **Payload Example:** “V_ID”: “Car1”, “Time”: 1765604842, “Sig”: “e3f66450”, “Data”: “7eaa3b5ce. . .”.
2. **Malicious Attack Traffic**
   The attacker node uses hping3 to launch two specific DoS attacks that exceed the IDPS’s detection thresholds.
   - **ICMP Ping Flood (Spoofed):** hping3 –icmp –spoof 10.0.0.3 –flood
     - Target: Exhausts the controller’s packet-in processing by flooding spoofed echo requests.
   - **TCP SYN Flood (Spoofed):** hping3 -S -p 80 –spoof 10.0.0.3 –flood
     - Target: Saturates the victim server’s TCP connection table (half-open connections).

4.5. IDPS Detection Configuration

Suricata was configured with specific rule sets to detect volumetric anomalies that bypass the initial topology check. Since we assume a resource-constrained edge environment,

we used signature-based detection for its deterministic performance, lower latency than complex ML models, and suitability for highly dynamic mobility environments.

As defined in Table 3, the system is tuned to trigger an alert if ICMP traffic exceeds 100 packets per second (pps) or if TCP SYN connection attempts exceed 10 pps from a single source. These thresholds were calibrated to avoid false positives during normal high-load conditions while maintaining sensitivity to flooding attacks.

Table 3. Suricata detection rules.

Rule Type	Configuration Logic	Threshold
ICMP Flood	alert icmp any → $HOME_NET (msg:"ICMP Flood"; threshold: type limit, count 100, seconds 1;)	>100 pps
TCP SYN Flood	alert tcp any → $HOME_NET 80 (flags:S; threshold: type limit, count 100, seconds 10;)	>10 pps

These thresholds and rate-filter mechanisms are environment-dependent and intended to be tuned based on observed baseline traffic rather than fixed defaults. Based on this, we set the thresholds of 100 pps and 10 pps to calibration against items that were valid car traffic in our testbed, which usually was below 5 pps. However, the system investigates the counter and identifies the existence of ICMP flood and TCP SYN Flood IP Spoofed attack when it exceeds 100 packet counts in Ingle et al. [23].

5. Performance Analysis and Discussion

This section presents the experimental results obtained from the Mininet-WiFi testbed. The evaluation focuses on three critical dimensions: the functional efficacy of the hierarchical defense tiers, the system's resource efficiency (CPU utilization), and the impact on service continuity (throughput, PDR, TMT, RT, and Scalability Control Plane Stress Test) under active DoS conditions. The results below are average 10 repeated trials of each scenario. Standard deviations were used to determine statistical significance, and the confidence intervals (95%) were within ±2% of the reported mean of CPU usage, RT, and the stress test.

5.1. Functional Validation of Hierarchical Defense

We verified the operational correctness of the dual-tier defense mechanism by analyzing the system logs during active attack scenarios. The statistics reveal that our system can distinguish between location-based spoofing and volumetric flooding, enabling it to initiate the appropriate mitigation approach for a particular threat.

5.1.1. Tier 1 Validation: Location-Aware Anti-Spoofing

In this scenario, the attacker (10.0.0.2) attempts to spoof the IP address of a legitimate vehicle (Car_1: 10.0.0.3) from a different access point (AP_2) while Car_1 is legally attached to (AP_1).

- **Detection:** As shown in the controller logs in Figure 7, our tests demonstrated a good ability of the SDN controller to identify the mismatch of the locations. According to the log entry *[ANTI-SPOOFING] IP 10.0.0.3 spoofing detected at S1!*, one can easily notice that the controller indicated the discrepancy between an anticipated path to the host-location mapping (Port 1) and the actual path that the approaching traffic followed (Port 2).
- **Mitigation:** The controller immediately triggered a defense action by installing a high-priority drop rule on the ingress switch. The log *[BLOCK] Flow entry installed...*

priority = 30000 verifies that the spoofed traffic was blocked at the network edge before it could reach the detection engine.

