IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Ingénierie des Systèmes d'Information
Vol. 29, No. 1, February, 2024, pp. 301-311
Journal homepage: http://iieta.org/journals/isi

Inter and Intra Domain DDoS Attack Mitigation for Software Defined Network Based on Hyperledger Fabric Blockchain Technology

Wurood Sadik Khorseed[1], Ali Hussein Hamad[2*]

[1] Informatics Institute for Postgraduate Studies, Commission for Computers and Informatics, Baghdad 10011, Iraq
[2] Department of Information and Communication Engineering, University of Baghdad, Baghdad 10011, Iraq

Corresponding Author Email: ahamad@kecbu.uobaghdad.edu.iq

https://doi.org/10.18280/isi.290130

Received: 7 October 2023
Revised: 1 December 2023
Accepted: 7 December 2023
Available online: 27 February 2024

Keywords:
DDoS, attack mitigation, SDN, Hyperledger Fabric, smart contract, inter domain, intra domain

ABSTRACT

The widespread adoption of Internet of Things devices has led to a significant rise in security concerns. Attackers can exploit the vulnerability of centralized control in software-defined networks (SDN) through distributed denial of service (DDoS) attacks on these networks. The concentration of control within a network introduces novel vulnerabilities and potential avenues for attacks. The present strategies employed for mitigating DDoS attacks face challenges arising from their constrained adaptability, inadequate allocation of resources, and reduced flexibility. The developing technology of blockchain offers a robust solution for cost-effective, optimized, and adaptable mitigation of inter and intra-domain SDN against DDoS attacks. This work utilizes the Hyperledger Fabric platform, a permissioned blockchain, to examine the detection of DDoS attacks using the entropy approach. The IP addresses of the victims are compiled into a blacklist, which is subsequently disseminated as transactions to generate a ledger of the blockchain over the network. Employing this method makes it unnecessary to obstruct the victim's ports. Two scenarios, namely, single and linear, have been employed to represent intradomain topology and one scenario for interdomain in the context of multicontroller environments. The experiment investigates the effects of two attack types, single attack and multi-attacker, across three different circumstances. The findings indicate that the duration of mitigation was decreased, demonstrating the efficacy of enhancing the overall network security with increased flexibility. This approach has promise for countering DDoS attacks. This work advances by using a permissioned network with an SDN to mitigate DDOS attacks and using drop packets rather than block ports. Using HLF makes setting various configurations possible, and this act can enhance performance. Results show that mitigation time in the three topologies (single, liner, and multi-controller) was 30, 21, and 48, respectively, at the victim side, while it takes 40, 43, and 60 at the controller side.

1. INTRODUCTION

A DDoS attack is an intentional and malicious endeavor to interrupt the regular operations of a network by inundating it with an excessive volume of internet traffic. These attacks can be launched from numerous infected devices, establishing a botnet that reacts to the attacker's orders to flood the target [1]. In the context of SDN network design, the control and data planes are distinct entities in networking devices. In this architecture, the controller has full responsibility for managing all decisions within SDN switches, resulting in the elimination of decision-making capabilities from the switches. The use of centralized control in the SDN network offers significant advantages in terms of simplified management.

Conversely, a centralized controller brings about a single point of vulnerability, potentially resulting in the complete dysfunction of the entire network if it fails [2]. Nevertheless, potential attackers will perceive this as an ideal target because no single controller can keep track of everything on a big network; it is typically broken up into smaller, more manageable chunks called SDN. As illustrated in Figure 1, a separate controller has been assigned to each domain. OpenFlow, which is both a protocol between SDN controllers and switches and a specification of the logical structure of the network switch functionalities [3], enables the interaction between Controller-Switch as it would in a typical SDN situation.

The issue of a single point of failure can be addressed by utilizing a decentralized and distributed ledger within the context of blockchain technology. Each node inside the network maintains a duplicate of the database that is kept in interconnected blocks arranged in chronological order, forming a chain of blocks. Blockchain refers to a distributed database structure that facilitates the storage of transactions, requiring consensus across all network nodes regarding the transactions and their sequential arrangement [4].

To validate transactions, blockchain networks rely on consensus procedures. Consensus across several nodes improves network security by requiring agreement from most participants. While blockchain technology has numerous cybersecurity benefits, it should be noted that it is not a panacea. Security concerns should remain thorough, and best practices such as frequent security audits, network monitoring, and adherence to established cybersecurity policies should be included. Integrating blockchain with existing cybersecurity measures allows for a multi-tiered approach to developing secure and resilient systems [5].

The classification is established based on the node's capacity to access or append a novel block within the network. These platforms have similarities, but their variances may impair their security capabilities. Many permissionless blockchain platforms use the computationally demanding proof of work (PoW) consensus process, which is deemed inappropriate for high-volume transaction applications. In a permissionless platform, nodes can join networks without permission, but reading/writing ledger activities require additional privacy and control. These platforms are Bitcoin and Ethereum [6].

Permissioned blockchain platforms offer extra layers of security. The system includes an access control layer responsible for overseeing the permissions issued to authorized nodes to execute specific operations. The electronic voting system planned for the elections for the Iraq Council of Representatives is based on the concept of the permissioned HLF platform. Blockchain technology can be described as a decentralized and unalterable ledger that all participants in a given network uphold. Blockchain technology can be characterized as a distributed and immutable ledger that all participants within a network maintain. Permissioned blockchain platforms Various blockchain platforms, including Corda, Fabric, Multichain, and Quorum, have gained significant recognition in the area. HLF, in particular, is a platform that leverages open-source software and deploys permissioned distributed ledger technology (DLT) specifically designed to cater to the needs of enterprise-level applications. The platform has unique characteristics that differentiate it from other commonly utilized distributed ledger platforms [7].

This work introduces the drop-packet technique as a means of mitigating DDoS attacks. It leverages HLF to develop a blockchain application that operates on the RYU controller and utilizes the OpenFlow protocol within an SDN network. The proposed method has exhibited its ability to effectively address DDoS attacks in SDN network configurations, including single-controller, linear-controller, and multi-controller topologies. Two attack scenarios exist, namely, single and many attackers, that can be observed over three distinct topologies. The work was conducted utilizing Python programming and executed within the Mininet emulator. This paper is organized as follows: Section 2 shows the related work. Section 3 Theatrical Background of the proposed system. Section 4 proposed a mitigation system design with two attack scenarios with three topologies conducted. Finally, Section 5 reviews the results and conclusion.

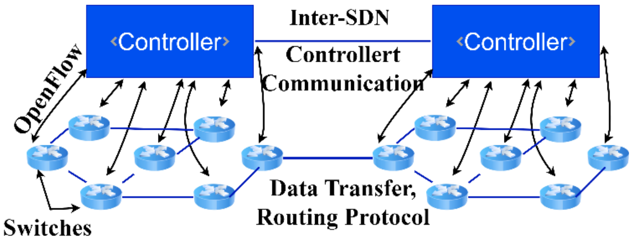


Figure 1. SDN inter domain structure [3]

