Efficient and Secure Architecture for Mitigating DDoS Attacks in Software-Defined Vehicular Networks

Meryem Chouikik[1], Mariyam Ouaissa[2]*, Mariya Ouaissa[3], Zakaria Boulouard[1], Mohamed Kissi[1]
[1]LIM, Hassan II University, Casablanca, Morocco
[2]LTI, Chouaib Doukkali University, El Jadida, Morocco
[3]LISI, Cadi Ayyad University, Marrakech, Morocco
E-mail : meryem.chouikik-etu@etu.univh2c.ma, ouaissa.mariyam@ucd.ac.ma, m.ouaissa@uca.ac.ma, zakaria.boulouard@fstm.ac.ma, mohamed.kissi@fstm.ac.ma
*Corresponding author

Keywords: ITS, SDN, SDVN, VANET, DDoS, IDS, snort, Mininet-Wifi, RYU controller

Received: December 27, 2024

Vehicular Ad Hoc Networks (VANETs) and Software-Defined Networking (SDN) have been combined to create SDN-enabled VANET architectures, which provide Intelligent Transportation Systems (ITS) with enhanced resource management, centralized control, and flexibility. However, the centralized control structure introduces new security challenges, notably Distributed Denial of Service (DDoS) attacks, which can significantly impact network stability and availability. In this study, we propose a DDoS detection framework for SDN-enabled VANETs, leveraging the Snort Intrusion Detection System (IDS) to effectively identify and mitigate DDoS threats. Our approach integrates Snort IDS into an SDN architecture based on the RYU controller, where Snort monitors network traffic for anomalies, and the SDN controller enforces dynamic mitigation strategies. The system is implemented using Mininet-WiFi, simulating a vehicular network environment with 10 vehicles nodes and a single RYU SDN controller. Performance evaluation under DDoS attack scenarios is conducted using the iPerf tool to measure key network metrics, including throughput, jitter, and packet loss rate. The results demonstrate that Snort IDS significantly improves network performance: jitter is reduced by up to 35%, packet loss rate decreases from over 40% to 15–25%, and throughput improves from 5–7 Mbps to a stable 10 Mbps. This study also explores the design and deployment of Snort within a Software-Defined Vehicular Network (SDVN) environment for effective DDoS detection and mitigation. By highlighting the importance of robust security mechanisms in SDN-enabled VANET architectures, this work contributes to the development of secure and reliable ITS infrastructures.

Povzetek: Članek predstavi varno SDVN arhitekturo z integracijo Snort IDS in RYU krmilnika za zaznavo DDoS napadov, kar izboljša prepustnost, zmanjša izgubo paketov in tresenje povezave.

1 Introduction

The rapid growth of Intelligent Transportation Systems (ITS) has sparked great interest in Vehicular Ad Hoc Networks (VANETs) as a vital technology for inter-vehicle communication and data exchange. VANETs, which are evolved from Mobile Ad Hoc Networks (MANETs), enable automobiles to act as mobile nodes, allowing for real-time information exchange that improves road safety, decreases traffic congestion, and improves the driving experience [1]. However, the dynamic and distributed nature of VANETs creates new issues for network management and security [2].

Software-Defined Networking (SDN) has emerged as a possible solution to some of the challenges associated with VANETs [3]. SDN offers centralized network management by divorcing the control plane from the data plane, as well as improving network flexibility and scalability. Integrating SDN with VANETs, resulting in an SDN-enabled VANET architecture, can improve traffic control, optimize resource allocation, and facilitate efficient routing. Despite these advantages, the centralized design of SDN creates weaknesses, most notably the possibility of Distributed Denial of Service (DDoS) attacks [4]. A DDoS attack on the SDN controller can jeopardize the overall stability and availability of the VANET network, posing a serious threat to ITS infrastructure [5].

To solve this issue, this paper presents a methodology for detecting DDoS attacks in SDN-enabled VANET infrastructures that employs the Snort Intrusion Detection System (IDS) [6]. Snort IDS is a powerful and popular program for monitoring network traffic and detecting potential security threats in real-time. Our approach uses Snort's capabilities to monitor network traffic patterns within the VANET, detecting odd behavior that could indicate a DDoS attack. The SDN controller enables rapid network response and coordination, allowing for timely threat mitigation [7].

In this paper, we present an in-depth analysis of DDoS attack detection in SDN-enabled VANETs, detailing the

integration of Snort IDS for enhanced network security. We conduct extensive simulations to evaluate the effectiveness of our proposed solution, demonstrating how Snort IDS can help maintain network stability and resilience against DDoS attacks. This research underscores the importance of robust security mechanisms in the future of ITS and contributes to the development of secure, efficient SDN-enabled VANET architectures. We measure throughput, jitter, and packet loss using widely accepted metrics before and after integrating Snort to assess its impact on network performance. Additionally, we explore the design and deployment of Snort within an SDVN environment to detect and mitigate DDoS attacks.

2 Background

In this section, we present the architecture of software-defined vehicular networks and their components, followed by a discussion of the security challenges associated with this architecture.

2.1 Software defined vehicular networks

Software Defined Vehicular Networks (SDVN) represent a transformative approach to managing and optimizing vehicular networks, which are essential components of modern ITS [8]. Traditional VANETs rely on decentralized architectures where each vehicle acts as a mobile node, communicating with other vehicles (V2V) or roadside infrastructure (V2I) [9]. While this decentralized structure allows for dynamic communication and data exchange on the move, it also presents challenges in terms of network management, scalability, and security [10]. This is where SDN comes into play, offering a centralized approach by separating the control plane from the data plane, thus allowing network intelligence and decision-making processes to be centralized in a single controller. In SDVN, the SDN controller can manage and coordinate vehicular network functions such as routing, traffic management, and security policies, making the network more flexible, adaptable, and easier to manage.