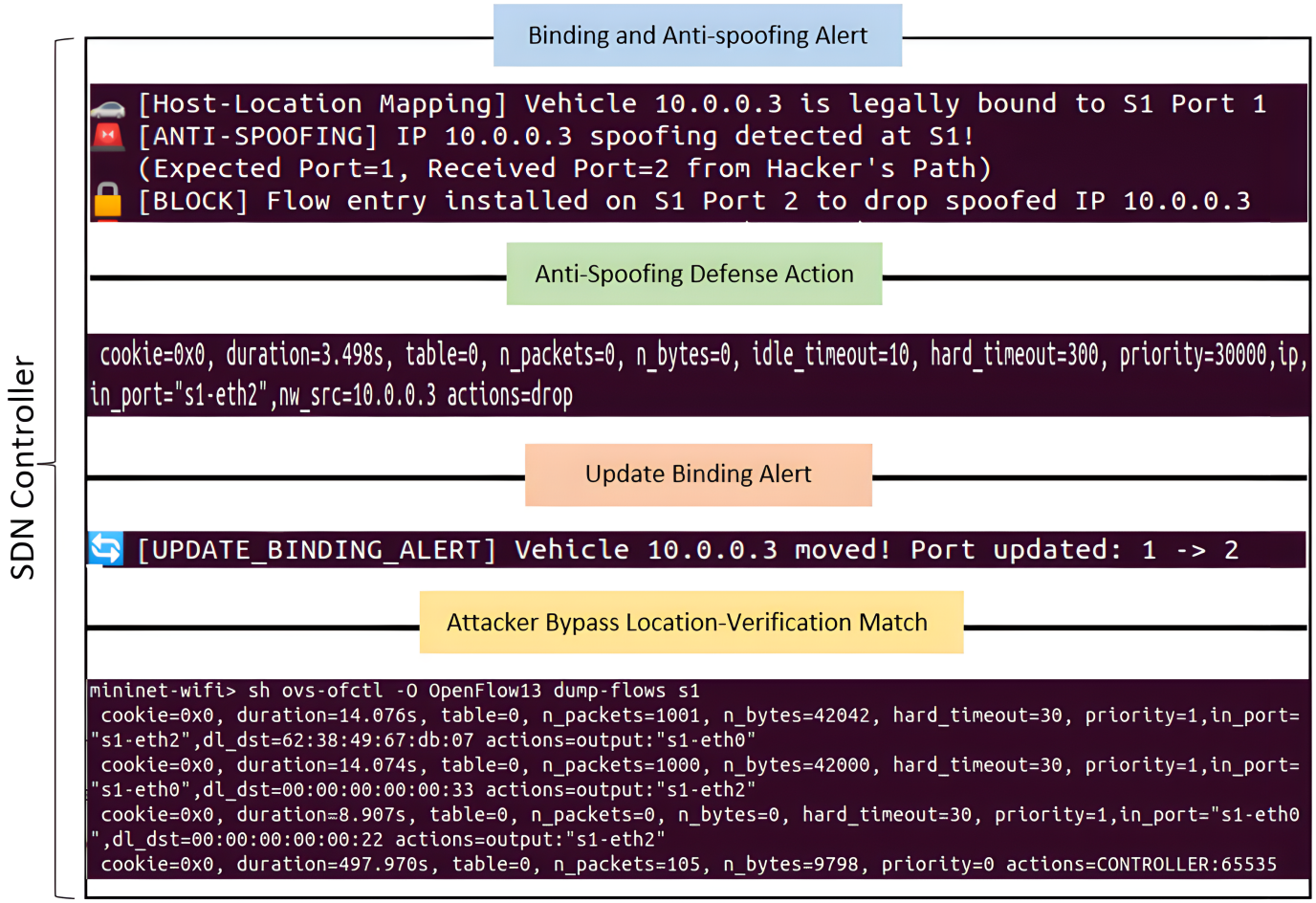


Figure 7. Controller logs demonstrating Tier 1: Location-Aware Detection and Automatic Blocking of Spoofed Traffic.

5.1.2. Tier 2 Validation: ICMP Ping Flood Mitigation

When the attacker launches a volumetric ICMP flood from a valid location (bypassing tier 1), the traffic reaches the Suricata engine. Figure 8 illustrates the complete detection and mitigation cycle for this attack vector.

- **Detection:** The Suricata engine was able to detect the anomaly in the traffic once it exceeded the 100 pps threshold. The trigger of the signature-based rule, *[1:1000001:2] ICMP Flood Detected (SPOOFED)*, confirms that the rule was activated and that the alert was timely sent to the controller.
- **Mitigation:** Upon receiving the alert, the SDN controller parsed a two-field match $\langle \text{Src}_{IP}, \text{Protocol} \rangle$ and installed a permanent blocking rule. As the data plane logs in Figure 8 show, a high-priority flow entry *priority = 65535, icmp, nw_src=10.0.0.3, actions=drop* was installed, and the counter values (n_packets) increased, which proves that the switch successfully removed malicious traffic, thus neutralizing the attack and achieving microsecond-level TMT at the AP2.

5.1.3. Tier 2 Validation: TCP SYN Flood Mitigation

We further validated the system against a high-rate TCP SYN flood targeting the web server port 80. The results are presented in Figure 9.

- **Detection:** The SYN flood pattern was rightly identified by the IDPS, and the log *[1:1000002:3] HIGH RATE TCP SYN Flood (SPOOFED)* demonstrates that the attack matched the specific Suricata rule set for TCP anomalies.
- **Mitigation:** The controller is responding correctly by verifying using OpenFlow flow dump: the log shows that a specified drop rule appears to be accurate: *priority = 65535, tcp, nw_src = 10.0.0.3, tp_dst = 80, actions = drop*. This shows that the system can do granular filtering with tri-field match $\langle \text{Src}_{IP}, \text{Protocol}, \text{Dst}_{port} \rangle$ and has been able to block malicious TCP streams without affecting legitimate traffic going through the network unhindered.

