ESCUELA NAVAL DE CADETES "ALMIRANTE PADILLA" Honor y Tradición

OnBoard Knowledge Journal

Article

Network Security against DDoS attacks using the Floodlight controller

Seguridad en la Red contra ataques DDoS utilizando el controlador Floodlight

Eylin Ortega Buelvas [1], Carlos Martínez Guerra [1] and Jorge Gómez Gómez [1]*

[1] Faculty of Engineering, Systems Engineering, Universidad de Córdoba, Montería, 800001, Colombia; eortegabuelvas76@correo.unicordoba.edu.co; cmartinezguerra93@correo.unicordoba.edu.co; jeliecergomez@correo.unicordoba.edu.co

* Correspondence: jeliecergomez@correo.unicordoba.edu.co

Abstract: Nowadays, the Internet has become one of the most widely used platforms for conducting transactions and managing information worldwide, which highlights the need to strengthen cybersecurity systems against increasing digital threats. Among the most common attacks are Distributed Denial of Service (DDoS) attacks, which aim to disrupt the operation of networks, services, or websites by overwhelming them with massive amounts of malicious traffic, exhausting their resources and preventing legitimate requests from being processed. This project aimed to analyze and implement a simulation environment to assess service vulnerability to DDoS attacks and propose early detection and mitigation strategies. A network architecture based on Software Defined Networking (SDN) was designed using the Floodlight controller, enabling dynamic traffic management and anomaly detection. Controlled attack simulations were conducted to observe system behavior and evaluate its response capacity. The results demonstrated that integrating the Floodlight controller significantly improves detection and response to traffic saturation events, reducing the likelihood of critical service disruptions. In conclusion, SDN based environments represent an effective alternative to enhance cybersecurity in network infrastructures, providing a flexible framework for monitoring, prevention, and mitigation of DDoS.

Keywords: DDoS attack; Floodlight; Cyber attack; Simulation.

Resumen: En la actualidad, Internet constituye uno de los medios más utilizados para la realización de transacciones y la gestión de información a nivel mundial, lo que resalta la necesidad de fortalecer la seguridad de los sistemas informáticos ante el incremento de amenazas cibernéticas. Entre los ataques más frecuentes se encuentran las Denegaciones de Servicio Distribuidas (DDoS), cuyo propósito es interrumpir el funcionamiento de redes, servicios o sitios web mediante el envío masivo de tráfico malicioso, saturando los recursos del sistema e impidiendo la atención de so-licitudes legítimas. El objetivo de este proyecto fue analizar e implementar un entorno de simu-lación que permitiera evaluar la vulnerabilidad de los servicios ante ataques DDoS y proponer estrategias de detección y mitigación temprana. Para ello, se diseñó una arquitectura de red basada en Software Defined Networking (SDN) empleando el controlador Floodlight, que facilita la administración dinámica del tráfico y la identificación de patrones anómalos. Se realizaron pruebas controladas de ataques simulados para observar el comportamiento del sistema y medir su capacidad de respuesta. Los resultados evidenciaron que la integración del controlador Floodlight permite mejorar significativamente la detección y respuesta ante eventos de saturación de tráfico, reduciendo la probabilidad de interrupciones críticas. En conclusión, la utilización de entornos SDN constituye una alternativa

Citation: Ortega, E.; Martínez, C.; Gómez, J. . Network Security against DDoS attacks using the Floodlight controller. *OnBoard Knowledge Journal* **2025**, 2, 1. https://doi.org/

Received: 17/07/2025
Accepted: 20/09/2025
Published: 24/10/2025

eficaz para fortalecer la ciberseguridad en infraestruc-turas de red, brindando un marco adaptable para el monitoreo, la prevención y la mitigación de ataques DDoS.

Palabras clave: Ataque DDoS; Floodlight; Ataque cibernético; Simulación.