SDVN provides a centralized view of the entire vehicular network, allowing for real-time monitoring and control. This centralized control enables efficient traffic management, dynamic resource allocation, and optimized routing based on real-time conditions, which is particularly beneficial for densely populated urban areas prone to traffic congestion [11]. Moreover, SDVN architectures can enhance the deployment of safety-critical applications, such as collision avoidance systems, emergency vehicle prioritization, and efficient traffic light management, ultimately improving road safety and reducing response times during emergencies. SDN's programmability also allows for faster implementation of new applications and services tailored to the specific needs of VANETs without requiring changes to the underlying hardware infrastructure. As a result, SDVN can better support emerging applications in ITS, including autonomous driving, connected vehicles, and real-time traffic analytics [12].

However, while SDVN offers substantial benefits, it also introduces new challenges, particularly around security and resilience. The centralized SDN controller becomes a potential single point of failure and a prime target for cyber-attacks, such as DDoS attacks, which can disrupt the entire vehicular network [13]. To mitigate these risks, security mechanisms such as IDS, firewalls, and redundancy techniques need to be integrated into SDVN architectures. Additionally, the dynamic nature of vehicular networks where vehicles are constantly moving requires sophisticated and adaptive security and network management solutions that can handle rapid changes in network topology [14].

SDVN represent a promising evolution of vehicular networks, providing enhanced control, flexibility, and efficiency in network management. By enabling centralized control, SDVN facilitates efficient resource allocation and supports a wide range of ITS applications that improve road safety and traffic flow [15]. As the automotive industry moves toward more connected and autonomous vehicles, the importance of robust, scalable, and secure SDVN architectures will only continue to grow [16]. However, addressing the unique challenges of SDVN, particularly around security and scalability, will be critical to realizing the full potential of this innovative approach in the future of smart transportation [17]. Figure 1 shows the components of the SDVN architecture.

The SDN controller: is the central component of an SDN architecture, acting as the "brain" of the network. It enables centralized control and management of traffic, making decisions on data routing based on the network's needs. The controller communicates with network devices via protocols like OpenFlow, allowing dynamic and programmable network management. Examples of SDN controllers include RYU, OpenDaylight, and ONOS [18].

Northbound API: this interface connects the SDN controller to application-level services and allows it to translate high-level network policies into specific configurations for the underlying infrastructure. It plays a crucial role in enabling application developers to control and monitor network behavior without dealing with low-level network details.

Base Stations (BS): Positioned within the network infrastructure, base stations act as access points that connect vehicles and Road-Side Units (RSUs) to the SDN controller. These base stations ensure reliable communication between mobile nodes (vehicles) and the broader network, providing high-speed wireless links for data transmission.

OpenFlow RSUs: RSUs are fixed communication units deployed along roadsides or intersections. In this architecture, RSUs are OpenFlow-enabled, meaning they act as programmable switches that follow the instructions of the SDN controller. RSUs manage the interaction between vehicles and the network infrastructure, forwarding packets and optimizing data flow to ensure efficient communication [19].

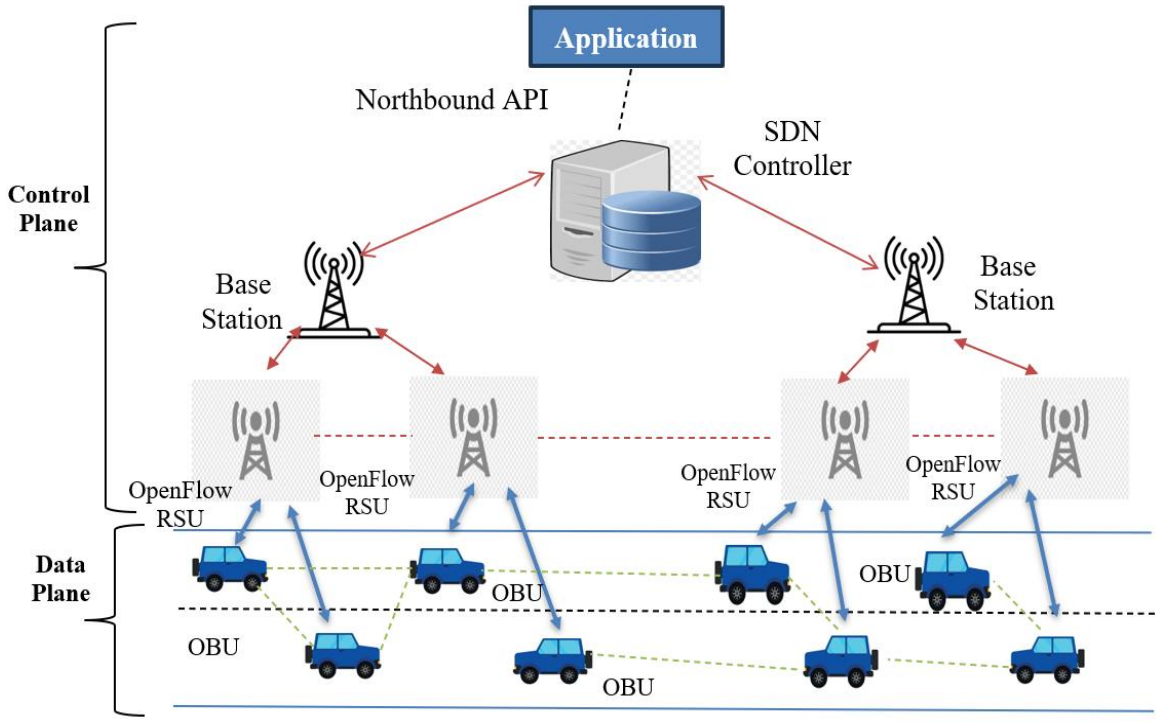

Figure 1: Software-defined vehicular network architecture

OBUs (On-Board Units): OBUs are installed in vehicles to enable wireless communication with RSUs and other vehicles. They support both Vehicle-to-Infrastructure (V2I) and Vehicle-to-Vehicle (V2V) communication, allowing vehicles to share critical information such as safety alerts, traffic updates, and navigation data.

Application Layer: Located above the SDN controller, this layer hosts various network applications and services, such as traffic management systems, route optimization tools, and safety applications. These applications leverage the centralized control provided by SDN to analyze network data and optimize vehicular communication dynamically.