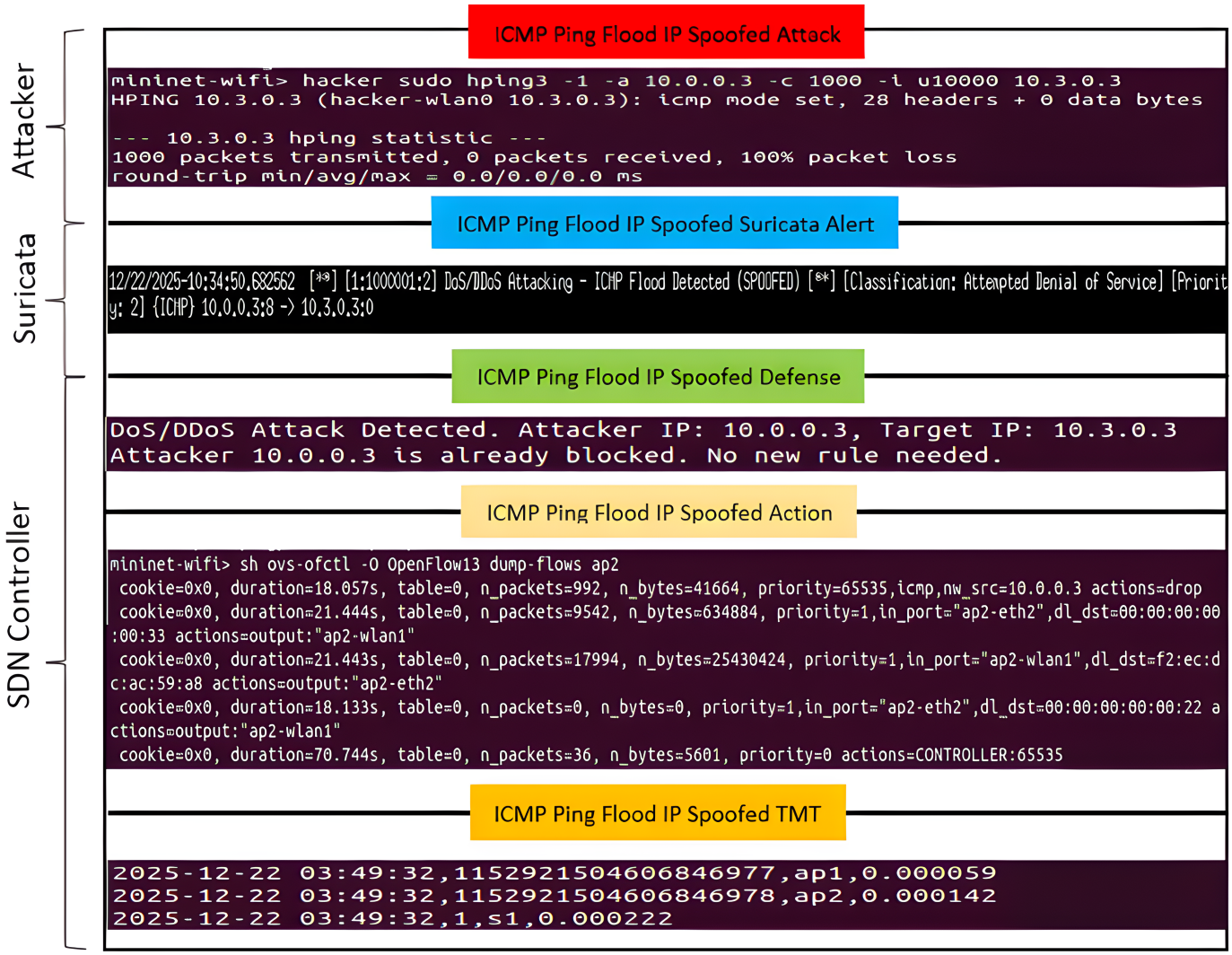


Figure 8. Tier 2 Mitigation logs: Suricata alerts triggering OpenFlow drop rules for ICMP ping flood.

5.2. Resource Efficiency Analysis (CPU Utilization)

A critical requirement for vehicular networks is minimizing the controller's computational load to prevent control-plane saturation. We compared the CPU utilization of the SDN controller under an ICMP Ping Flood attack (ranging from 1000 to 1,000,000 packets) with and without the proposed SDN-IDPS active.

As illustrated in Figure 10, the difference in resource consumption is substantial:

- **Without Defense:** CPU usage increases linearly with attack volume, peaking at 9.0%. This load stems from the controller continuously processing OFPT_PACKET_IN messages for every spoofed packet that does not match a flow rule. The baseline represents a standard SDN deployment where traffic is forwarded to the application

plane for inspection. This effectively mirrors a "Suricata-only" setup, where the controller is fully exposed to the flood traffic before the IDPS can trigger a response, resulting in the observed 9.0% CPU saturation.

- **With SDN-IDPS:** CPU usage remains negligible, staying consistently below 0.5%.