1. Introdution

In the current context, the Internet is a key tool for transactions and services at a global level, which highlights the importance of protecting computer systems against threats such as Distributed Denial of Service (DDoS) attacks. These attacks aim to saturate the resources of networks or applications through massive traffic of malicious requests, affecting the availability and performance of legitimate services [6]. Software-Defined Networks (SDN) offer an effective solution to mitigate these attacks by centralizing traffic management and enabling rapid responses to anomalous patterns [2]. Controllers such as Floodlight stand out for their ability to monitor and filter malicious traffic, improving network security. In addition, the use of simulators such as Mininet is crucial for testing mitigation strategies in a controlled environment, allowing systems to be trained, vulnerabilities identified, and their behavior evaluated against possible attacks.

Various authors have addressed DDoS attack mitigation in SDN environments from different perspectives. According to [4], software-defined networks (SDN) represent a paradigm shift in communication networks by separating the control and data planes, using a central controller (in this case OpenDayLight) that allows for more flexible and adaptive management. This undergraduate thesis develops a defense mechanism against Denial of Service (DDoS) attacks, specifically the HTTP flooding type, implementing a dual strategy: proactive (periodically rejuvenating replicas) and reactive (responding to detected threats). The system was implemented in a simulated environment using Mininet, with a web server distributed across several nodes, and experimental tests demonstrated that the mechanism provides an additional effective layer of security against DDoS attacks while maintaining quality of service.

From a broader perspective, the use of controllers in Software-Defined Networks (SDN) has become a fundamental strategy for protection against DDoS attacks, especially when these involve IoT devices. In the SDN context, controllers centralize network management and allow precise traffic monitoring. This approach facilitates the rapid identification of unusual traffic patterns characteristic of a DDoS attack and enables the application of immediate countermeasures, such as blocking suspicious IPs or limiting bandwidth to mitigate the network impact. This study highlights how SDN controllers, in combination with IoT devices, offer an effective detection and response capability in a controlled environment, thus testing network efficiency and minimizing the effects of DDoS attacks on affected servers.

According to [8], to face DDoS attacks, network administration must be centralized, and real-time defense strategies must be applied. By implementing a centralized controller in an SDN, it is possible to monitor network traffic, detect attack patterns, and make rapid decisions to mitigate their impact by isolating malicious connections and maintaining service for legitimate users. Recent studies demonstrate that the use of controllers in SDN provides an additional layer of security against DDoS attacks, optimizing network resource use and minimizing response time through simulated attacks in controlled environments, as achieved with tools such as NeSSi2, a simulator that allows these scenarios to be modeled and studied in depth.

Similarly, [1] offer an effective solution against DDoS attacks. By using robust firewall configurations and software modules such as IPTables and MOD_EVASIVE, SDN control systems can significantly reduce the load of malicious packets on servers, blocking suspicious IP addresses and preventing malicious traffic from affecting critical network resources. This strategy is particularly effective, as SDN platforms allow for automatic

blocking configurations and request limiting policies on critical servers, reducing up to 80% of the impact of DDoS attacks on network infrastructure.

In this context, the present study focuses on the implementation of the Floodlight controller in simulated environments, with the purpose of strengthening detection and response strategies against DDoS attacks. It seeks to highlight the role of simulators in the evaluation and improvement of the defensive capabilities of network infrastructure, thereby contributing to the enhancement of cybersecurity.

2. Theoretical Framework

2.1. Definition and Types of DDoS Attacks

According to [9], Distributed Denial of Service (DDoS) attacks have the main objective of interrupting the normal operation of a system, network, or service by overloading its resources with an excessive volume of traffic. These attacks are classified into three main categories: volumetric, which generate a large volume of data to overload the system; protocol-based, which exploit weaknesses in network protocols to exhaust resources; and application-layer attacks, which directly affect applications or services such as web servers through disguised malicious requests. Each of these types uses different methods and metrics to measure its impact, such as bits per second (bps) or requests per second (rps).

2.2. Impact on Networks and Systems

DDoS attacks can cause significant consequences, such as the interruption of essential services, economic losses due to the inaccessibility of resources, and damage to the reputation of organizations. Their growth has been exponential, with an increasing number of devices connected to the network becoming targets or tools within botnets, which increases both their frequency and magnitude. Among the motivations for carrying out this type of attack are financial gain and personal interests [9].