2.2 Network security in software defined vehicular networks

Network security in Software-Defined Vehicular Networks is crucial, as SDVNs introduce a new paradigm that enhances the flexibility and manageability of communications through centralized control. The integration of the SDN concept into vehicular networks enables dynamic and programmable network management, allowing for better traffic flow optimization and timely data distribution [20]. However, this shift towards a centralized control system introduces unique security challenges that require stringent measures to ensure safe operation. Cyber-attacks, such as DDoS attacks, can exploit the SDN controller's centralized nature, potentially disrupting communications critical to vehicle safety and traffic management. Moreover, due to the mobility and highly dynamic nature of vehicular networks, ensuring real-time, secure communication between vehicles and infrastructure becomes complex, necessitating robust intrusion detection systems and intelligent threat mitigation strategies. Intrusion detection systems like Snort can play a significant role in identifying and countering threats by analyzing traffic patterns and detecting anomalies [21].

The controller's centralized structure in SDVNs makes it both a key component and a potential single point of failure, subject to different cyberattacks such as DDoS attacks, spoofing, and malware injection. The dynamic and highly mobile nature of vehicular networks complicates security even more, as vehicles join and depart the network on a regular basis, opening up new attack routes [22]. Furthermore, the sharing of sensitive data, such as location information and driving behavior, emphasizes the importance of secure communication protocols and encryption technologies for protecting user privacy [23].

2.3 Related work

Table 1 presents a comparative analysis of recent works on DDoS detection frameworks in SDN based VANETs, highlighting the detection methods, tools used, platforms, and evaluation metrics. This comparison underscores the novelty and practical effectiveness of our approach in addressing real-time SDVN security challenges through measurable network performance indicators.

Table 1: Comparative analysis of existing works on SDN based VANETs security

Study	Detection Method	IDS Tool	Platform	Metrics Employed
Türkoğlu et al. [24]	ML (KNN, SVM, DT) + MRMR + Bayesian Optimization	Custom ML pipeline	SUMO + Mininet-WiFi + POX + sFlow + InfluxDB (SD-VANET)	Accuracy (99.35%), Sensitivity, Specificity, F1-score
Elubeyd & Yiltas-Kaplan [25]	Hybrid DL (1D-CNN + GRU + DNN)	Custom DL model	RYU + Mininet + OVS (Colab), CICDDoS2019 & synthetic data	Accuracy (99.88%), Precision, Recall, F1-score, ROC-AUC
Setitra & Fan [26]	Optimized TabNet + Adam + GSCV	Opt-TabNet	Mininet + RYU with SDN-DDoS & InSDN datasets	Accuracy (99.42%), Precision, Recall, F1-score
Ma et al. [27]	RF with Heterogeneous Integrated Feature Selection + Edge/Distributed Computing	EDRFS (edge-deployed RF model)	CIC-DDoS2019 on SDN with distributed controller simulation	Accuracy (99.99%), Precision, Recall, F1-score, Prediction time (0.4s)

Compared to existing studies that primarily rely on machine learning or deep learning techniques for DDoS detection in SDN environments, the proposed approach distinguishes itself by integrating a signature-based Snort IDS with a RYU SDN controller for real-time anomaly detection and mitigation. While previous works focus on achieving high detection accuracy using offline datasets like CICDDoS2019 or NSL-KDD, they often lack real-world deployment and runtime traffic evaluation. In contrast, this study implements a practical, real-time defense system within a simulated vehicular network using Mininet-WiFi, where performance under DDoS attacks is assessed using network-level metrics such as throughput, jitter, and packet loss. This combination of signature-based detection, SDN programmability, and real-world traffic analysis offers a more deployable and responsive solution, especially suited for dynamic and latency-sensitive environments like vehicular networks.

3 Proposed scheme

This article focuses on cyber-attacks, with a special emphasis on the impact of DDoS assaults on SDVNs [28]. These attacks are primarily intended to impair network performance, cause disruptions, and undermine the centralized management and control of network resources. This is accomplished by dividing the network's design into two planes: control and data [29]. SDVNs provide benefits like as flexibility and scalability, but they also pose new security concerns. DDoS assaults are particularly significant risks to SDVNs, as a massive flood of malicious traffic can disable network resources and disrupt vital services. DDoS assaults have serious consequences for SDVN networks, resulting in performance deterioration, network unavailability, and significant financial ramifications.

In the fight against network threats, IDS serve as critical defenses. Snort, a renowned open-source IDS, exemplifies this approach to network security. It monitors network traffic by combining packet sniffing with signature-based detection to identify potential threats and intrusions [30]. Acting as an IDS, Snort continuously inspects traffic and compares it against predefined rules and signatures. These rules define behavioral patterns associated with known attacks or malicious activities. When a match is found, Snort generates alerts or logs detailed information, providing a comprehensive report on the suspected intrusion. Through deep packet inspection of network protocols and payloads, Snort effectively detects various types of attacks. These capabilities collectively fortify SDVN against threats targeting switches, controllers, and communication links

This section further explores the design and deployment of Snort within an SDVN environment to detect and mitigate DDoS attacks

3.1 System modeling

Snort was selected as the IDS for this study due to its lightweight architecture, open-source availability, and proven effectiveness in detecting a wide range of DDoS attack signatures [31]. Its design aligns well with the computational constraints typical of vehicular nodes and edge environments in SDVNs. Compared to alternatives such as Suricata, which offers multi-threaded packet processing for high-throughput applications, and Zeek, which provides deep contextual and behavioral traffic analysis, Snort offers simpler configuration and lower resource overhead. These features make it a practical choice for latency-sensitive environments where computational efficiency and ease of integration are critical. Additionally, Snort benefits from an active community, a rich rule set, and smooth interoperability with SDN architectures through log parsing and alert generation.

In our deployment, Snort was configured to detect common and disruptive DDoS attack types in vehicular

networks, specifically UDP and TCP floods, chosen for their prevalence and impact on bandwidth and latency in mobile scenarios. Upon detection, Snort generates alerts that the Ryu SDN controller interprets to modify flow rules. A priority-based flow table distinguishes confirmed threats, suspicious activity, and normal traffic: high-priority rules immediately block or reroute malicious flows, medium-priority rules flag questionable behavior for monitoring, and low-priority rules manage standard forwarding. This approach enables real-time mitigation without disrupting legitimate communication, with rule timeouts preventing table overflow.