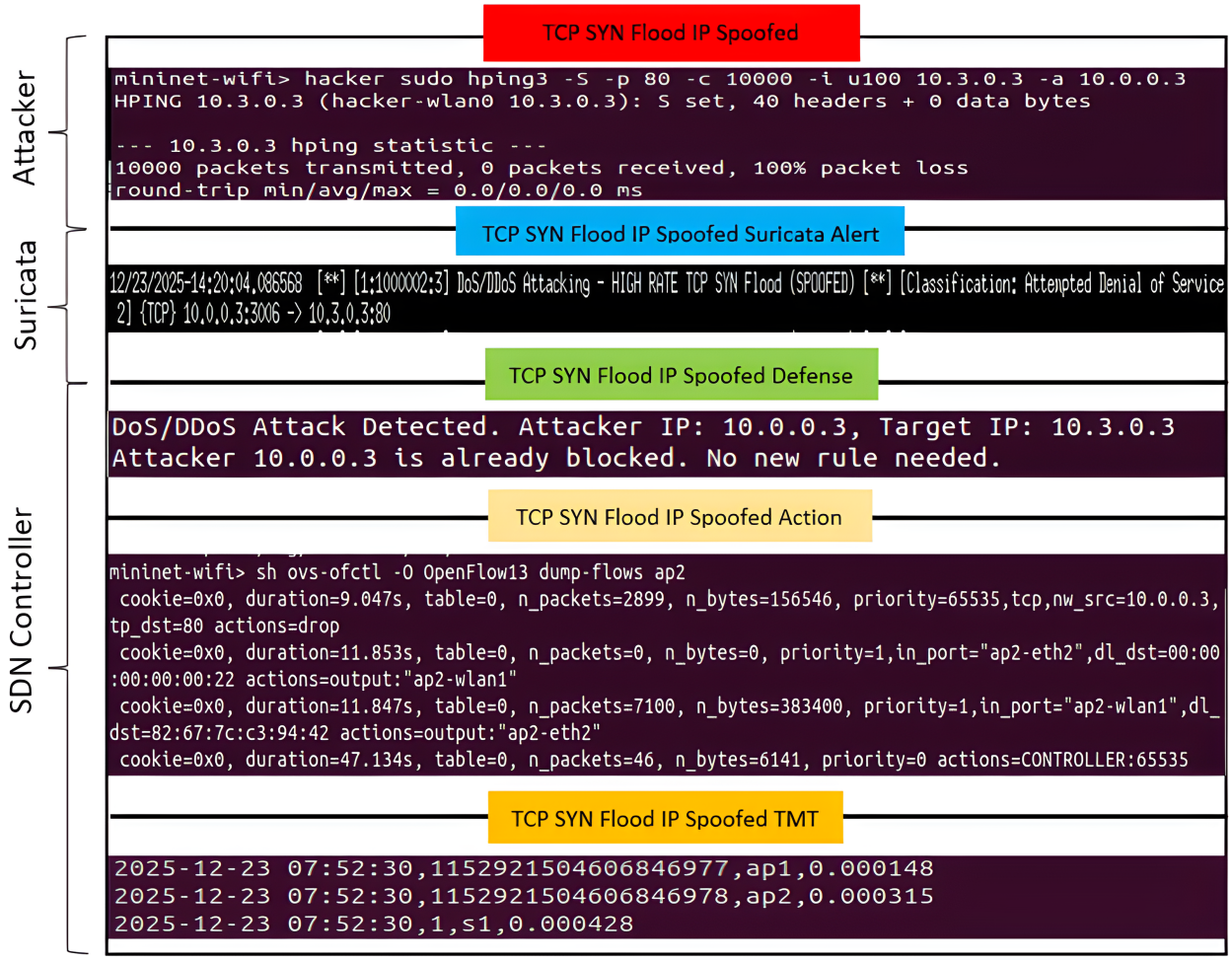


Figure 9. Tier 2 Mitigation logs: Detection and Dlocking of TCP SYN Flood.

Discussion: This is an 18-fold decrease in CPU overhead, which confirms the efficiency of the hierarchical strategy. The control plane is not exposed to the volumetric load, and so the resources can be reserved for the legitimate mobility management activities by moving the blocking logic onto the Data Plane (OpenFlow switches) once detected. Moreover, the negligible CPU consumption is maintained due to the fact that the tier 2 Suricata engine uses an effective Aho–Corasick automaton for pattern matching. This ensures that inspecting the remaining verified traffic incurs a constant computational cost per byte and does not cause processing bottlenecks even with complex attack signatures [22].

5.3. Service Continuity and Recovery Time Analysis

To measure the impact on user experience, we analyzed the TCP throughput of legitimate background traffic (VATI) before, during, and after a TCP SYN flood attack.

5.3.1. Baseline Performance (Normal Operation)

Under normal conditions, as illustrated in Figure 11, the network exhibits a stable TCP "sawtooth" pattern with an average throughput of 35 Mbps. The segment length remains constant at 1350 bytes, confirming reliable transmission of authentication tokens.

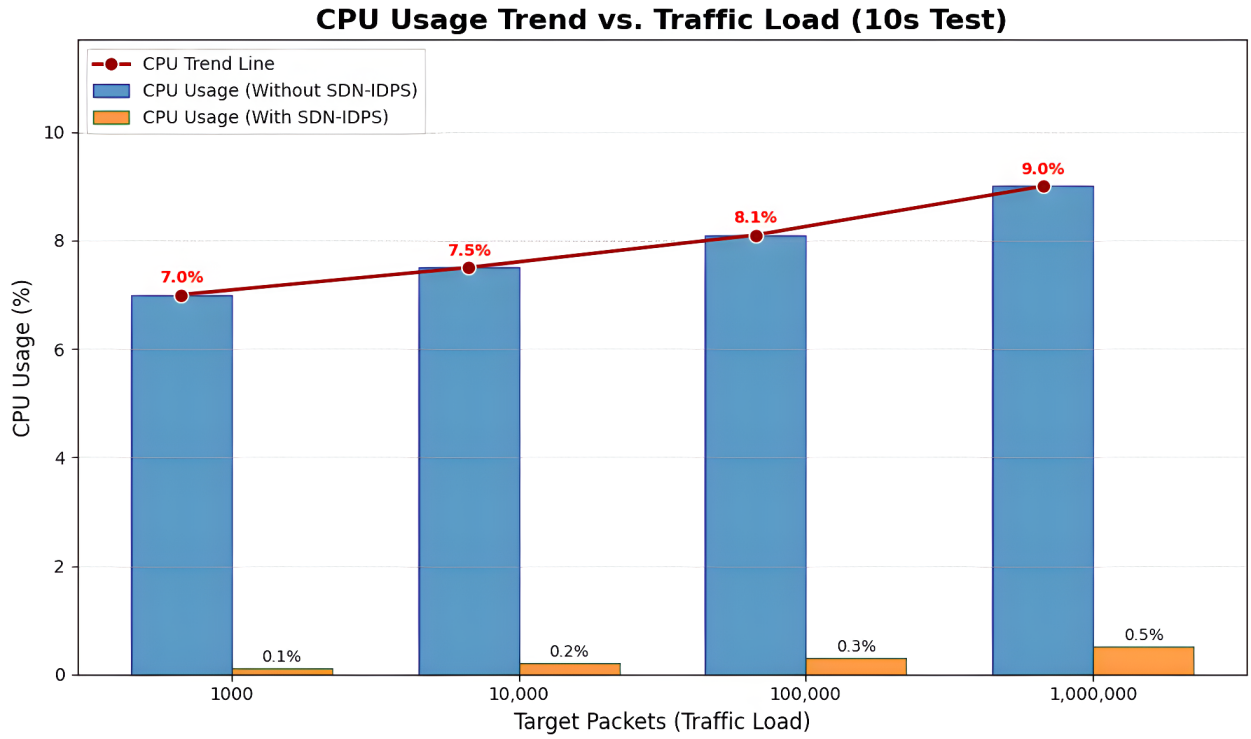


Figure 10. Comparison of controller CPU utilization vs. attack traffic load.