2.3. Architecture and Principles of SDN

SDN separates the control plane (where network decisions are made) from the data plane (responsible for packet forwarding), centralizing network intelligence in a software controller. This facilitates the configuration and management of the network through programming, improving scalability and flexibility. The SDN architecture includes three main layers:

- Infrastructure layer: Contains devices such as switches and routers that execute the controller's instructions.
- Control layer: Where the SDN controller operates, managing forwarding decisions.
- Application layer: Integrates services and applications through specific interfaces (Northbound APIs).

This model allows networks to be configured and managed dynamically and openly, eliminating the limitations of traditional hardware-based approaches.

2.4. Advantages of SDN in Network Management

According to [10], SDN provides key advantages in network management, such as centralization, which allows efficient control from a single point; flexibility to adapt the network to new services; automation, which minimizes human errors and speeds up configurations; and scalability, facilitating the incorporation of new devices through open standards such as OpenFlow. These characteristics make SDN ideal for dynamic environments such as data centers.

2.5. Role of SDN Controllers in Network Security

The SDN controller is fundamental to network security, as it centralizes policy management, detects and mitigates threats, and constantly monitors the state of the network to respond to attacks in real time. In addition, it facilitates the integration of advanced

security systems such as firewalls and IDS/IPS. However, as a critical point, it requires robust protection measures such as encryption, authentication, and forensic analysis to prevent vulnerabilities [10].

2.6. Floodlight: Characteristics and Functionalities

Floodlight is a Software-Defined Networking (SDN) controller based on the OpenFlow protocol, developed in the Java programming language. This controller stands out for its compatibility with a wide range of devices, both physical and virtual switches, and for offering tools that facilitate the management of complex network topologies. Among its main functionalities are the creation of access control rules, remote device monitoring, and real-time visualization of the network status. In addition, Floodlight allows administrators to configure the network easily, ensuring high performance, fault tolerance, and robust security options such as user authentication, authorization, and anomaly detection [10].

2.7. Security Modules in Floodlight

This controller can include several security functionalities designed to protect SDN networks. Its architecture allows the implementation of access rules through Access Control Lists (ACLs) as well as traffic segmentation policies through VLANs. It also supports advanced functions such as anomaly detection in traffic, fault recovery, and authentication and authorization tools to guarantee secure access to the controller and connected devices. These modules make Floodlight an effective solution for managing networks in a centralized manner with high levels of security [7].

2.8. Floodlight Compatibility with Simulation Environments

It is widely used in simulation environments thanks to its integration with tools such as Mininet and MiniEdit, which allow the design and testing of SDN networks without the need for physical hardware. Its ability to operate on virtualized platforms such as VirtualBox, along with its support for OpenFlow, makes it ideal for educational and experimental environments. These characteristics allow users to emulate complex networks, verify host connectivity, analyze packet traffic, and optimize performance before implementing them in a physical environment [7].

2.9. Mininet for the Creation of Virtual Topologies

It is a widely used network simulation tool for emulating virtual topologies (Fig 1). It allows the creation of hosts, switches, controllers, and virtual links on a physical machine, making use of optimized resources. One of its main features is the ability to design customized topologies through Python scripts or via the graphical interface Miniedit. In addition, Mininet supports OpenFlow compatible switches, making it an ideal platform for testing SDN architectures. It is a scalable, realistic, and easy-to-use solution for experimenting with complex networks without the need for physical hardware [5].

python Copiar código

```python
# Ejemplo básico de una red virtual en Mininet
from mininet.net import Mininet
net = Mininet()
h1 = net.addHost('h1', ip='10.0.0.1')
h2 = net.addHost('h2', ip='10.0.0.2')
s1 = net.addSwitch('s1')
net.addLink(h1, s1)
net.addLink(h2, s1)
net.start()
net.pingAll()
net.stop()
```

Figure 1. Creation of a topology.