2. RELATED WORK

A considerable body of literature exists about detecting and mitigating DDoS attacks in SDN, conventional networks, and blockchain technologies. These publications demonstrate a range of strategies and objectives that are achieved through different approaches. Abdulkarim et al. [8] proposed DDoS attack detection and mitigation in the SDN data plane; the approach involves developing a Python-based SDN application that leverages the identification of malicious traffic abnormalities to minimize the interference caused by normal traffic. The evaluation findings indicate that the time required for detection and mitigation falls within 100 to 150 seconds. Mohsin and Hamad [9] The suggested approach involves computing the entropy of IP addresses in the destination network traffic to identify the attack promptly. The suggested method effectively identified DDoS attacks in three SDN network topologies, namely, single, linear, and multi-controller topologies and the mitigation process-based block port is equal to 120 seconds. Al'aziz et al. [10] recommended using the SNORT IPS blockchain to distribute blacklisted IP lists. The smart contract stores the attack source or banned IP information. The program sends and retrieves attack sources or blacklisted IP information from the smart contract. A blacklisted IP's expiration time can be set. The overall number of attacking packets was lowered from an average of 115,578 to 27,165. Hajizadeh et al. [11] introduced a secure platform that channels and verifies member access to private information. Smart contract-based security partnerships need to achieve trustworthiness. An SDN control plane that enforces security policies quickly, reducing cyberattack mitigation time. Shafi and Basit [12] presented an IoT software-defined DDoS botnet avoidance technique. Distributed ledger updates occur continuously across the network. It may instantly download flow rules throughout the SDN controller blockchain network and look for suspicious behaviour or traffic on an innocent network. It detects DDoS botnets and targets traffic. Changes to the system data plane, topological properties, and flow mode communication can indicate malicious updates. It also uses blockchain features to stop Internet of Things devices from being used as slaves in botnets. Hayat et al. [13] presented a blockchain-based multilevel DDoS mitigation strategy for IoT devices, a blockchain-based device verification technique to exclude illegal IoT devices. Three benchmark apps were used to assess the performance of the suggested framework, which was created using the blockchain benchmark tool Hyperledger Caliper. The results demonstrate that the suggested framework delivers up to 35% throughput improvement, up to 40% latency improvement, and up to 25% better CPU utilization. El Houda et al. [14] proposed that the framework, Co-IoT, is a blockchain-based solution designed to mitigate collaborative DDoS attacks. It leverages smart contracts, specifically Ethereum's smart contracts, to enable the coordination of attack mitigation efforts among SDN-based domains and the secure and efficient transfer of decentralized attack information. The implementation of Co-IoT is deployed on the Ropsten test network, the official test network for Ethereum. The experimental findings validate that Co-IoT demonstrates attributes of flexibility, efficiency, security, and cost-effectiveness, hence establishing it as a promising approach for mitigating large-scale DDoS attacks.

Bitanet et al. [15] The proposal entails an incentive-driven decentralized DDoS protection system, which effectively enables operators to counteract attacks from sources nearby. This approach leverages existing network devices for mitigation purposes. The utilization of blockchain technology and smart contracts enables operators to have complete visibility of all accounts and ensures accurate payment processing. The demonstration tool computes the proportion of filtered attack traffic by leveraging operator involvement. It demonstrates that even with a 50% participation rate, one can achieve a filtration rate of 89.2% and a false positive rate of 0%. Wang et al. [16] A novel approach is suggested for implementing a DDoS blacklist mechanism within the context of an IPv6-SAVI network. This proposal involves utilizing a smart contract to facilitate the functionality of the mechanism. In the SAVI environment, the credibility of DDoS source information discovered by the Intrusion Detection System (IDS) is acknowledged. This study proposes a dynamic update technique for managing the reputation of trustworthy addresses based on the detection findings and the subsequent formation of a blacklist. Finally, extensive tests measure the proposed mechanism's performance in terms of latency, overhead, reputation-changing accuracy, and so on. demonstrating that the blacklist may serve as a DDoS traffic filtering reference to increase DDoS mitigation capabilities.

The study by El Houda et al. [14] is strongly aligned with our research as it incorporates the utilization of Ethereum's smart contracts within the context of SDN-based systems. Our research employs the utilization of Hyperledger Fabric's smart contract on a Software-Defined Networking (SDN) network, addressing all issues identified in earlier studies of a similar nature.

3. HLF PLATFORM

A blockchain refers to a decentralized ledger system in which encrypted entries, known as "blocks," are appended. The blockchain refers to a decentralized and unchangeable database ledger that securely saves transactions within blocks, which are connected through cryptographic hashes and organized chronologically. Blockchain technology operates as a decentralized peer-to-peer (P2P) network wherein individuals are uniquely identified by using private and public keys [17, 18]. The first block in a blockchain, commonly known as the "genesis block," lacks a corresponding hash value for any prior block. Verbalizing all blocks can be accomplished by tracing each block back to the Genesis block. A cryptographic link is established between each subsequent block inside the chain and its preceding block. After recording transaction information and data in a block, any alteration to this information would need the revision of all preceding blocks. Establishing a connection to the blockchain network and subsequently transmitting transactions to it is a viable undertaking. The Hyperledger project encompasses multiple open-source subprojects: Iroha, Sawtooth, Fabric, Indy, and Burro.

HLF is a widely adopted open-source distributed ledger platform on a permissioned blockchain network [19]. HLF does not incorporate any form of cryptocurrency similar to that found in Bitcoin and Ethereum. The network access to this platform is restricted solely to individuals who are network members. The ledger embedded within the fabric consists of two distinct elements: a global state and a blockchain. The global state refers to a centralized storage system responsible for preserving ledger states' current values.

Furthermore, fabric can effectively manage a versioned key-value store while offering support for state databases like CouchDB and LevelDB [20]. The present study used CouchDB as its chosen database management system. The version number of each key is incremented upon being written. A blockchain is composed of distinct units of transactions, which can be classified as either successful or failed. HLF facilitates the development of smart contracts using programming languages such as Go, Java, and Node.js.

A chaincode is a specific form of the smart contract found within the fabric framework. It comprises the comprehensive definition of all functions invoked by a transaction. Endorsement policies are associated with chaincodes and apply to interconnected smart contracts. A channel can ensure the confidentiality of transactions conducted among participants inside a network by implementing a privacy-preserving mechanism. Each channel is responsible for upholding its ledger, ensuring that the member nodes within that channel are the sole entities capable of accessing transactions and data. A fabric network comprises several entities associated with separate organizations, such as peer nodes, orderer nodes, and clients. The Membership Service Provider (MSP) is responsible for assigning a unique identification to any entity present within the network. Certificate Authorities (CAs) generate identities by establishing a cryptographic key pair comprising a public key and a private key, which can be employed for confirming identification [21].

The execution of transactions and the maintenance of ledgers are the primary responsibilities of peer nodes. The nodes designated as orderers in a network are responsible for initiating the proposal of new blocks and participating in a consensus process to arrange all transactions efficiently. HLF comprises a trichotomy of ordering services. The three literary pieces being examined are "Solo," "Kafka," and "Raft." [22]. All peer nodes assume the role of committers, wherein they receive state updates ordered via the ordering service in the context of a transaction block. Moreover, these nodes are incumbent upon maintaining the ledger's integrity. Upon receiving a new block, the peer node validates transactions, alterations their local ledgers, and subsequently appends the block to the blockchain. In a distributed network, peer nodes can serve as endorsers by offering transaction recommendations. The customer simultaneously submits the transaction request to multiple peers to obtain endorsements for enhanced security. The transaction is documented and communicated to the party placing the order for incorporation into a block. Subsequently, the data is disseminated across all individuals inside the network to verify its accuracy and ensure its steadfastness [23].

The user has provided a numerical reference. Execute transactions by querying and invoking the function with the specified arguments. The process may entail retrieving and modifying data inside the state database, resulting in either a successful or unsuccessful outcome. The inquiry transaction executes the designated function and retrieves the status of the peer as a result. The modification of the distributed ledger is solely carried out by the transactions that have been triggered. In order to achieve successful transactions, it is important to execute, order, and validate.

Figure 2 depicts the transaction flow within the HLF platform. The client application utilizes the fabric SDK to generate a transaction proposal for executing the chaincode function. This function is responsible for both reading and writing data to the ledger. The proposal is received by one or more endorsed peers after its signing by the client using their respective credentials. Then, peers verify endorsement details such as transaction format, duplicate, issuer signature, etc. Client transactions are ordered by the ordering service using a consensus protocol. A transaction block is established when the following three requirements are satisfied: block timeout, block size, or block max bytes. The concluding phase involves the process of validation, wherein each peer verifies the orderer's signature on the block. Initially, the orderer's signatures are verified by peers. Subsequently, the signatures are decoded, and the endorsement policy is assessed by employing the validation system chaincode (VSCC). This evaluation aims to ascertain whether an adequate number of endorsed peer signatures are present. Ultimately, the peer assesses the key version by employing the Multisession Chaincode. If both VSCC and MVCC validation checks are successful, the write sets will be written to the world state. However, if any of the validation checks fail, the validation process will be considered unsuccessful. Clients who have subscribed to the service can receive notifications from their peers regarding committed events [24].