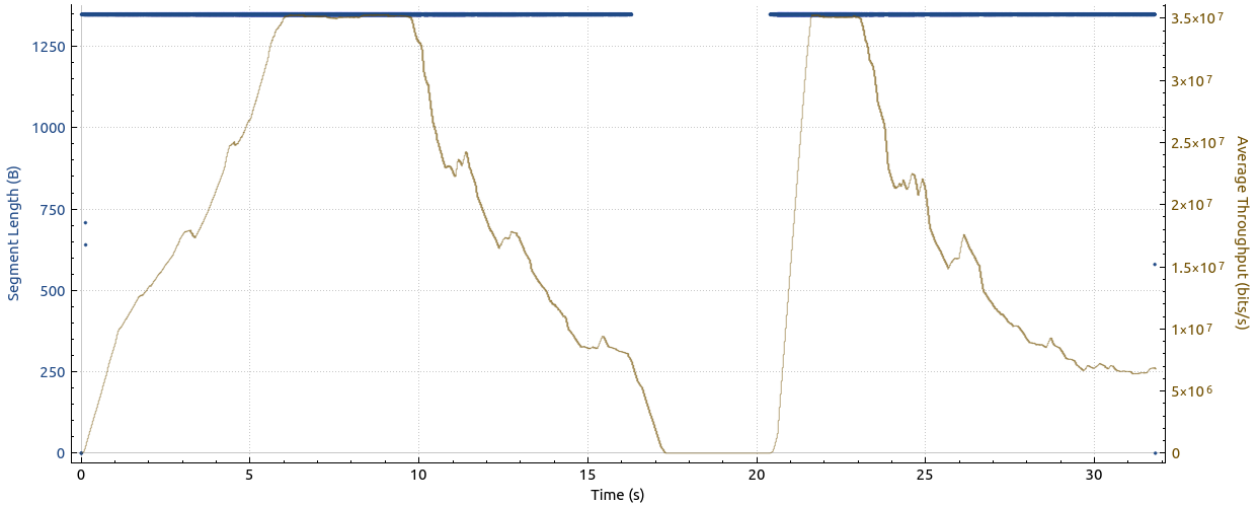


Figure 11. TCP throughput under normal conditions.

5.3.2. Impact of Attack (Unprotected vs. Protected)

- **Without Defense:** As shown in Figure 12, the onset of the attack causes an immediate service collapse. The server's connection queue is saturated by half-open SYN requests, causing legitimate throughput to flatline at 0 Mbps for the duration of the attack.
- **With SDN-IDPS:** Figure 13 demonstrates the system's resilience. When the attack launches at the 24.5 s mark, there is a momentary dip in throughput as the IDPS detects the anomaly. However, the system detects and mitigates the threat, allowing throughput to stabilize within 2.3 s.

5.3.3. Key Findings

The recovery time (RT) of 2.3 s represents the total latency for detection, signaling, rule installation, and TCP congestion control recovery. This rapid response ensures that safety-

critical V2I communications experience only transient disruptions rather than total denial of service.

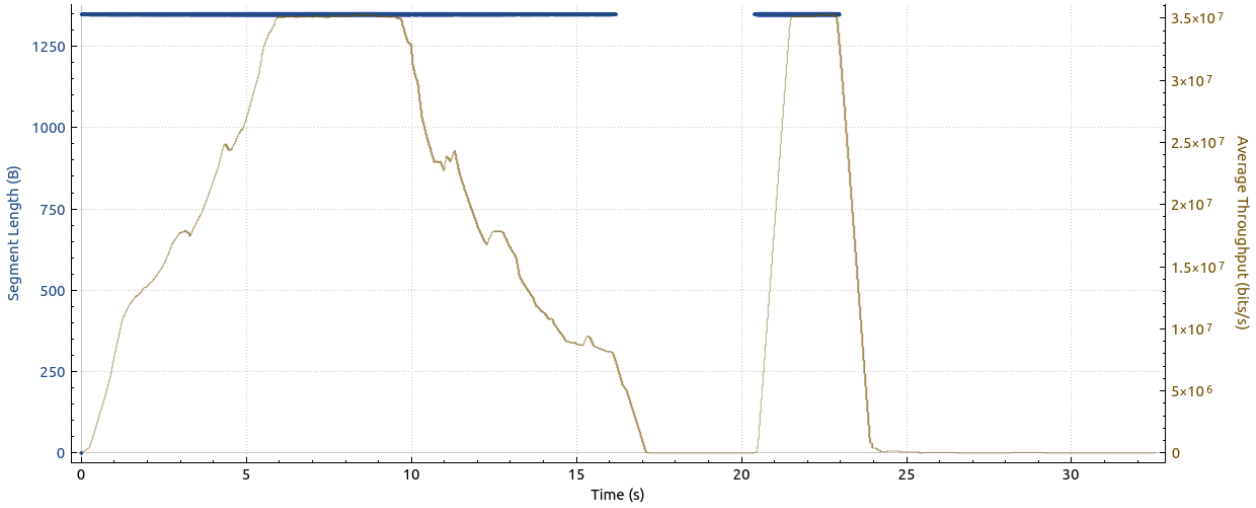


Figure 12. TCP throughput collapse during an attack without SDN-IDPS.