2.10. Use of ICMP Tools such as Ping for Connectivity Testing

Ping is a fundamental tool that uses the Internet Control Message Protocol (ICMP) to verify connectivity between devices in a network (Fig 2. In simulations performed with Mininet, it allows verification that nodes are properly interconnected. In addition, it measures packet latency and diagnoses issues such as data loss or high response time. This simple yet powerful command is essential for validating configurations in virtual networks and ensuring their operability [5].

```
bash                                                    Copiar código

mininet> h1 ping h2
```

Figure 2. Verification of connectivity between h1 and h2.

2.11. Simulation of Traffic and Attacks in Controlled Environments

Mininet allows the emulation of network traffic and attack scenarios in a safe and controlled environment, facilitating the evaluation of security configurations and network performance. For example, it is possible to generate traffic between nodes using tools such as iperf to measure bandwidth or to simulate DoS attacks to analyze network resilience (Fig 3). These simulations are useful for testing security policies, such as Access Control Lists (ACL) and firewall configurations, before their implementation in real networks [5].

```
bash                                                    Copiar código

mininet> iperf h1 h2
```

Figure 3. Simulation of traffic and attacks.

2.12. Monitoring and Analysis of Anomalous Traffic

Monitoring and analyzing anomalous traffic is based on the constant observation of the network to identify unusual behavior patterns that may indicate a DDoS attack. This strategy uses advanced tools such as Intrusion Detection Systems (IDS) and packet analysis to identify abnormal data flows. For example, sudden spikes in traffic volume, suspicious IP addresses, or repeated attempts to access restricted resources are monitored. These indicators allow for the implementation of rapid responses to mitigate the impact, such as temporarily blocking suspicious addresses or limiting bandwidth [3].

2.13. Filtering of Malicious Packets

This is a mitigation technique that involves inspecting incoming data packets to detect harmful content. Through the use of firewalls, Access Control Lists (ACL), and Deep Packet Inspection (DPI) technologies, suspicious or malicious requests are discarded before they reach the servers. This process includes identifying attack signatures, malicious IP addresses, and traffic patterns that match known attacks, ensuring that only legitimate packets gain access to network resources [3].

3. Materials and Methods

3.1. Preparation of the Test Environment

To carry out the DDoS attack mitigation experiment, a controlled environment was created to replicate the behavior of a real network. This process involved the use of specific tools such as Mininet for network simulation, Floodlight as the SDN controller, and a custom script designed to monitor and analyze network traffic.

First, the Floodlight controller (Fig. 4) was executed, which was responsible for managing the network workflows.

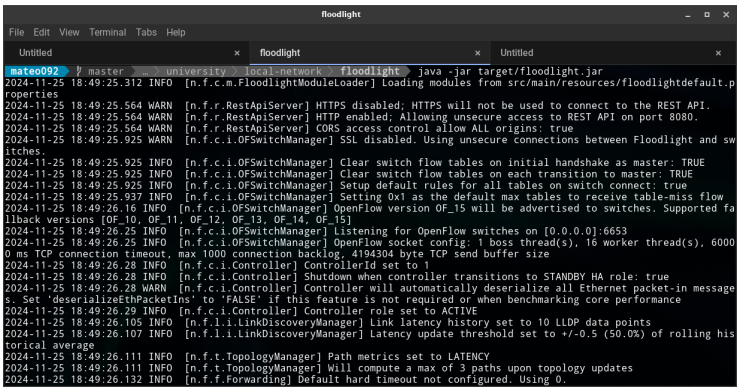


Figure 4. Execution of the Floodlight controller.

Second, the script (Fig. 5) responsible for monitoring possible DDoS attacks was run. This script records and displays detailed statistics in the console, including information about monitored hosts and those that have been blocked as part of the mitigation measures.