Transactions in HLF are subject to endorsement policies, establishing the criteria for a transaction to be declared valid. The network may withstand DDoS attacks by enforcing tight endorsement policies. The HLF pluggable consensus process enables the adoption of DDoS-resistant algorithms. Practical Byzantine Fault Tolerance (PBFT) and Raft consensus techniques, for example, can give improved resistance to malevolent nodes seeking to disrupt the consensus process [25]. The marriage of technologies such as blockchain and SDN can improve network security, truthfulness, and efficiency. A decentralized threat intelligence platform can be built using blockchain. SDN nodes can communicate information about ongoing attacks, and consensus procedures can aid in the detection and mitigation of DDoS attacks [26].

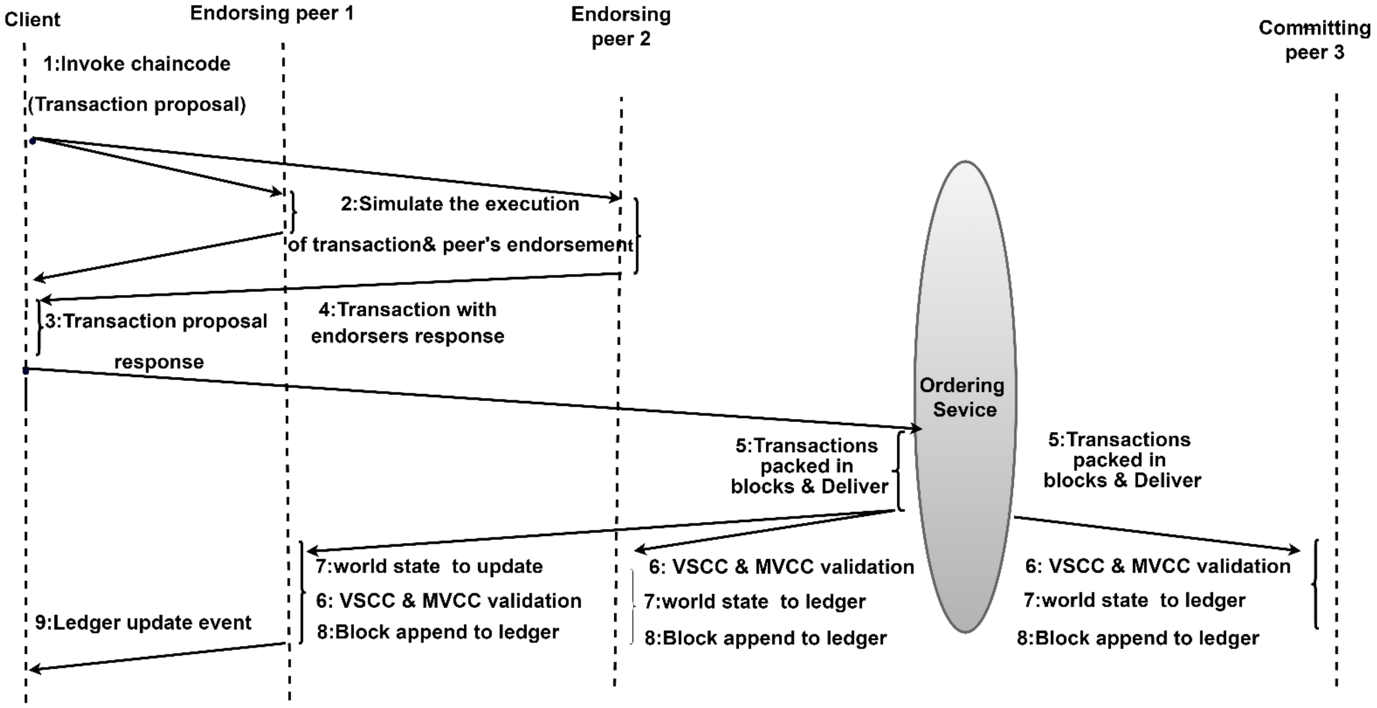


Figure 2. Transaction flow of Hyperledger Fabric [19]

4. DDOS ATTACK IN SDN

SDN is a network design that offers a potential solution to the challenges of managing large-scale networks. The system's architectural design effectively separates the control plane from the data plane, enabling adaptable and programmable network administration. The process of decoupling in this context involves the introduction of a network object referred to as the controller. This controller is capable of dynamically modifying the rules within the switch flow table through the utilization of the OpenFlow protocol. The OpenFlow protocol is a communication interface facilitating interaction between the controller and switches. The centralized controller in SDN is a good target for attackers, but the SDN can provide a good defense mechanism to eliminate any attack traffic. When attack traffic is detected, the controller can decide to drop the flow by modifying the switch flow table rules [27].

One of the prevalent threats to network security, which poses a significant risk to SDN, is the DDoS attack. This type of attack involves the deliberate inundation of a target system with a high volume of traffic originating from numerous infected servers, sometimes referred to as zombies or botnets (as depicted in Figure 3). A DDoS attack is a deliberately endeavor to exhaust the resources of a network or host by inundating it with an excessive volume of malicious packets, a technique known as flooding. The objective of such an attack is to render the targeted host, the victim, incapable of providing its intended services [28]. Therefore, SDN brings many benefits to network security through dynamic flow management that can be depended on for early detection of DDoS attacks and reduced resource wasting. Moreover, SDN makes it possible to add intelligent detection and mitigation algorithms [29].

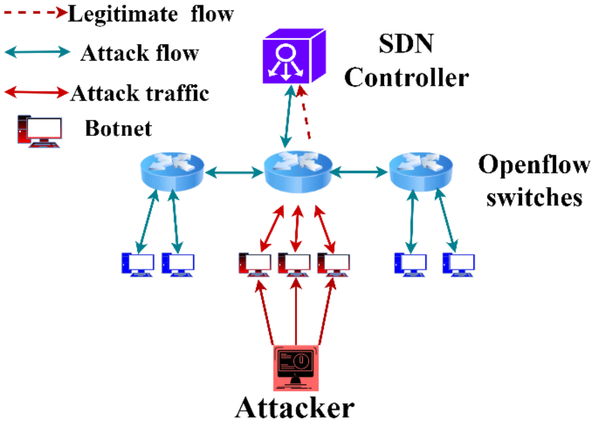


Figure 3. DDoS attack in SDN [24]

5. MULTIPLE ATTACK SCENARIOS

The system under consideration was tested using the Mininet emulator on the Linux platform, Ubuntu 20.04 LTS. The programming language employed for this purpose was Python 3.10.7. The study established three distinct network topologies and examined two attack scenarios to assess the impact of altering the topology on the effectiveness of detection and mitigation measures. Three autonomous systems, inter- and intra-domain, are utilized for their implementation. The single topology configuration comprises a singular controller, a solitary switch, and eight hosts. The linear topology configuration comprises a single controller, three switches, and 24 hosts. The system, referred to as "linear with multiple controllers," comprises two controllers, four switches, and 32 host topologies. The multi-controller architecture comprises three controllers, three switches, and 24 hosts. Table 1 presents the temporal durations of several attackers over three distinct topologies. The suggested system includes three distinct topologies: single controller, linear controller, and multi-controller. Each topology is depicted in Figure 4. Both single-attacker and multiple-attacker attack scenarios are considered for each topology.

The first scenario starts by running an attack script (python code) on one host to attack the victim in the topology; this script runs as a DDoS attack by sending packets from spoofed IP addresses to the target. In the second scenario, the attack script runs on four hosts; first, normal traffic is launched for 10 seconds, then the attack traffic is run one by one on four hosts after 10 seconds to attack the target. Single topology consisting of 8 hosts, one switch, and one controller, a single attack scenario started by running the attack script on host 1 to attack host 3. There was also normal traffic running on host 1. Controller C0 detects the attack and starts mitigation by dropping the packet sent to host 3.