Although Suricata's multi-threading and Zeek's behavioral analysis offer potential advantages in throughput and detection of unknown attacks, they often demand more resources and complex configuration. Integrating Snort with the SDN controller and prioritizing efficient flow rule management presents a practical and effective security solution for SDVNs, balancing reliability and latency requirements in performance-critical ITS environments [32].

The suggested system contains a DDoS attack detection and mitigation method by incorporating an IDS into the SDN architecture, which is intended for usage in both home and business networking settings [33]. The system functions on a feedback loop that includes three major architectural components: the network, the IDS, and the controller [34]. The network depicts all data flow and the point at which a DDoS attack could be initiated. The IDS detects DDoS attacks by analyzing all network traffic. When the IDS identifies an ongoing DDoS attack, it alerts the controller. Once the IDS alerts, the controller sends new flow rules to network devices along the data channel to restore normal network functioning as soon as possible (Figure 2).

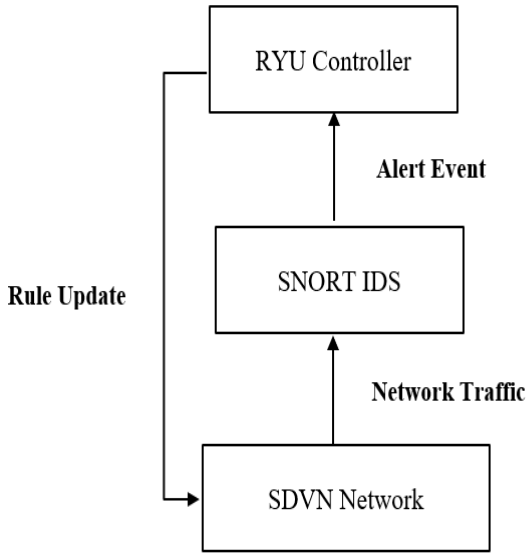

Figure 2: System architecture design

3.2 System deployment

The system comprises three important phases: detection, communication, and mitigation. The detection phase focuses on the system's ability to detect DDoS attacks. The communication phase happens when the IDS notifies the controller of the observed DDoS attack. The mitigation phase occurs when the controller applies specific flow rules to the local switch, preventing harmful traffic. These flow rules are stored permanently in the switch.

Figure 3 illustrates the conceptual model and workflow of the proposed system, which integrates an SDN controller with an Intrusion Detection System (IDS) to form an SDN-Based IDS Monitor for detecting and mitigating DDoS attacks. When a new packet enters the system, it is classified as part of a flow. The OpenFlow switch first checks its flow table for a matching rule. If a rule exists, the packet is forwarded to its destination. If no rule is found, the switch queries the SDN controller for further instructions. Simultaneously, the packet is analyzed by the IDS, such as Snort, which classifies flows as "good" or "malicious" based on predefined rules. If the IDS identifies a packet as part of an attack, it alerts the SDN controller, which responds by installing a blocking flow rule in the switch to discard packets from the malicious flow. For legitimate packets, the SDN controller configures the switch with appropriate forwarding rules, ensuring they reach their destination. This coordinated approach enables real-time detection and mitigation of malicious traffic while maintaining efficient and secure network operations

This diagram in figure 4 presents the series of events in an SDN-based automotive network that processes ICMP Echo Requests while incorporating Snort IDS and the RYU controller. The car sends an ICMP Echo Request, which is routed through the switch and NAT before reaching the server. In addition, a copy of the request is transmitted to Snort IDS for examination. If the IDS detects any malicious activity, it alerts the RYU controller, urging it to create a flow rule to prevent malicious traffic. The server validates the valid ICMP request and sends an ICMP Echo Reply back to the vehicle via the network channel, ensuring secure and efficient communication.