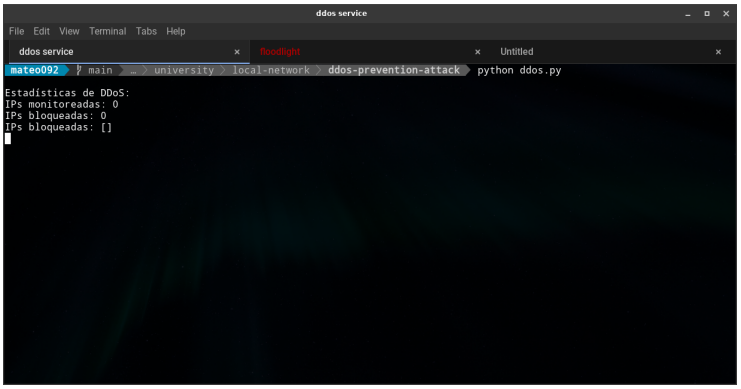


Figure 5. Execution of the DDoS attack prevention script.

Subsequently, a network topology was configured, consisting of a central switch connected to seven hosts (Fig. 6), representing a typical network scenario. Figure 7 graphically illustrates this topology as visualized in the controller's web interface.

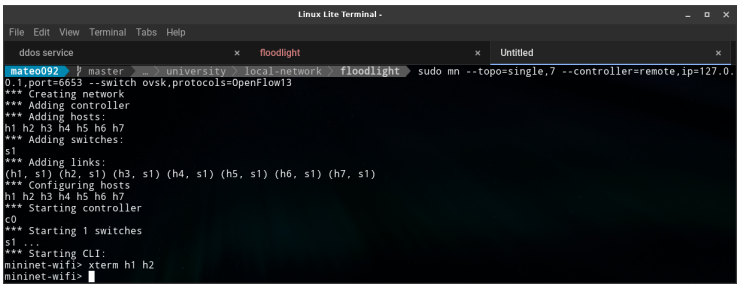


Figure 6. Execution of the topology in Mininet.

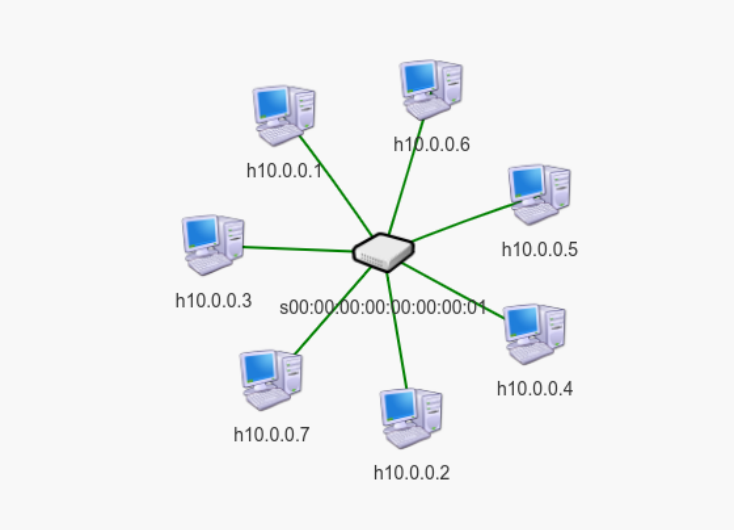


Figure 7. Network topology simulated in Mininet.

3.2. DDoS Attack Simulation

Once the test environment configuration was completed, the generation of normal traffic in the network began (Fig. 8), using the ping command between host1 and host3. As a result, responses were obtained from host3 along with their statistics, demonstrating that connectivity between both hosts was functioning correctly.