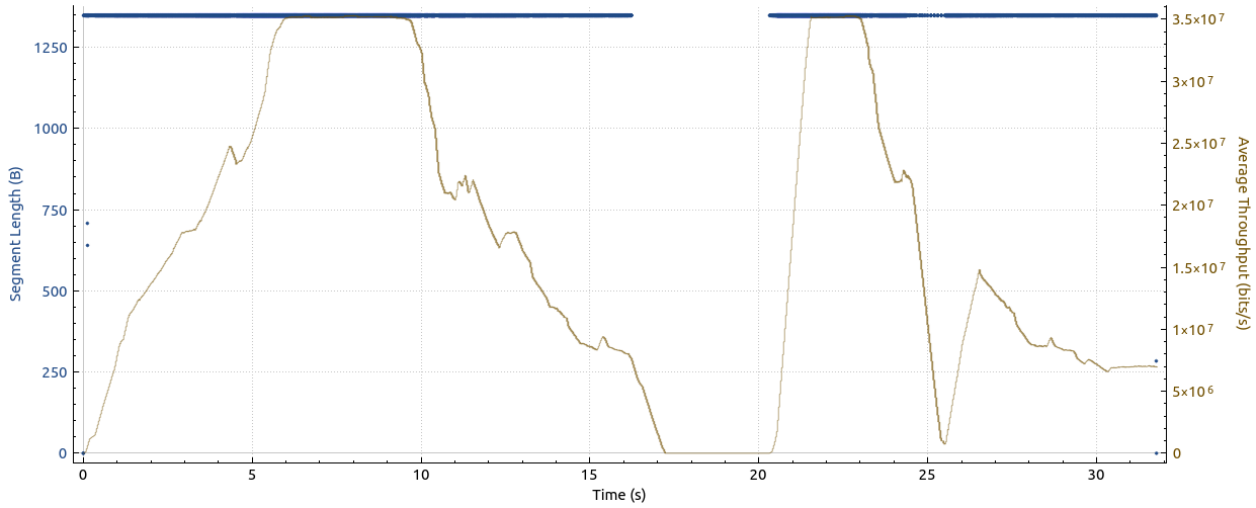


Figure 13. TCP throughput recovery with SDN-IDPS enabled.

5.4. Scalability Control Plane Stress Test

To verify scalability beyond the testbed's node count, we conducted a control plane stress test using the Cbench tool [24] at 10 loops. This evaluates the maximum flow-processing capacity of the Location-Aware module in high-density environments.

- **Methodology:** The Ryu controller was tested with 16 virtual switches and 100,000 unique MAC addresses. We utilized the Cbench tool to generate a stream of PACKET_IN messages, simulating concurrent flow setup requests. We compared baseline (no defense) to our proposed logic in throughput mode to measure concurrent flow setup capacity. In this context, the baseline represents an OpenFlow learning switch without any security or Location-Aware modules.
- **High-Capacity Processing:** The controller can handle approximately 1.37 million flows per minute (22,998 × 60). In a real-world VANET environment, even at a busy intersection, the flow request rate rarely exceeds a few thousand per minute. Thus, our system provides more than enough capacity to prevent bottlenecks.
- **Performance Stability:** The standard deviation for the proposed logic (61.32) is 76.26% lower than the baseline (258.28). This significant reduction in volatility indicates that

our $O(1)$ mapping logic provides highly stable, predictable response times, which are critical for safety-critical vehicular communications.

The comparative performance analysis achieved for each result is shown in Table 4. The proposed system maintains an average throughput of 22,918.02 flows/s even when handling 1.6 million unique MAC addresses across 16 switches, which shows that the security overhead is minimal. Although there is a minor decrease in throughput compared to the baseline due to the additional packet parsing (Ethernet and IP headers) and mapping logic, the performance is more than sufficient for large-scale VANETs. Because the verification logic utilizes an $O(1)$ hash-map structure, the time required to validate a vehicle's location does not increase as the number of vehicles grows, thereby addressing potential scalability bottlenecks in dense traffic conditions.

Table 4. Performance comparison scalability control plane stress test.

Metric (flows/s)	Baseline SDN Controller	Proposed Location-Aware SDN-IDPS
Avg. Throughput	27,734.73	22,918.02
Min. Throughput	27,355.61	22,800.78
Max. Throughput	28,177.20	22,998.88
Std. Deviation	258.28	61.32
Lookup/Verification Complexity	$O(N)$ (MAC learning table scan)	$O(1)$ (Hash-based host-location mapping)

5.5. Impact on Legitimate Traffic (PDR and FPR)

Finally, we quantified the system's reliability using the packet delivery ratio (PDR) and the False Positive Rate (FPR).

5.5.1. Packet Delivery Ratio (PDR)

The PDR is calculated as the percentage of unique legitimate packets successfully received by the server.

- **Normal Baseline:** 99.75% (reflecting minor wireless retransmissions).
- **Under Attack (Protected):** The system maintained a PDR of 99.73%. This statistically insignificant drop (0.02%) confirms that the mitigation rules are precise and do not collaterally damage legitimate flows.
- **Under Attack (Unprotected):** The PDR plummeted to 0% during the attack window as the server became unreachable.

5.5.2. False Positive Rate (FPR)

Throughout the experimental duration, the system recorded an FPR of 0%. This is attributed to the strict thresholding in IDPS and the specificity of OpenFlow drop rules, which target specific source IPs rather than broad network segments.

5.5.3. Summary

The experimental data conclusively prove that the Location-Aware SDN-IDPS Framework provides robust security with minimal performance overhead, maintaining 99.73% reliability for legitimate vehicles even while actively mitigating high-rate DoS attacks.

5.5.4. Discussion

Although a centralized Ryu controller was used in this study to prove the location-aware logic, we are aware of the concerns of scalability of large-scale VANETs. Our architecture will be open to new distributed architectures. In a multi-controller network such as CoCFL, the (Source IP, Switch Port) mapping table is synchronized across domain controllers using East–West bound protocols, such as a Distributed Hash Table (DHT) or a

Redis-backed distributed state store. This synchronization mechanism ensures global consistency of vehicle location information during inter-domain handovers. The $O(1)$ lookup efficiency demonstrated in the scalability stress tests (Section 5.4) is particularly critical in this context as it guarantees predictable synchronization overhead even as the number of domain controllers and vehicular nodes increases.