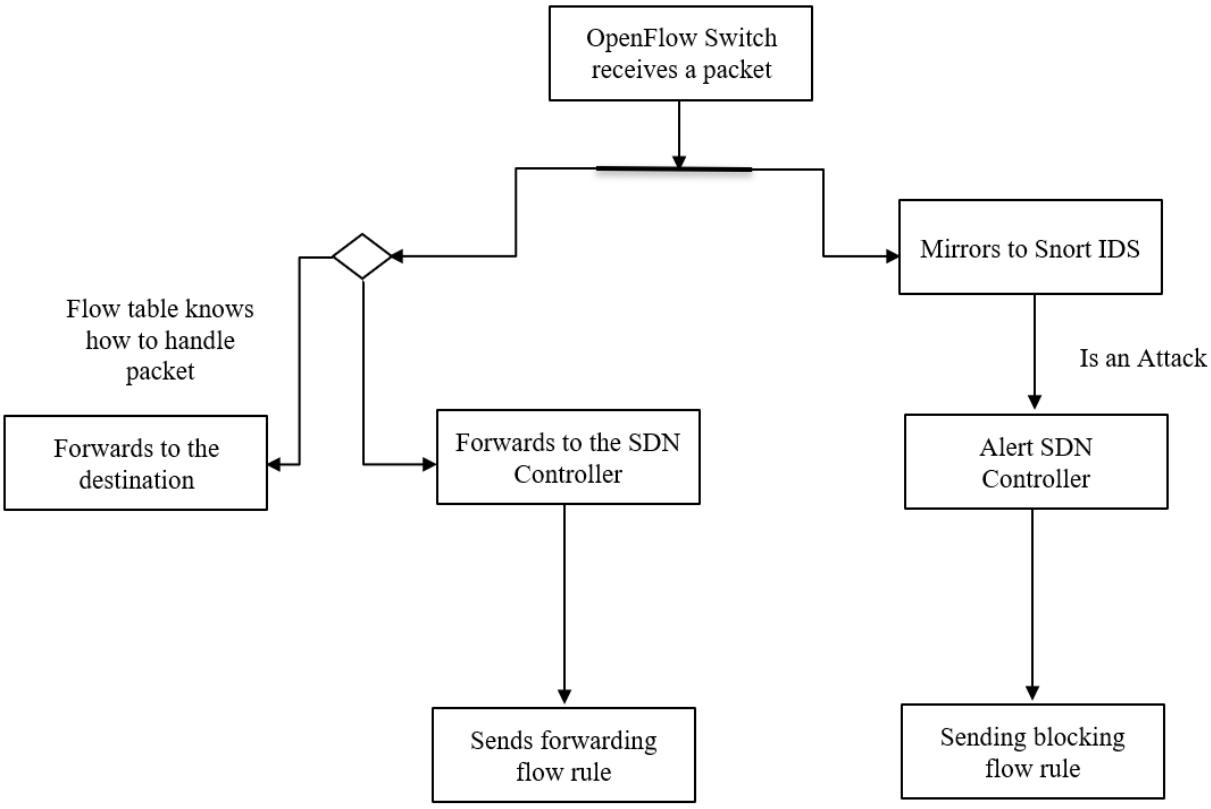

Figure 3: System workflow

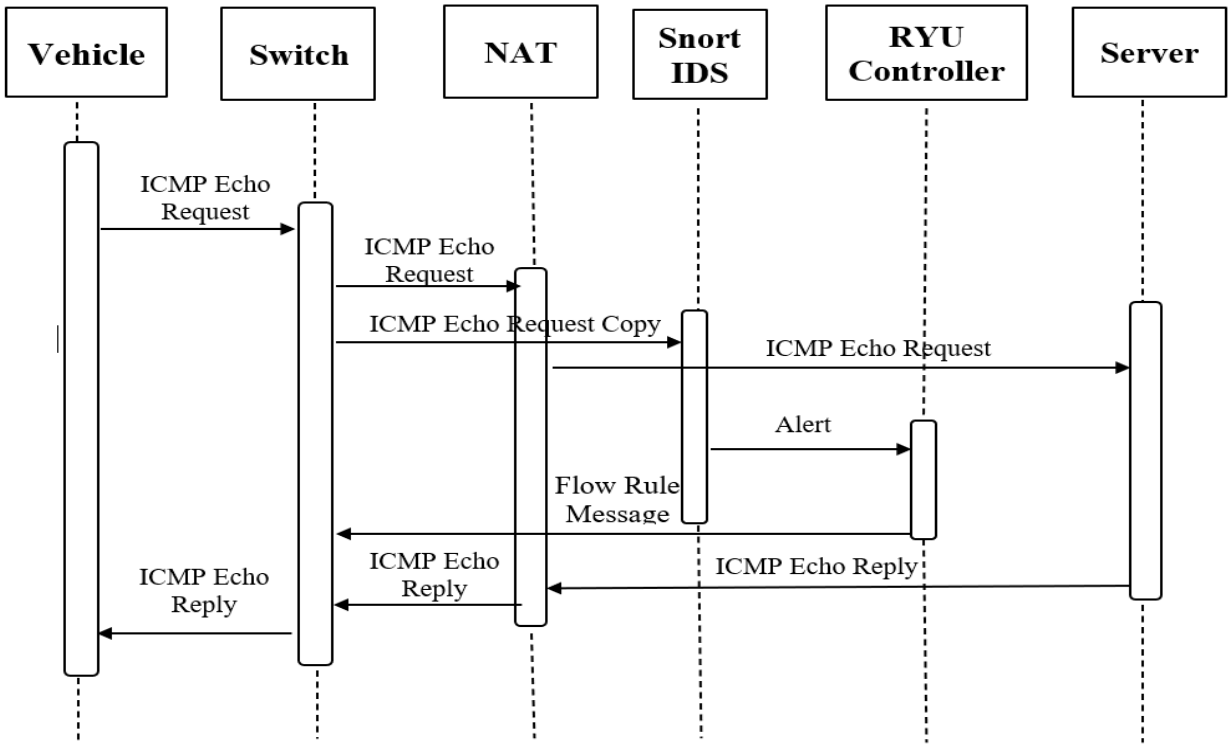

Figure 4: System message sequence diagram

4 Results and discussion

In this simulation, we carry out a DDoS attack on an SDVN network and include the Snort intrusion detection system. We then collect data before to and following the implementation of the Snort IDS.

4.1 Experiment configuration

Our experimental system was installed on a server running Ubuntu 20.04, with an Intel(R) Core (TM) i7-1165G7 CPU @ 2.80 GHz and 16 GB RAM.We used Mininet-WiFi [35] with the RYU controller to simulate a vehicular network consisting of 10 nodes representing vehicles [36]. The RYU controller and switches communicated using the OpenFlow protocol (version 1.3) [37]. We employed hping3 to simulate various DDoS attack types, including TCP flood and UDP flood attacks. This setup allowed us to rigorously evaluate the robustness of our DDoS detection framework by assessing Snort IDS's detection and mitigation capabilities under diverse threat scenarios. Additionally, iPerf was used to evaluate throughput, jitter, and packet loss, providing reliable performance metrics for analysis [38].

To create our SDN-based IDS, we merged the capabilities of an SDN controller and an IDS. The SDN controller in our configuration is based on RYU, an open-source Python framework. Within this controller, we created a Python method called process_snort_alert to handle IDS alerts. When an alert is identified as a "RYU block," the function sends a flow rule to the switch to prevent malicious traffic; otherwise, it ignores the traffic. For the IDS, we employed Snort, a widely used tool for detecting attacks through predefined rule signatures. We customized Snort by modifying the snort.conf file and adding specific DDoS detection rules in the ddos_detection.rules file. These rules were designed to address the DDoS attack scenarios tested in our setup, demonstrating the flexibility of our solution to handle different types of DDoS cyber-attacks.

4.2 Performance evaluation

The analysis highlights the impact of integrating Snort IDS on jitter behavior in an SDVN network. Without Snort IDS, jitter increases sharply and fluctuates widely, indicating network instability and inconsistent packet delivery, which can degrade the quality of service, particularly under attack conditions such as DDoS. In contrast, with Snort IDS, jitter remains stable and significantly reduced, demonstrating improved network performance through reduced delay variability and consistent packet delivery.