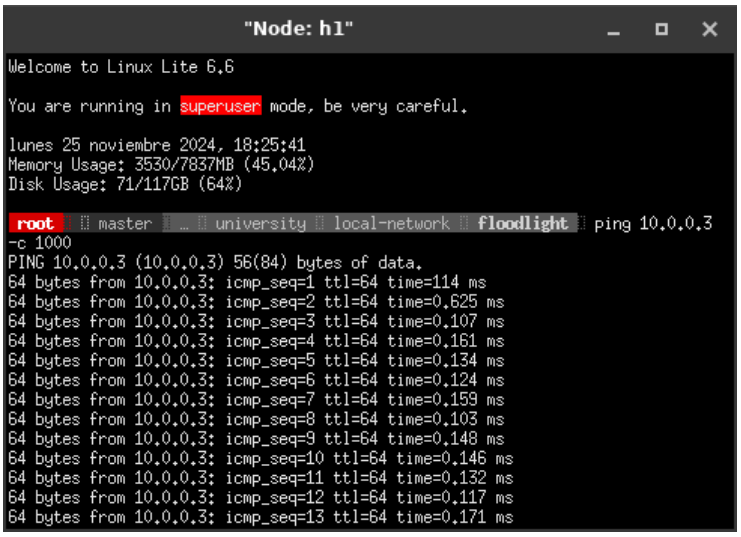


Figure 8. Execution of the ping command between host1 and host3.

Finally, to simulate the DDoS attack, the ping command with the -f parameter was used between host2 and host4. In Figure 9, it can be observed that the first packets sent are correctly responded to by host4; however, once the massive sending of packets begins, they are represented in the console as dots, highlighting the intensity of the generated traffic.

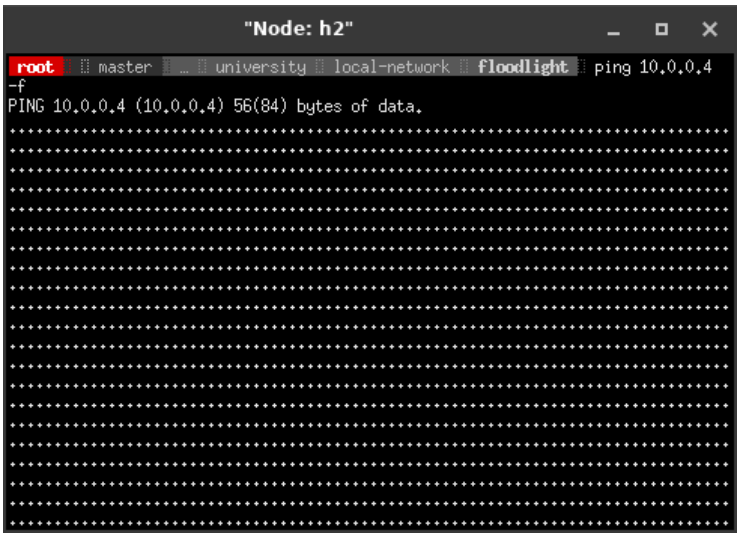


Figure 9. Execution of the ping command between host1 and host2 with the -f parameter.

4. Results

During the execution of the experimental section, a detailed monitoring of the number of packets sent over time was carried out, covering the different scenarios considered (Fig. 10). This analysis made it possible to identify traffic patterns under both normal conditions and during the simulation of a DDoS attack.

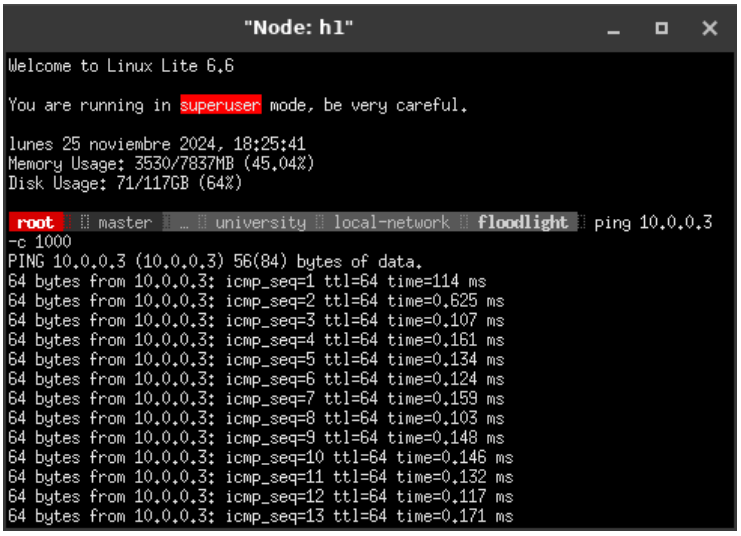


Figure 10. Normal packet traffic rate between hosts (IP: 10.0.0.1, 10.0.0.3, 10.0.0.2, and 10.0.0.4) as a function of time.