Furthermore, the proposed logic can be offloaded to 5G/Edge Hybrid Zones by implementing the tier 1 verification at Multi-Access Edge Computing (MEC) nodes using eBPF or P4-programmable data planes. By shifting the anti-spoofing verification to the network edge, malicious traffic is filtered at line rate, effectively preventing PACKET_IN flood messages from reaching the SDN control channel. This architectural flexibility ensures that the high-throughput performance (exceeding 22,998.02 flows/s) observed in the control-plane saturation tests is preserved, even in complex, multi-layered vehicular networks, by distributing the computational load across edge and core network components.

Even though in our experiments we were interested in a linear mobility model, the basic defense operation is based on $O(1)$ hash table lookups for host-location verification. Therefore, the computational cost per packet remains constant with more elaborate mobility patterns (like the Manhattan Grid) or denser deployments. At scale, however, the biggest problem will probably be the bandwidth of the control channel of messages in the OFPT_PACKET_IN format, which can be mitigated via distributed controller architectures as noted in Future Work. We also accept that the existing signaling is being based on UDP-based syslog messages, which can experience packet loss in the case of heavy congestion. Subsequent revisions will inject a comfortable TCP-based REST API to communicate with the IDPS to Controller.

6. Conclusions and Future Work

This paper highlights the unique challenges associated with the management of mobility in Software-Defined Vehicular Networks (SDN-VANETs). These challenges arise due to the central controller being exposed to potential denial-of-service (DoS) attacks during handover procedures. The absence of traditional security solutions in this area is primarily due to the lack of mechanisms to address the content of the traffic with the physical topology of the network leading to exhaustion of resources and increased latency.

We offered and recommended a Location-Aware SDN-IDPS Framework in order to overcome this gap. Through the Hierarchy Defense Strategy, the system will focus on a low-cost, topology-based verification (tier 1) to reject spoofed traffic in the network edge promptly and only allocate signature-based (tier 2) to the verified flows.

6.1. Key Findings from Our Experimental Evaluation

1. **Operational Efficiency:** The framework showed an 18-fold reduction in controller CPU utilization (from 9.0% to <0.5%) during active attacks. This demonstrates that offloading anti-spoofing logic to the data plane protects the control plane from getting saturated.
2. **Service Resilience:** The system has a fast recovery time (RT) of 2.3 s, proving that legitimate vehicular communication (VATI) was disrupted temporarily and will not be completely denied.
3. **Scalability:** This is evidenced by a 76.26% reduction in throughput standard deviation (from 258.28 to 61.32), confirming that the $O(1)$ complexity of our mapping mechanism provides highly predictable performance in high-density vehicular environments.
4. **Mitigation Performance:** The system achieves rapid mitigation due to the microsecond-level installation of flow rules, boosting the service's resilience against attacks.

https://doi.org/10.3390/fi18020087

5. **High Reliability:** The system showed a 0% False Positive Rate, even during high-volume ICMP and TCP SYN floods, to maintain a PDR of 99.73% for legitimate users, demonstrating the system's appropriateness for safety-critical ITS applications.

6.2. Future Work

While this study provides a robust foundation for edge-based vehicular security, we recognize limitations regarding attack complexity and simulation realism. Future research will focus on:

1. Advanced Threat Detection: The existing framework is mainly against volumetric attacks. The lightweight machine learning models will be used in the future to determine Application-Layer Floods (L7), Low-Rate DDoS attacks that slowly reduce the quality of the services, and distributed botnets that closely mimic legitimate traffic patterns, which are difficult to identify using threshold-based detection alone.
2. Architecture Scalability: In order to test the system on scale, we will scale up the framework to a Multi-Controller SDN environment to determine the performance of the system in a more heavily congested environment when inter-domain handovers occur.
3. Optimization of Signaling: Substituting our current use of event-driven alerts with our current syslog-based signaling so that we can further decrease our detection latency.
4. Realistic Mobility Simulation: We will use real-world map traces to test the entire framework in SUMO so that we will be able to evaluate its performance in complex urban road networks and real traffic patterns.

Author Contributions: Conceptualization, K.T. and A.A.; methodology, A.A. and K.T.; software, A.A.; validation, A.A., K.T., and A.I.; formal analysis, A.A. and K.T.; investigation, A.A.; resources, K.T. and A.I.; data curation, A.A.; writing—original draft preparation, A.A.; writing—review and editing, K.T. and A.I.; visualization, A.A.; supervision, K.T. and A.I.; project administration, K.T.; funding acquisition, K.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: No new data were created or analyzed in this study.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Tantayakul, K.; Dhaou, R.; Paillassa, B. Impact of SDN on mobility management. In Proceedings of the IEEE 30th International Conference on Advanced Information Networking and Applications (AINA), Crans-Montana, Switzerland, 23–25 March 2016; pp. 260–265. [CrossRef]
2. Gharsallah, A.; Zarai, F.; Neji, M. Exploring Software Defined Network (SDN) for Seamless Handovers in Future Vehicular Networks. *Int. J. Smart Secur. Technol.* **2022**, *9*, 304071. [CrossRef]
3. Olufemi, D.; Ejiade, A.O.; Ikwuogu, F.O.; Olufemi, P.E.; Bobie-Ansah, D. Securing software-defined networks (SDN) against emerging cyber threats in 5G and future networks—A comprehensive review. *Int. J. Eng. Res. Technol.* **2025**, *14*, 1–10.
4. Su, Y.; Xiong, D.; Qian, K.; Wang, Y. A comprehensive survey of distributed denial of service detection and mitigation technologies in software-defined network. *Electronics* **2024**, *13*, 807. [CrossRef]
5. Fatima, K.; Zahoor, K.; Bawany, N. SDN control plane security: Attacks and mitigation techniques. In Proceedings of the ACM International Conference (ICETE), Virtual Event, 6–9 July 2021; pp. 1–6. [CrossRef]
6. Farooq, M.S.; Riaz, S.; Alvi, A. Security and privacy issues in software-defined networking (SDN): A systematic literature review. *Electronics* **2023**, *12*, 3077. [CrossRef]
7. Fares, A.A.Y.R.; de Caldas Filho, F.L.; Giozza, W.F.; Canedo, E.D.; Lopes de Mendonça, F.L.; Amvame Nze, G.D. DoS attack prevention on IPS SDN networks. In Proceedings of the 2019 Workshop on Communication Networks and Power Systems (WCNPS), Brasilia, Brazil, 3–4 October 2019; pp. 1–7. [CrossRef]