This is further illustrated in the figure 5, which compares jitter over a time period. The jitter without Snort IDS reflects network instability, while the jitter with Snort IDS highlights a more stable and resilient network. The integration of Snort IDS effectively mitigates the impact of attacks by detecting anomalies and applying countermeasures, enhancing the stability and reliability of SDVN environments.

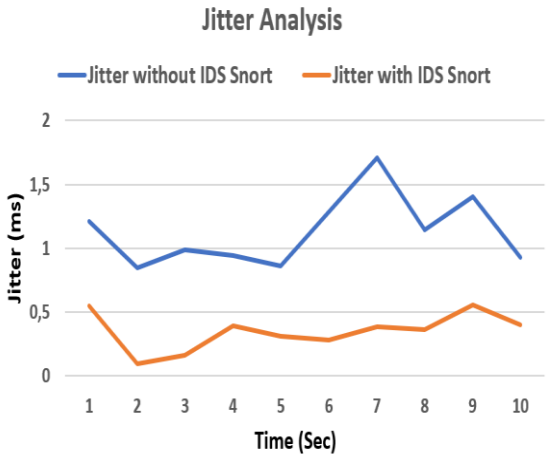

Figure 5: Comparison of Jitter in SDVN scenarios: with and without IDS Snort integration

Figure 6 show the impact of Snort IDS on the Packet Loss Rate (PLR) in the network. Without Snort IDS, the PLR is significantly higher, often exceeding 40%, indicating the adverse effects of undetected DDoS attacks, including resource overload and severe performance degradation. In contrast, with Snort IDS, the PLR is significantly reduced, typically ranging from 15% to 25%. These results highlight Snort's effectiveness in detecting and mitigating DDoS attacks by efficiently managing network traffic, thereby reducing packet loss and improving network reliability and performance.

.

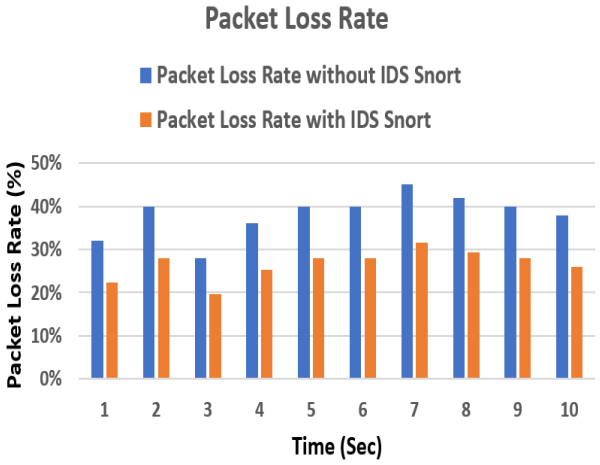

Figure 6: Comparison of Packet loss rate in SDVN scenarios: with and without IDS Snort integration

Figure 7 presents the impact of Snort IDS on network throughput. Without Snort IDS, the throughput remains low, fluctuating between approximately 5 Mbps and 7 Mbps, reflecting the effects of a DDoS attack that disrupts data transmission and overloads the network. In contrast, with Snort IDS, the throughput is consistently higher and stable, reaching nearly 10 Mbps. These results

demonstrate that integrating Snort IDS enhances network performance by effectively filtering malicious traffic, reducing congestion, and ensuring resource availability for legitimate users. This leads to a more reliable and high-performing network.

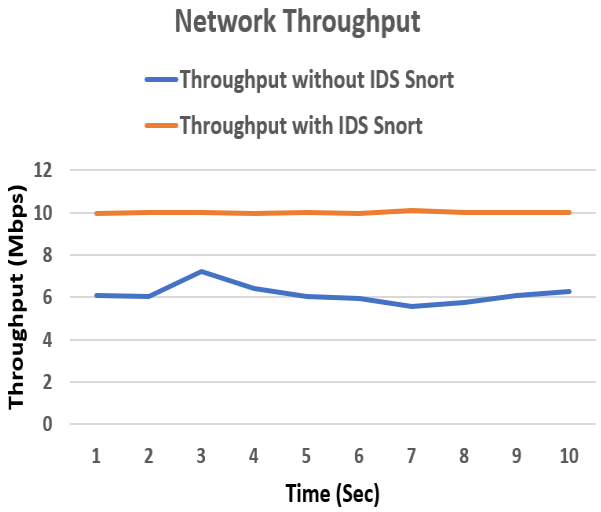

Figure 7: Comparison of throughput in SDVN scenarios: with and without IDS Snort integration

The assessment of network performance shows how implementing the Snort IDS has major advantages. Snort IDS continuously increases network efficiency and dependability across three important metrics: jitter, throughput, and packet loss rate. Throughput is increased, guaranteeing faster and more reliable data transmission rates; jitter is reduced, offering smoother performance for latency-sensitive applications; and packet loss is decreased, improving data delivery. These findings demonstrate that Snort IDS is a useful tool for performance optimization in real-time and high-demand network contexts since it not only improves security by thwarting possible threats but also raises network quality generally.

4.3 Discussion

The experimental results demonstrate the effectiveness of integrating Snort IDS in enhancing the stability and performance of SDVNs under DDoS attack scenarios. The inclusion of Snort leads to measurable improvements in key network metrics, including reduced jitter, decreased packet loss, and increased throughput. These outcomes validate Snort's role in real-time traffic monitoring and anomaly detection, contributing to improved network resilience in dynamic vehicular environments.

While these improvements—such as jitter reduction by up to 35%, packet loss rate reduction to 15–25%, and throughput stabilization around 10 Mbps—are significant, it is important to contextualize them within the demands of real-world ITS. Although reducing packet loss from over 40% to as low as 15% represents a substantial enhancement, a 25% loss rate remains relatively high for safety-critical applications. In scenarios like collision avoidance or emergency coordination, even minor packet loss or jitter can compromise communication reliability and jeopardize safety. Therefore, while Snort enhances baseline network stability, additional mechanisms—such as traffic prioritization, redundant paths, or edge-assisted processing—may be needed to meet the strict real-time requirements of modern ITS environments.