Figure 11 shows the packet traffic sent by different hosts over time. The orange line represents the traffic between hosts 10.0.0.1 and 10.0.0.3, showing a constant increase. Meanwhile, the brown line reflects the rise in packet transmission from host 10.0.0.2 to host 10.0.0.4, which initially corresponds to a normal traffic environment.

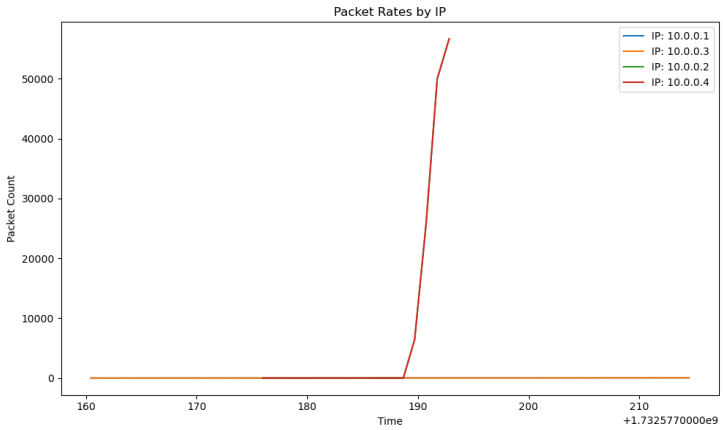


Figure 11. Exponential increase in packet traffic rate during the DDoS attack.

After a few seconds, host 4 begins to receive more than 50,000 packets, indicating an unusual behavior compared to the other hosts. This abrupt increase stands out in the graph, making the traffic sent to host 3 appear insignificant in relative terms (Fig. 12).

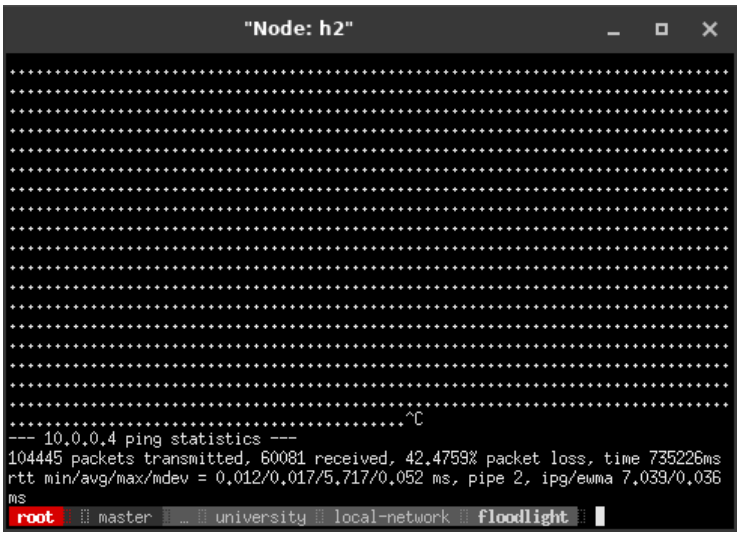


Figure 12. Network statistics after DDoS attack mitigation.

Once the script in charge of preventing DDoS attacks detects this unusual behavior, it sends the necessary rules to the controller to block the host generating the massive traffic. This prevents further resource consumption and mitigates the impact of the attack.

In the ping command statistics, out of the 104,445 packets transmitted over a 12-minute period, 42% were not received by the host. This packet loss is due both to the impact of the massive traffic and to the action of the script responsible for blocking the sending host, thus mitigating excessive resource consumption in the network.