Hosts 1, 2, 3, and 4 sequentially attack host8 in multiple attacks. The controller detects and mitigates attacks and removes the effect of DDoS attacks.

The linear topology contains two controllers and two switches with 16 hosts divided between two domains; the single attack started from h4 toward h11 and h1 toward h24; the controller of the victim domain first detects the attack and mitigates it; all controllers detect attacks and start mitigation. Also, multiple attacks are implemented, and the number of hosts is increased to 24 in topology using one controller and three switches to get different results; four hosts start attacks (h1, h2, h3, and h4 sequential attack h24). C0 can detect and mitigate attacks**.** Multi-controller topology contains three domains, each consisting of one controller, one switch, and eight hosts: host four attacks host 11 and host one attacks host 24; the controller of the victim detects an attack, and all controllers start to drop packets to stop attacks toward victims. In another scenario, the multi-controller topology changed by decreasing the number of controllers to two controllers and increasing switches and hosts (four switches and 32 hosts), so the result changed.

Table 1. Time and traffic type for topologies in multiple attacker scenarios

No.	**Type of Traffic**	**Time**
1	Normal traffic	0-10s
2	DDoS Attacker (host1)	10-20s
3	DDoS Attacker (host2)	20-30s
4	DDoS Attacker (host3)	30-40s
5	DDoS Attacker (host4)	40-60s

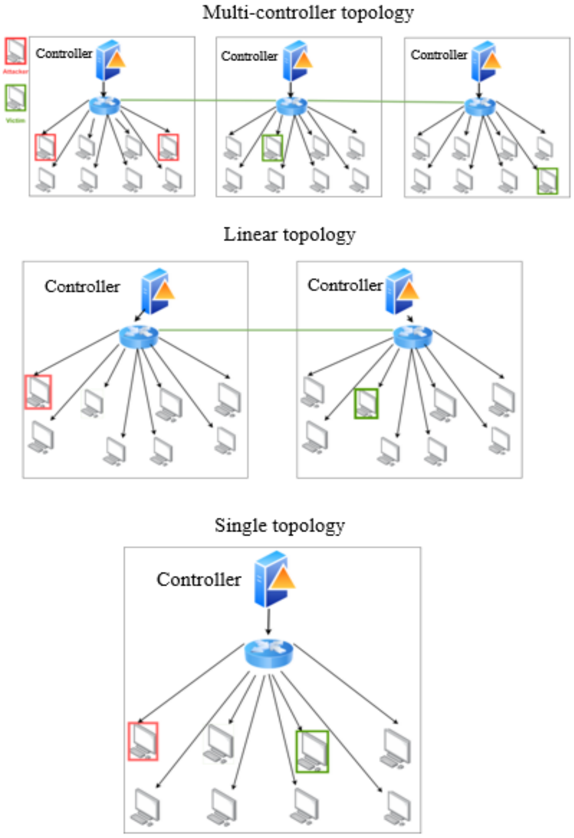


Figure 4. Single, linear, and multicontroller topologies in a single attacker scenario

6. MITIGATION PROCESSING USING HLF

Figure 5 is a high-level diagram of the proposed system. The autonomous system is divided into three network domains: the source, intermediate, and destination domains. In this proposed system, once the detection-based entropy is activated and the target host (victim) has been identified, the defense process added to the controller will be activated. The source network domain is the network in which attackers start DDoS attacks. The destination network domain is the domain where the victim is hosted.

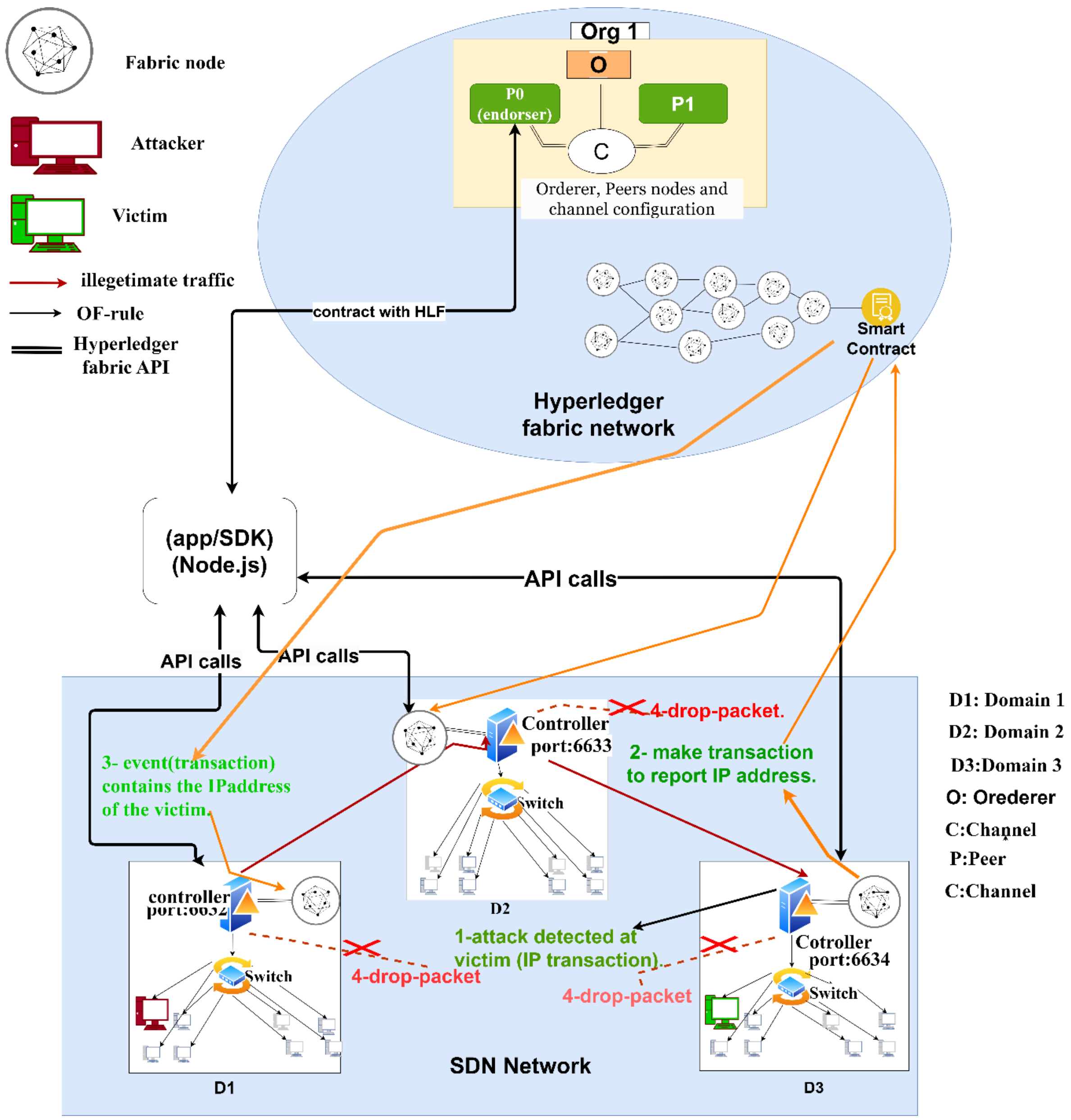


Figure 5. High-level architecture of the proposed system

The HLF network configuration includes one organization with two peers and one channel; peer0 is an endorser. The testing and evaluation, particularly in a development or testing environment, can aid in improving configurations prior to deploying a HLF network in a production environment.

The mitigation technique can be explained as follows:

Step 1: Upon detection of the attack by the control plane (the SDN controller) within the victim's domain, measures are taken to mitigate distributed denial-of-service (DDoS) attacks occurring within such domains.