While Snort performs reliably in this context, its signature-based detection approach limits its ability to identify novel or zero-day attacks. By contrast, IDS solutions such as Suricata offer multi-threaded packet processing and higher throughput capabilities, while Zeek provides deeper contextual and behavioral traffic analysis. However, these alternatives often require more computational resources and involve more complex configurations. For lightweight and latency-sensitive environments like VANETs, Snort represents a practical trade-off between detection capability and system simplicity.

Nonetheless, Snort introduces computational overhead due to deep packet inspection and rule-based evaluation. Although resource limitations were not a bottleneck in the current simulation, real-world deployments—particularly in high-density or resource-constrained settings—may face scalability challenges. Addressing these issues may require lightweight or distributed IDS implementations, potentially integrated with fog or edge computing architectures to ensure sustained performance under load.

Another key consideration is that Snort's detection effectiveness strongly depends on the quality and relevance of its rule sets. A high false positive rate can result in unnecessary traffic blocking or mitigation actions, thereby reducing overall network efficiency. To mitigate this, effective rule tuning is essential. Future enhancements could involve automated rule optimization or the incorporation of machine learning-based anomaly detection to improve adaptability and responsiveness in dynamic vehicular scenarios. In summary, while Snort proves to be a viable and efficient solution for SDVN security enhancement, future research should focus on hybrid IDS frameworks and resource-aware deployment strategies to overcome scalability and detection limitations in real-world VANET implementations.

5 Conclusion

This study examined the detection and mitigation of DDoS attacks in SDVN networks using Snort IDS. By simulating a DDoS attack and analyzing network performance—jitter, packet loss, and throughput—before and after integrating Snort IDS, we observed significant improvements in network stability. The integration of Snort IDS resulted in reduced jitter, lower packet loss, and more consistent throughput, demonstrating its effectiveness in enhancing the resilience of SDVN networks against DDoS attacks. The IDS quickly detects and mitigates attacks, ensuring reliable communication and improved service quality. Additionally, the implementation of Snort IDS in an SDN-based VANET showed significant benefits in network performance, such

as reduced packet loss, increased throughput, and minimized jitter. These improvements are vital for the latency-sensitive nature of VANETs, where reliable communication is crucial for road safety and intelligent transportation applications. The use of SDN further enhances traffic management and centralized control, complementing the effectiveness of Snort IDS. This study emphasizes the combined benefit of SDN and IDS technologies in improving both security and performance in next-generation vehicular networks.

Future work will focus on enhancing the scalability, resilience, and intelligence of the proposed framework to support real-world deployment in large-scale VANETs. Although our experiments were conducted in a simulated environment, the modular design of SDN and Snort IDS enables distributed deployment across roadside infrastructure and vehicular edge nodes. To improve fault tolerance and reduce control latency, we plan to integrate a multi-controller SDN architecture, ensuring load balancing and redundancy across dynamic network regions. Precise benchmarking of alert-to-response latency using tools like tcpdump and controller logs will also be conducted to evaluate real-time mitigation capabilities. Additionally, we aim to incorporate entropy-based metrics for lightweight, dynamic anomaly detection and explore hybrid detection approaches that combine statistical analysis with machine learning models, such as Random Forest or SVM, to improve detection accuracy and adaptability. Context-aware detection and behavioral profiling of vehicles will further enhance the system's ability to distinguish between benign and malicious traffic in real-time ITS scenarios.

References

[1] Al-shareeda, M. A., Alazzawi, M. A., Anbar, M., Manickam, S., & Al-Ani, A. K. (2021, July). A comprehensive survey on vehicular ad hoc networks (vanets). In *2021 International Conference on Advanced Computer Applications (ACA)* (pp. 156-160). IEEE.

[2] Mundhe, P., Verma, S., & Venkatesan, S. J. C. S. R. (2021). A comprehensive survey on authentication and privacy-preserving schemes in VANETs. Computer Science Review, 41, 100411.

[3] Pavithra, T., & Nagabhushana, B. S. (2020, July). A survey on security in VANETs. In *2020 second international conference on inventive research in computing applications (ICIRCA)* (pp. 881-889). IEEE.

[4] Mekki, T., Jabri, I., Rachedi, A., & Chaari, L. (2022). Software-defined networking in vehicular networks: A survey. *Transactions on Emerging Telecommunications Technologies*, *33*(10), e4265.

[5] Sultana, R., Grover, J., & Tripathi, M. (2021). Security of SDN-based vehicular ad hoc networks: State-of-the-art and challenges. *Vehicular Communications*, *27*, 100284.

[6] Nisar, K., Jimson, E. R., Hijazi, M. H. A., Welch, I., Hassan, R., Aman, A. H. M., ... & Khan, S. (2020). A survey on the architecture, application, and security of software defined networking: Challenges and open issues. *Internet of Things*, *12*, 100289.

[7] Arif, M., Wang, G., Geman, O., Balas, V. E., Tao, P., Brezulianu, A., & Chen, J. (2020). Sdn-based vanets, security attacks, applications, and challenges. *Applied Sciences*, *10*(9), 3217.

[8] Mohammed, B. A. (2022). Review on Software-Defined Vehicular Networks (SDVN). *IJCSNS*, *22*(9), 376.

[9] Houmer, M., Ouaissa, M., Ouaissa, M., & Hasnaoui, M. (2020). SE-GPSR: Secured and enhanced greedy perimeter stateless routing protocol for vehicular ad hoc networks.

[10] Houmer, M., Ouaissa, M., & Ouaissa, M. (2022). Secure authentication scheme for 5g-based v2x communications. *Procedia Computer Science*, *198*, 276-281.

[11] Kumar, R., & Agrawal, N. (2023). A survey on software-defined vehicular networks (SDVNs): a security perspective. *The Journal of Supercomputing*, *79*(8), 8368-8400.