8. Agapiou, S.; Peratikou, A.; Stavrou, S. Providing network resilience through an intelligent SDN network. In Proceedings of the 2024 Panhellenic Conference on Electronics & Telecommunications (PACET), Thessaloniki, Greece, 28–29 March 2024; pp. 1–6. [CrossRef]
9. Iqbal, M.; Iqbal, F.; Mohsin, F.; Rizwan, M.; Ahmad, F. Security Issues in Software Defined Networking (SDN): Risks, Challenges and Potential Solutions. *Int. J. Adv. Comput. Sci. Appl.* **2019**, *10*, 298–303. [CrossRef]
10. Jamal, Z.; Lakrami, F.; Sounni, H.; El Kamili, M.; Kobbane, A.; Baddi, Y. Examining the effects of DDoS attacks on SDN architectures: Experimental study. In Proceedings of the 2024 International Conference on Ubiquitous Networking (UNet), Marrakech, Morocco, 26–28 June 2024; pp. 1–5. [CrossRef]
11. Cao, R. Research on DDoS attack and defense system based on SDN controller. In Proceedings of the 2025 5th International Symposium on Computer Technology and Information Science (ISCTIS), Xi'an, China, 16–18 May 2025; pp. 594–598. [CrossRef]
12. Tseng, C.-W.; Wu, L.-F.; Hsu, S.-C.; Yu, S.-W. IPv6 DoS attacks detection using machine learning enhanced IDS in SDN/NFV environment. In Proceedings of the 2020 21st Asia-Pacific Network Operations and Management Symposium (APNOMS), Daegu, Republic of Korea, 23–25 September 2020; pp. 263–266. [CrossRef]
13. Mazhar, N.; Salleh, R.; Zeeshan, M.; Hameed, M.M.; Khan, N. R-IDPS: Real time SDN based IDPS system for IoT security. In Proceedings of the 2021 IEEE 18th International Conference on Smart Communities: Improving Quality of Life Using ICT, IoT and AI (HONET), Karachi, Pakistan, 11–13 October 2021; pp. 71–76. [CrossRef]
14. Tymoshchuk, O.; Shakhovska, A.; Lytvyn, V. Detection and classification of DDoS flooding attacks by machine learning method. *arXiv* **2024**, arXiv:2412.18990. [CrossRef]
15. Clinton, U.B.; Hoque, N.; Singh, K.R. Classification of DDoS attack traffic on SDN network environment using deep learning. *Cybersecurity* **2024**, *7*, 23. [CrossRef]
16. Poonia, L.; Tinker, S. DDoS mitigation by software-defined network (SDN) in the context of ICMP and SYN flood attacks. *Int. J. Intell. Syst. Appl. Eng.* **2023**, *11*, 1–9.
17. Bin Zamrai, M.; Yusof, K.; Azizan, A. Dissecting denial of service (DoS) SYN flood attack dynamics and impacts in vehicular communication systems. *Proc. ITM Web Conf.* **2024**, *63*, 01008. [CrossRef]
18. Al-Dhuraibi, W.A.; Elhadef, M. Securing vehicular ad-hoc networks: A DDoS case study. In Proceedings of the 2021 2nd International Conference on Computation, Automation and Knowledge Management (ICCAKM), Dubai, United Arab Emirates, 19–21 January 2021; pp. 112–117. [CrossRef]
19. Jaiswal, A. DOS attack network traffic monitoring in software defined networking using Mininet and RYU controller. *Res. Sq.* **2022**, 1–10. [CrossRef]
20. Scarfone, K.; Mell, P. *Guide to Intrusion Detection and Prevention Systems (IDPS)*; NIST Special Publication 800-94; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2007.
21. Singh, C.; Jain, A.K. A comprehensive survey on DDoS attacks detection & mitigation in SDN-IoT network. *e-Prime–Adv. Electr. Eng. Electron. Energy* **2024**, *8*, 100543.
22. Gagniuc, P.A.; Păvăloiu, I.-B.; Dascălu, M.-I. The Aho-Corasick Paradigm in Modern Antivirus Engines: A Cornerstone of Signature-Based Malware Detection. *Algorithms* **2025**, *18*, 742. [CrossRef]
23. Ingle, A.; Gour, A.; Kshirsagar, K. DDoS attack detection algorithms based on pattern classification and machine learning. *J. Univ. Shanghai Sci. Technol.* **2021**, *23*, 1007–6735. [CrossRef]
24. Gamess, E.; Tovar, D.; Cavadia, A. Design and Implementation of a Benchmarking Tool for OpenFlow Controllers. *Int. J. Inf. Technol. Comput. Sci.* **2018**, *10*, 1–13. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.