Step 2: In the specified domain, the SDN controller's fabric node initiates a transaction to the smart contract of HLF to report any IP address deemed suspect, provided that the transaction has been duly allowed.

Step 3: A smart contract automatically emits an event after confirming a transaction within the HLF network. Authorized collaborators of the cooperation contract, such as the SDN controller of the source and destination network domains, then receive this event.

Step 4: At the end, the authorized collaborators drop a packet sent to the victim.

The detection procedure begins with the packet counting function, extracting the destination IP from the packets-per-second statistic and comparing this value to the threshold value for the normal packet rate that is determined by observing the behavior of the network under normal conditions. Then mitigation started, and the IP of the victim (destination IP) was reported by the first controller that detected the attack to the HLF network (make transactions to HLF contain the IP of the victim), and the process of adding the IP to the HLF network took ~ 60 seconds. After appending the victim's IP to the HLF network, the drop packet started, and a new flow rule was pushed from the controller into the switch's flow table to drop similar malicious flows.

The mitigation process started to get the destination IP from the packet per second statistics and checked against the normal packet rate threshold calculated during the normal network behavior. The IP of the victim (destination IP) is reported by the first controller that detects an attack on the HLF network (makes the transaction to the blockchain contain the IP of the victim), and the process of adding the IP to the blockchain network (10-60 sec). Then, the drop packet starts when the victim IP becomes into the HLD network. Fabric node refers to a participant in the blockchain network that maintains a copy of the ledger and executes transactions. The controller pushes the new flow rule into the switch's flow table to drop similar malicious flows. By dropping packets, the system can mitigate the impact of DDoS attacks on the network. Implementing packet dropping can effectively mitigate the impact of DDoS attacks on the network. The detection and mitigation process of the suggested system is illustrated in Figure 6.

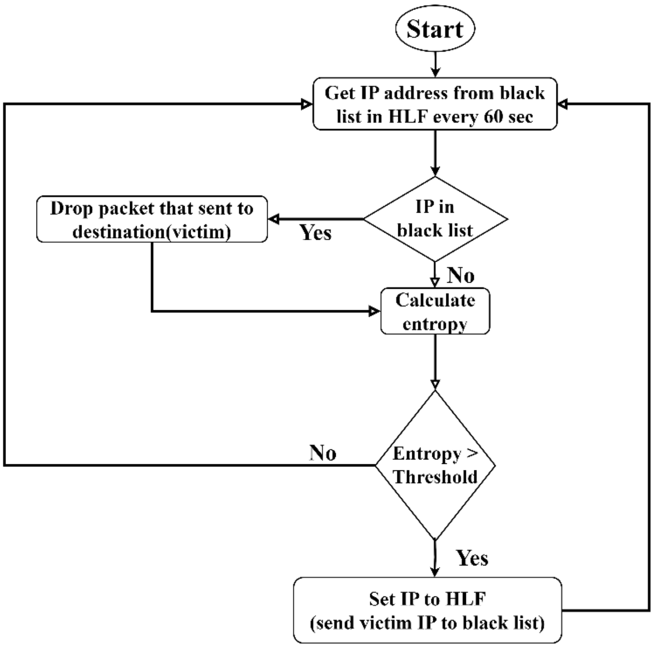

Start
Get IP address from black list in HLF every 60 sec
IP in black list
Yes
Drop packet that sent to destination(victim)
No
Calculate entropy
Entropy > Threshold
No
Yes
Set IP to HLF (send victim IP to black list)

Figure 6. Detection and mitigation flowchart of proposed system

The present work has employed entropy-based attack detection. The suggested approach can be elucidated through a two-step process.

(1) The calculation of entropy following the generation of normal and attack traffic.

(2) The determination of the appropriate threshold value for each topology, including single, linear, and multi-controller, is computed.

The traffic is created using the scapy library in the Python programming language. During all experimental trials, two instances of scapy programs are executed simultaneously. One method involves creating regular network traffic, while the second method involves launching a malicious attack that floods the network with packets at a higher pace than typical traffic. The initiation of regular network traffic occurs from a certain source host, whereas the commencement of malicious network traffic originates from any host with falsified IP addresses. This malicious traffic is directed towards the target host, resulting in the creation of a table miss entry within the switch. The attack rate can be further amplified by executing the attack script repeatedly on numerous hosts.

The calculation of the cumulative entropy for a window size of 50 packets can be determined by employing Eqs. (1) and (2). In order to obtain the entropy values, the network was executed during the testing phase. Initially, regular traffic is initiated in order to gather entropy values during a typical network activity and document the highest and lowest recorded values. Furthermore, the attack traffic is initiated in order to gather the entropy value during the attack, afterwards documenting both the minimum and maximum values of this entropy. In order to achieve accurate attack detection, it is necessary to select an appropriate threshold value after conducting various tests on the proposed topology. This is based on the criterion that if the entropy value is lower than the threshold for five consecutive instances, it is classified as an attack. In order to ascertain the suitable threshold, a series of tests were conducted to analyze the impact of attacks on entropy across various topology types and attack rates. The management of attack velocity is achieved through the utilization of many hosts to execute the attack.

A series of four attackers hosts were systematically deployed to target a sole target with the purpose of observing and contrasting the impact of escalating DDoS attack intensity on the entropy value. In the event that a single host initiates a UDP flood attack, the attack can be classified as a low-rate attack due to the injection of a maximum of 14 additional packets per second. Consequently, multiple attack traffic rates are employed, and the entropy value is carefully checked at each rate. The threshold can be determined by analyzing the minimum entropy value observed in normal traffic and the maximum entropy value observed in attack traffic acquired during the preceding phase. Figure 7 show the Entropy variation and threshold selection during normal and attack traffic. Eq. (3) is employed for the computation of the threshold value, which serves as the boundary for the detection of attacks. Eq. (3) demonstrates the process of selecting the threshold in the three different topologies. Table 2 displays the threshold selection process based on the entropy value within three distinct topologies.

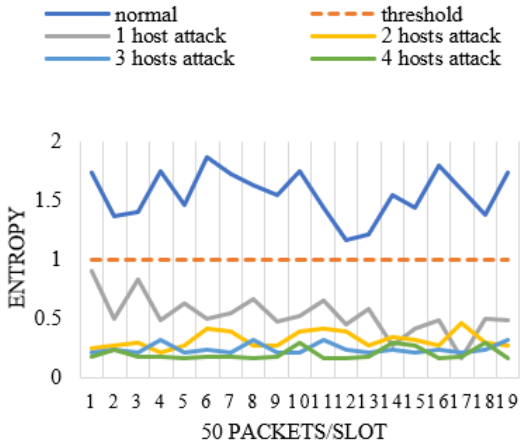
$$P(Y) = \frac{\text{Number of packets with Y dest.IP address}}{\text{Total No.of packets}} \quad (1)$$