[12] Hussein, N. H., Yaw, C. T., Koh, S. P., Tiong, S. K., & Chong, K. H. (2022). A comprehensive survey on vehicular networking: Communications, applications, challenges, and upcoming research directions. *IEEE Access*, *10*, 86127-86180.

[13] Raut, R. M., & Asole, S. (2023, April). A Survey on Security Threats in VANET and Its Solutions. In *International Conference on Recent Trends in Artificial Intelligence and IoT* (pp. 229-240). Cham: Springer Nature Switzerland.

[14] Sheikh, M. S., Liang, J., & Wang, W. (2019). A survey of security services, attacks, and applications for vehicular ad hoc networks (vanets). *Sensors*, *19*(16), 3589.

[15] Boucetta, S. I., & Johanyák, Z. C. (2022, May). Survey on security attacks in software defined VANETs. In *2022 IEEE 16th International Symposium on Applied Computational Intelligence and Informatics (SACI)* (pp. 000185-000190). IEEE.

[16] Ayodele, B., & Buttigieg, V. (2024). SDN as a defence mechanism: a comprehensive survey. *International Journal of Information Security*, *23*(1), 141-185.

[17] Chouikik, M., Ouaissa, M., Ouaissa, M., Boulouard, Z., & Kissi, M. (2022). Software-defined networking security: A comprehensive review. *Big Data Analytics and Computational Intelligence for Cybersecurity*, 91-108.

[18] Carrascal, D., Rojas, E., Arco, J. M., Lopez-Pajares, D., Alvarez-Horcajo, J., & Carral, J. A. (2023). A comprehensive survey of in-band control in sdn: Challenges and opportunities. *Electronics*, *12*(6), 1265.

[19] Aldaoud, M., Al-Abri, D., Awadalla, M., & Kausar, F. (2023). Leveraging ICN and SDN for future internet architecture: a survey. *Electronics*, *12*(7), 1723.

[20] Maleh, Y., Qasmaoui, Y., El Gholami, K., Sadqi, Y., & Mounir, S. (2023). A comprehensive survey on SDN security: threats, mitigations, and future

directions. *Journal of Reliable Intelligent Environments*, *9*(2), 201-239.

[21] Shaji, N. S., & Muthalagu, R. (2023). Survey on security aspects of distributed software-defined networking controllers in an enterprise SD-WLAN. *Digital Communications and Networks*.

[22] Chouikik, M., Ouaissa, M., Ouaissa, M., Boulouard, Z., & Kissi, M. (2023, July). Impact of DoS attacks in software defined networks. In *AIP Conference Proceedings* (Vol. 2814, No. 1). AIP Publishing.

[23] Kumar, R., & Agrawal, N. (2023). A survey on software-defined vehicular networks (SDVNs): a security perspective. *The Journal of Supercomputing*, *79*(8), 8368-8400.

[24] Türkoğlu, M., Polat, H., Koçak, C., & Polat, O. (2022). Recognition of DDoS attacks on SD-VANET based on combination of hyperparameter optimization and feature selection. *Expert Systems with Applications*, *203*, 117500.

[25] Elubeyd, H., & Yiltas-Kaplan, D. (2023). Hybrid deep learning approach for automatic DoS/DDoS attacks detection in software-defined networks. *Applied Sciences*, *13*(6), 3828.

[26] Setitra, M. A., & Fan, M. (2024). Detection of DDoS attacks in SDN-based VANET using optimized TabNet. *Computer Standards & Interfaces*, *90*, 103845.

[27] Ma, R., Wang, Q., Bu, X., & Chen, X. (2023). Real-time detection of DDoS attacks based on random forest in SDN. *Applied Sciences*, *13*(13), 7872.

[28] Adedeji, K. B., Abu-Mahfouz, A. M., & Kurien, A. M. (2023). DDoS attack and detection methods in internet-enabled networks: Concept, research perspectives, and challenges. *Journal of Sensor and Actuator Networks*, *12*(4), 51.

[29] Chouikik, M., Ouaissa, M., Ouaissa, M., Boulouard, Z., & Kissi, M. (2024). Detection and mitigation of DDoS attacks in SDN based intrusion detection system. *Bulletin of Electrical Engineering and Informatics*, *13*(4), 2750-2757.

[30] Su, Y., Xiong, D., Qian, K., & Wang, Y. (2024). A Comprehensive Survey of Distributed Denial of Service Detection and Mitigation Technologies in Software-Defined Network. *Electronics*, *13*(4), 807.

[31] Pandey, S. K., & Sinha, D. (2024). A Novel Approach for Detection of DoS/DDoS Attack in Network Environment using Hellinger Distance Technique. *Informatica*, *48*(20).

[32] Zobary, F. (2024). Optimizing SDN Controller to Switch Latency for Controller Placement Problem. *Informatica*, *48*(8).

[33] Deneke, B. B., Beyene, A. M., & Haile, E. A. (2024). Improving Software Defined Network controllers in a multi-vendor environment. *Heliyon*, *10*(4).

[34] Zhang, W., Jing, S., & Zhao, C. (2023, June). A Survey of SDN Data Plane Attacks and Defense Strategies. In *Proceedings of the 2023 2nd International Conference on Networks, Communications and Information Technology* (pp. 59-65).

[35] Fontes, R. R., Afzal, S., Brito, S. H., Santos, M. A., & Rothenberg, C. E. (2015, November). Mininet-WiFi: Emulating software-defined wireless networks. In *2015 11th International conference on network and service management (CNSM)* (pp. 384-389). IEEE.

[36] Fontes, R. D. R., & Rothenberg, C. E. (2016, August). Mininet-wifi: A platform for hybrid physical-virtual software-defined wireless networking research. In *Proceedings of the 2016 ACM SIGCOMM Conference* (pp. 607-608).

[37] Ram, A., Dutta, M. P., & Chakraborty, S. K. (2024). A Flow-Based Performance Evaluation on RYU SDN Controller. *Journal of The Institution of Engineers (India): Series B*, *105*(2), 203-215.

[38] Zieliński, B. (2023). Assessment of iPerf as a Tool for LAN Throughput Prediction. *International Journal of Electronics and Telecommunications*, 523-528.