5. Conclusions

This study evaluated the effectiveness of the Floodlight controller in mitigating DDoS attacks within environments based on Software-Defined Networks (SDN). The experimental results demonstrated that the implemented architecture allows the identification and blocking of anomalous traffic patterns in real time, significantly reducing resource saturation and packet loss, which in the test scenarios reached up to 42% before system intervention.

The simulation in Mininet made it possible to analyze network behavior in a controlled manner under different levels of attack, showing that the automatic response of

the controller prevents critical interruptions and improves service availability. This approach confirms that SDNs represent an efficient and scalable alternative to strengthen cybersecurity in network infrastructures.

Likewise, the integration of Floodlight with detection and filtering modules proved to be a viable strategy for implementing adaptive defense policies, adjustable to the dynamic conditions of traffic. It is recommended to further explore future research lines aimed at incorporating machine learning and predictive analysis techniques to anticipate malicious behavior and optimize response times to security incidents.

Overall, the findings of this research provide practical evidence of the capability of SDN controllers to manage complex environments, offering a solid framework for the development of preventive and mitigation solutions against DDoS attacks in modern networks.

Author Contributions: **Eylin Ortega:** Software, Visualization, Validation, Formal analysis, Investigation, Resources.
Carlos Martínez: Conceptualization, Methodology, Writing – original draft, Writing – review & editing.
Jorge Gómez: Data curation, Supervision, Project administration, Funding acquisition.
All authors have read and agreed to the published version of the manuscript. Please refer to the CRediT taxonomy for the definitions of the terms. Authorship is limited to those who have made substantial contributions to the reported work.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable, since the present study does not involvehuman personnel or animals.

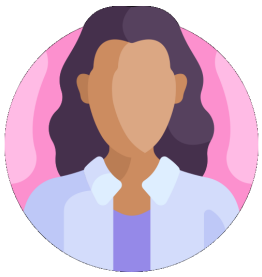
Informed Consent Statement: This study is limited to the use of technological resources, so nohuman personnel or animals are involved.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. (2016). Evaluación de ataques de denegación de servicio dos y ddos, y mecanismos de protección. *GEEKS DECC-REPORTS*, 2(1).
2. Alenezi, M., Al-Haija, Q. A., and Alqaralleh, B. (2023). Intelligent ddos attack detection and mitigation in sdn: A hybrid machine learning approach. *IEEE Access*, 11:45678–45691.
3. Clavo Tafur, C. J. (2022). Análisis comparativo de técnicas de mitigación de ataque de ddos en cloud computing. Trabajo académico.
4. Cortés, J. P. (2016). Defensa proactiva y reactiva ante ataques ddos en un entorno simulado de redes definidas por software. Trabajo académico.
5. Guanoluisa Jaramillo, E. D. (2019). Diseño de la arquitectura de una red sdn mediante el protocolo openflow con simulación en el software mininet para la infraestructura de una pymes.
6. Kaur, P. and Singh, A. (2022). A comprehensive survey on ddos attacks and defense mechanisms in cloud and software-defined networks. *Computer Networks*, 211:108964.
7. Melendres Del Pezo, S. M. (2023). Propuesta de implementación de una red sdn por medio del controlador floodlight y mininet para la institución unidad educativa americano.
8. Molina, L., Cevallos, Y., Machado, G., and Furfaro, A. El enemigo moderno: Ataques ddos en la capa de red y de aplicación. Publicación académica.
9. Pérez Gómez, P. (2017). Estudio de ataques ddos basados en dns.
10. Ruipérez Cuesta, J. (2021). *Seguridad en Redes definidas por software (SDN)*. Doctoral dissertation, Universitat Politècnica de València, Valencia, España.

Authors' Biography



Eylin Ortega Buelvas Robotics and programming teacher.

Carlos Martínez Guerra Systems Engineer, Backend developer specialized in RESTful APIs with Laravel and Spring Boot, frontend with Vue.js.

Jorge Gómez Gómez Research professor at the Faculty of Systems Engineering.