$$\text{H(Y)} = \text{P(Y)}.\log_2 \frac{1}{\text{P(Y)}} \quad (2)$$

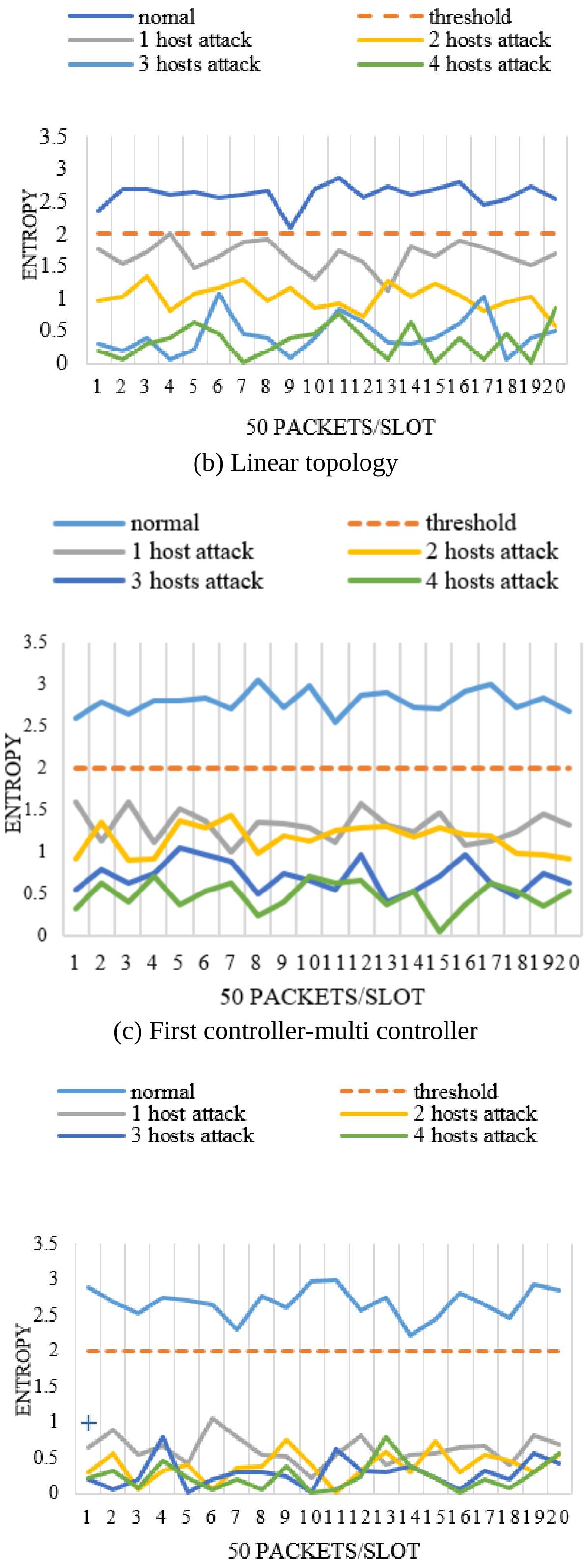
$$\text{Threshold} = \frac{\text{Min.Normal entropy+Max.attack entropy}}{2} \quad (3)$$

Table 2. Threshold selection depending on entropy value

Topology	Normal Min Entropy	Normal Max Entropy	Threshold
Single	1.16	0.9	1
Linear	2.1	1.92	2
Multi-controller	2.55	1.6	2



(a) Single topology



(b) Linear topology

(c) First controller-multi controller

(d) Second controller-multi controller

Figure 7. Entropy variation and threshold selection during normal and attack traffic

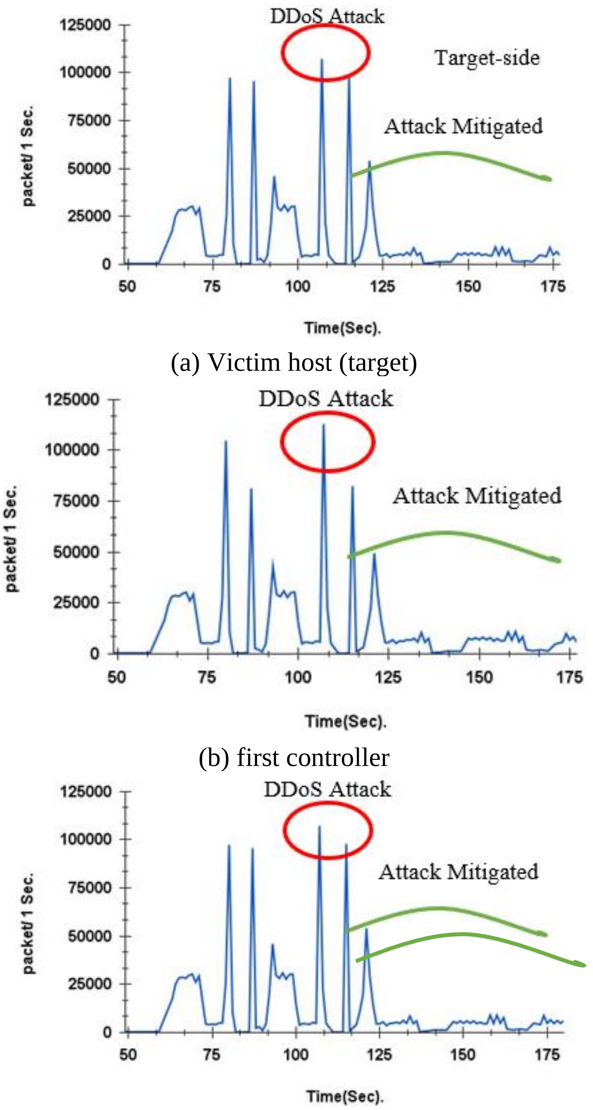
7. RESULT

In the following section a result and discussion for all topologies used in this work with both single and multiple attacks.

7.1 Single attack scenario

Results show that the mitigation process takes 30 sec. to drop packet of the attack in single topology at target and controller side 40 sec. The packet rate reached 8500 packets/sec in the target and 90,000 in the controller.

Figure 8 a, b and c shows the mitigation results for the linear topology at target side and both controller within the network where the attack starts from the first domain (first controller) to other domain (second controller). The mitigation time in the target was 21 sec while 43 in both controllers. It can be seen that when stopping the attack after the mitigation process starts, the DDoS attack is removed completely from the SDN topology and never starts again due to the use of HLF, which is distributed among all hosts within the network, and they will drop any packet targeting the IP address of the target victim. Figure 9 shows the mitigation results in multi-controller topology scheme. Three controller where used in this topology C0, C1 and C2. Two host attacks two victim h11 and h24. Figure 9 a and b show the mitigation time in target h11 was 10 sec and 50 sec for target h24. The difference in time is due to the difference of victim located within the domains. Figure 9 c,d,e show the mitigation time in the three controllers which was 50 sec.

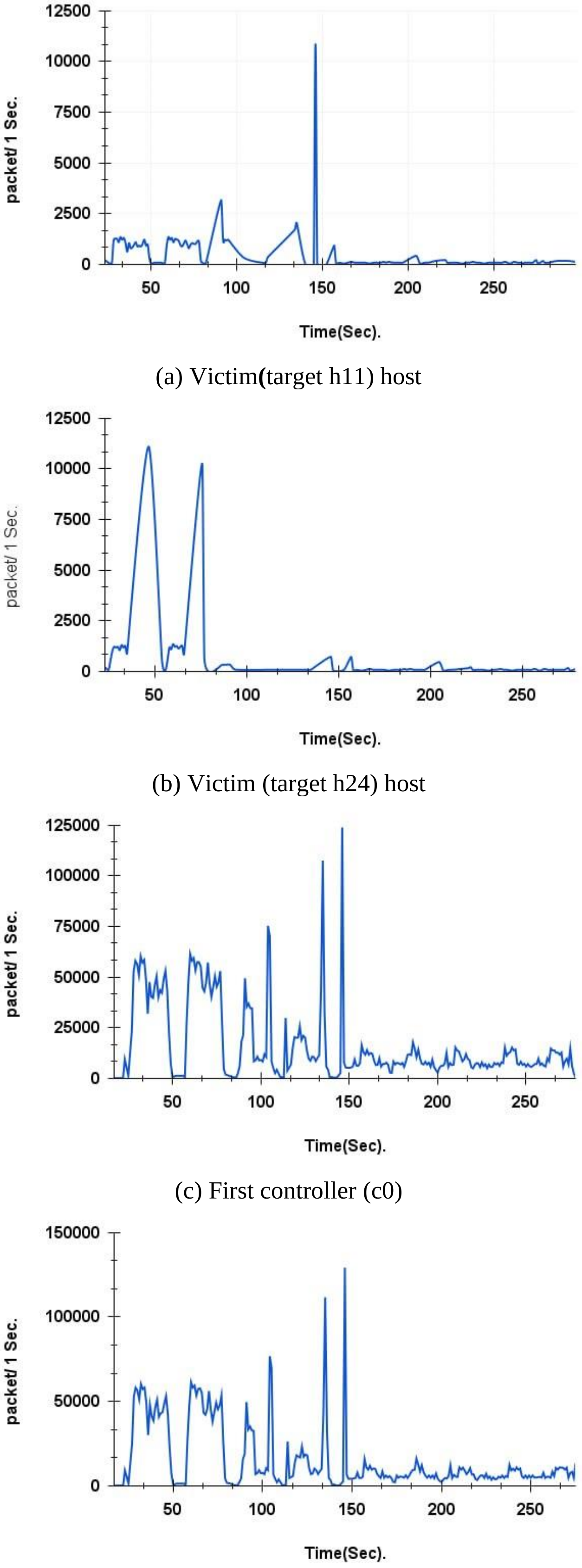


(a) Victim host (target)

(b) first controller

(c) second controller

Figure 8. Mitigation result of linear topology

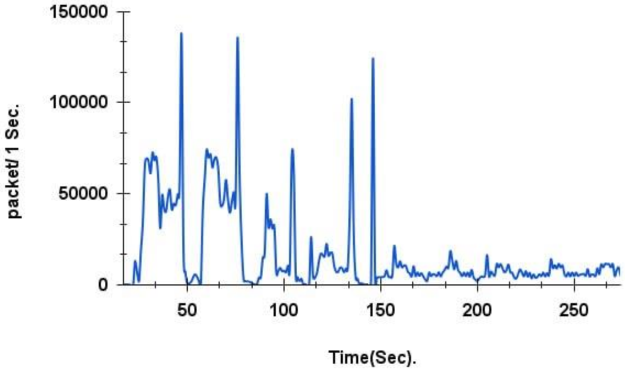


(a) Victim**(**target h11) host

(b) Victim (target h24) host

(c) First controller (c0)

(d) Second controller (c1)

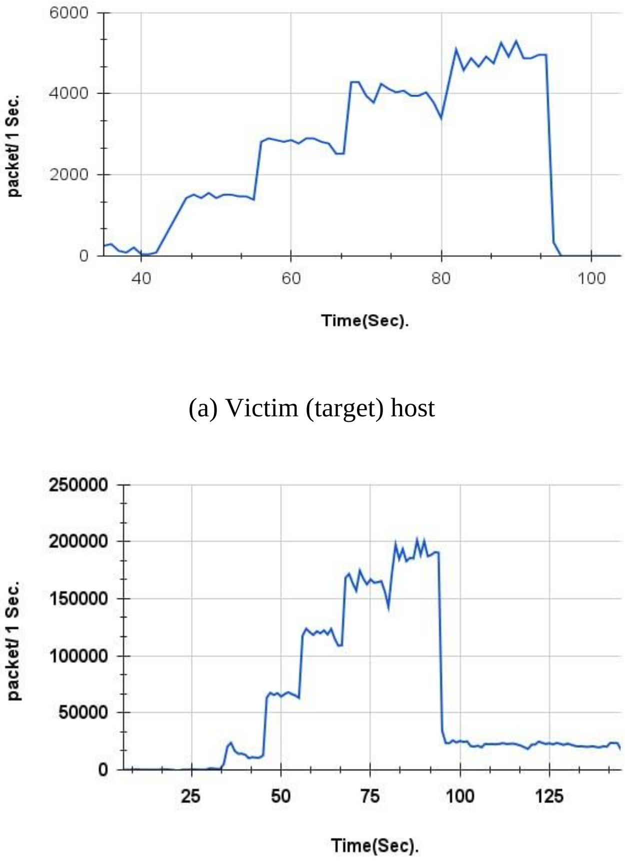


(e) Third controller (c2)

Figure 9. Mitigation result of multi-controller topology

7.2 Multi attack scenario

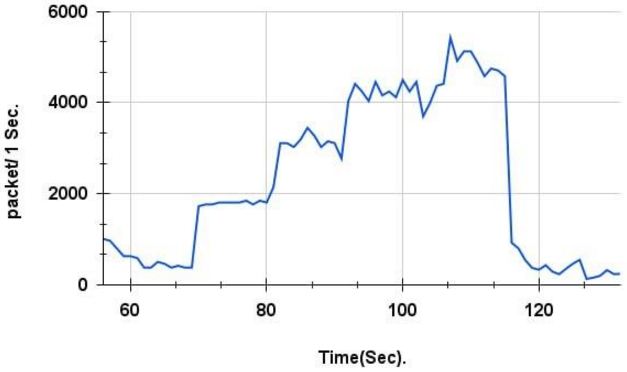
For linear topology the mitigation time was 48 sec and 60 sec at the controllers side as shown in Figure 10 a and b. Figure 11 a shows the mitigation time in target side of the multi-controller topology which was 45 sec while Figure 11 b shows the mitigation time in all controllers with time 45 sec also.



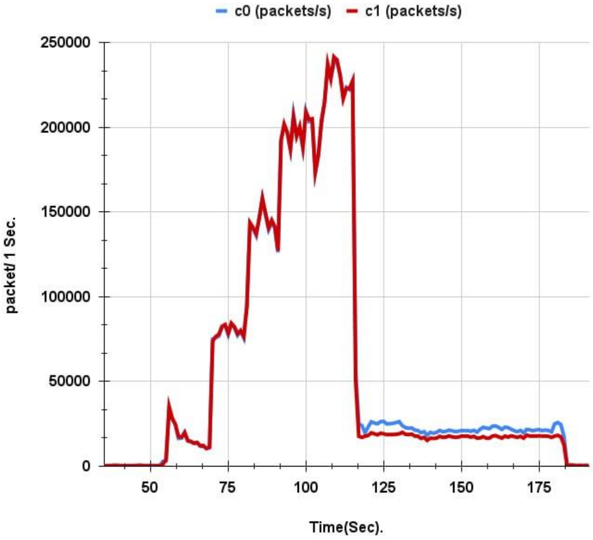
(a) Victim (target) host

(b) Controller

Figure 10. Mitigation result of linear topology



(a) Victim (target) host



(b) first &second controller

Figure 11. Mitigation result of multicontroller topology (a), (b) first & second controller

8. CONCLUSIONS

In this work, we proposed a mitigation technique against DDoS attacks in SDN using HLF blockchain technology. Using Blockchain overcomes the problem of the centralized nature of SDN by using a distributed ledger over the hosts of the network. The proposed mitigation technique shows significant flexibility where the victim does not need to block its port for a long time by eliminating suspicious packets from its source. Compared to El Houda et al. [14] using a permissioned platform rather than a permissionless one, this work also studied three scenarios, single, linear, and multi-controller, with the single attacker and multi attacker.

The developing technology of Blockchain offers a robust solution for cost-effective, optimized, and adaptable mitigation of inter and intra-domain SDN against DDoS attacks. This work, compared to Mohsin and Hamad [9], the mitigation strategy, employs the IP addresses of the victims to be compiled into a blocklist, which is subsequently disseminated as transactions to generate a ledger of the Blockchain over the network. By doing so, it becomes unnecessary to obstruct the victim's ports.

This work can be developed by exploring other attacks to implement an intrusion detection system IDS mitigation using Blockchain, testing and evaluating new blockchain platforms, and increasing the number of nodes.

The gap in such a security mechanism system using blockchain technology is the high computation and resources required to implement such an algorithm. Implementing the proposed system requires more evaluation and usage of different blockchain platforms.

REFERENCES

[1] Ahalawat, A., Dash, S.S., Panda, A., Babu, K.S. (2019). Entropy based DDoS detection and mitigation in OpenFlow enabled SDN. In 2019 International Conference on Vision Towards Emerging Trends in Communication and Networking (ViTECoN), Vellore, India, pp. 1-5. https://doi.org/10.1109/ViTECoN.2019.8899721

[2] Abdullah, M., Al-awad, N., Hussein, F. (2019). Implementation of entropy-based distributed denial of service attack detection method in multiple POX controllers. Review of Computer Engineering Studies, 6(2): 29-38. https://doi.org/10.18280/rces.060201

[3] Vijayan, P., Manju, R. (2017). Network resource management and control in inter-domain SDN. In 2017 International conference of Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, India, pp. 76-81. https://doi.org/10.1109/ICECA.2017.8212768

[4] Abou El Houda, Z., Hafid, A.S., Khoukhi, L. (2023). Mitfed: A privacy preserving collaborative network attack mitigation framework based on federated learning using sdn and blockchain. IEEE Transactions on Network Science and Engineering, 10(4): 1985-2001. https://doi.org/10.1109/TNSE.2023.3237367

[5] Gimenez-Aguilar, M., De Fuentes, J.M., Gonzalez-Manzano, L., Arroyo, D. (2021). Achieving cybersecurity in blockchain-based systems: A survey. Future Generation Computer Systems, 124: 91-118. https://doi.org/10.1016/j.future.2021.05.007

[6] Sey, C., Lei, H., Qian, W., Li, X., Fiasam, L.D., Kodjiku, S.L., Adjei-Mensah, I., Agyemang, I.O. (2022). VBlock: A blockchain-based tamper-proofing data protection model for internet of vehicle networks. Sensors, 22(20): 8083. https://doi.org/10.3390/s22208083

[7] Saeed, S.H., Hadi, S.M., Hamad, A.H. (2022). Iraqi paradigm E-voting system based on Hyperledger Fabric blockchain platform. Ingénierie des Systèmes d'Information, 27(5): 737-745. https://doi.org/10.18280/isi.270506

[8] Abdulkarem, H.S., Dawod, A. (2020). DDoS attack detection and mitigation at SDN data plane layer. In 2020 2nd Global Power, Energy and Communication Conference (GPECOM), Izmir, Turkey, pp. 322-326. https://doi.org/10.1109/GPECOM49333.2020.9247850

[9] Mohsin, M.A., Hamad, A.H. (2022). Implementation of entropy-based DDoS attack detection method in different SDN topologies. American Academic Scientific Research Journal for Engineering, Technology, and Sciences, 86(1): 63-76.

[10] Al'aziz, B.A.A., Sukarno, P., Wardana, A.A. (2020). Blacklisted IP distribution system to handle DDoS attacks on IPS Snort based on blockchain. In 2020 6th Information Technology International Seminar (ITIS), Surabaya, Indonesia, pp. 41-45.

https://doi.org/10.1109/ITIS50118.2020.9320996

[11] Hajizadeh, M., Afraz, N., Ruffini, M., Bauschert, T. (2020). Collaborative cyber attack defense in SDN networks using blockchain technology. In 2020 6th IEEE Conference on Network Softwarization (NetSoft), Ghent, Belgium, pp. 487-492. https://doi.org/10.1109/NetSoft48620.2020.9165396

[12] Shafi, Q., Basit, A. (2019). DDoS botnet prevention using blockchain in software defined internet of things. In 2019 16th international Bhurban conference on applied sciences and technology (IBCAST), Islamabad, Pakistan, pp. 624-628. https://doi.org/10.1109/IBCAST.2019.8667147

[13] Hayat, R.F., Aurangzeb, S., Aleem, M., Srivastava, G., Lin, J.C.W. (2022). ML-DDoS: A blockchain-based multilevel DDoS mitigation mechanism for IoT environments. IEEE Transactions on Engineering Management, 1-14. https://doi.org/10.1109/TEM.2022.3170519

[14] El Houda, Z.A., Hafid, A., Khoukhi, L. (2019). Co-IoT: A collaborative DDoS mitigation scheme in IoT environment based on blockchain using SDN. In 2019 IEEE Global Communications Conference (GLOBECOM), Waikoloa, HI, USA, pp. 1-6. https://doi.org/10.1109/GLOBECOM38437.2019.9013542

[15] Bitan, S., Molkho, A., Dankner, A. (2023). Decentralized incentive-based DDoS mitigation. In 2023 26th Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN), Paris, France, pp. 33-35. https://doi.org/10.1109/ICIN56760.2023.10073483

[16] Wang, Y., Wang, X., Zeng, R., Huang, M. (2022). A CIDS mode DDoS blacklist mechanism based on smart contract in SAVI-enable IPv6 network. In 2022 IEEE 22nd International Conference on Communication Technology (ICCT), Nanjing, China, pp. 1363-1369. https://doi.org/10.1109/ICCT56141.2022.10072937

[17] Jafar, Z., Hamad, A.H. (2023). Performance evaluation of a multi organizations secure internet of vehicles based on Hyperledger Fabric blockchain platform. Ingénierie des Systèmes d'Information, 28(3): 703-709. https://doi.org/10.18280/isi.280320

[18] Ibraheem, S.S., Hamad, A.H., Jalal, A.S.A. (2018). A secure messaging for internet of things protocol based RSA and DNA computing for video surveillance system. In 2018 Third Scientific Conference of Electrical Engineering (SCEE), Baghdad, Iraq, pp. 280-284. https://doi.org/10.1109/SCEE.2018.8684055

[19] Kadhum, O.I., Hamad, A.H. (2023). Performance evaluation of multi-organization e-government based on Hyperledger Fabric blockchain platform. Ingénierie des Systèmes d'Information, 28(2): 499-507. https://doi.org/10.18280/isi.280227

[20] Saeed, S.H., Hadi, S.M., Hamad, A.H. (2022). Performance evaluation of E-voting based on Hyperledger Fabric blockchain platform. Revue d'Intelligence Artificielle, 36(4): 581-587. https://doi.org/10.18280/ria.360410

[21] The Linux Foundation Project. https://www.hyperledger.org, accessed on 10 June 2022.

[22] Hyperledger white paper working group. (2018). Hyperledger Blockchain Performance Metrics. San Francisco, CA, USA: Linux Foundation, pp. 1-17. https://hyperledger.org/.

[23] Casino, F., Dasaklis, T.K., Patsakis, C. (2019). A systematic literature review of blockchain-based applications: Current status, classification and open issues. Telematics and Informatics, 36: 55-81. https://doi.org/10.1016/j.tele.2018.11.006

[24] Farahani, B., Firouzi, F., Luecking, M. (2021). The convergence of IoT and distributed ledger technologies (DLT): Opportunities, challenges, and solutions. Journal of Network and Computer Applications, 177: 102936. https://doi.org/10.1016/j.jnca.2020.102936

[25] Onireti, O., Zhang, L., Imran, M.A. (2019). On the viable area of wireless practical byzantine fault tolerance (pbft) blockchain networks. In 2019 IEEE Global Communications Conference (GLOBECOM), Waikoloa, HI, USA, pp. 1-6. https://doi.org/10.1109/GLOBECOM38437.2019.9013778

[26] Rahman, A., Montieri, A., Kundu, D., Karim, M.R., Islam, M.J., Umme, S., Nascita, A., Pescapé, A. (2022). On the integration of blockchain and sdn: Overview, applications, and future perspectives. Journal of Network and Systems Management, 30(4): 73. https://doi.org/10.1007/s10922-022-09682-4

[27] Mohsin, M.A., Hamad, A.H. (2022). Performance evaluation of SDN DDoS attack detection and mitigation based random forest and K-nearest neighbors machine learning algorithms. Revue d'Intelligence Artificielle, 36(2): 233-240. https://doi.org/10.18280/ria.360207

[28] AbdelAzim, N.M., Fahmy, S.F., Sobh, M.A., Eldin, A.M.B. (2021). A hybrid entropy-based DoS attacks detection system for software defined networks (SDN): A proposed trust mechanism. Egyptian Informatics Journal, 22(1): 85-90. https://doi.org/10.1016/j.eij.2020.04.005

[29] Ali, J., Lee, S., Roh, B.H. (2018). Performance analysis of POX and Ryu with different SDN topologies. In Proceedings of the 1st International Conference on Information Science and Systems, pp. 244-249. https://doi.org/10.1145/3209914.3209931