MAJOR PROJECT REPORT

On

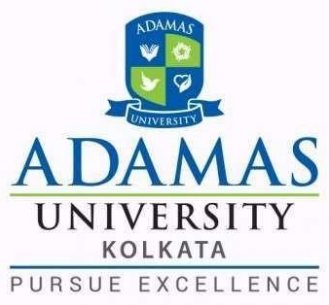
"Mitigation of DDoS attack using Anycast and OSPF as unique routing protocol"

Submitted in partial fulfilment of the requirements for the award of

Bachelor of Technology (B.Tech)

In the department of

Computer Science & Engineering



Submitted by:
Pragya Das (UG/02/BTCSE/2019/013)

Soumya Roy (UG/02/BTCSE/2019/014)

Deblin Ray (UG/02/BTCSE/2019/023)

Anindan Mondal (UG/02/BTCSE/2019/041)

Under the Guidance of

Mr. Sayantan Singha Roy

School of Engineering & Technology
ADAMAS University, Kolkata, West Bengal
Jan 2023 – July 2023

CERTIFICATE

This is to certify that the Internship report entitled “Literature survey of different mitigation technique of DDoS attack”, submitted to the School of Engineering & Technology (SOET), **ADAMAS UNIVERSITY, KOLKATA** in partial fulfilment for the completion of internship of the degree of Bachelor of Technology in the department of Computer Science & Engineering, is a record of bonafide work carried out by Pragya Das (UG/02/BTCSE/2019/013), Soumya Roy (UG/02/BTCSE/2019/014), Deblin Ray (UG/02/BTCSE/2019/023)., Anindan Mondal (UG/02/BTCSE/2019/041), under our guidance.

All help received by us from various sources have been duly acknowledged.

No part of this report has been submitted elsewhere for award of any other degree.

_____

Mr . Sayantan Singha Roy

(Assistant Professor)

_____

Dr. Pranav Kumar

(Assistant Professor)

_____

Prof. (Dr.) Sajal Saha

(HOD CSE)

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mentioning of the people whose constant guidance and encouragement made it possible. We take pleasure in presenting before you, our Internship, which is the result of a studied blend of both research and knowledge.

We express our earnest gratitude to our Mr. Sayantan Singha Roy (Assistant Professor), Department of CSE, for their constant support, encouragement and guidance. We are grateful for their cooperation and valuable suggestions.

Finally, we express our gratitude to all other members who are involved either directly or indirectly for the completion of this Internship.

DECLARATION

We, the undersigned, declare that the Internship entitled ‘Internship Topic, being submitted in partial fulfillment for the award of internship certificate in Bachelor of Engineering Degree in Computer Science & Engineering, affiliated to ADAMAS University, is the work carried out by us.

Pragya Das
(UG/02/BTCSE/2019/013)

Soumya Roy
(UG/02/BTCSE/2019/014)

Deblin Ray
(UG/02/BTCSE/2019/023)

Anindan Mondal
(UG/02/BTCSE/2019/041)

TABLE OF CONTENTS

List of Figures

List of Tables

ABSTRACT

DDoS attack[1] has been one of the most prominent security threats for the network servers throughout the world. These DDoS attacks are now happening at an unprecedented rate all over the world. Every existing server faces the probability of being attacked by someone sitting hundreds of kilometers away. The main reason these attacks are so effective is because of the difficulties in differentiating the attack traffic from regular traffic in the server. So by the time realization of the attack comes, the server has already been saturated with requests from the DDoS traffic. In our report we try to find the ways to mitigate these attacks that renders the servers useless from time to time. We have studied the different technologies used to mitigate these attacks and resolve the issue. These attacks can be brought under control through NFV and blockchain technologies. Multilevel frameworks can be created to mitigate the attacks and also Anycast has huge potential in mitigating the DDoS attacks. After going through various kinds of DDoS mitigation techniques, we can safely say that once the attack traffic has reached the server, it is impossible to get rid of the attack traffic without compromising the functionalities of the server with the current technologies. The server will shut down for a certain amount of time even with the help of the fastest mitigation techniques.

CHAPTER 1

INTRODUCTION

1.1 Background

The DDoS attacks has possibly been the one of the biggest security issues in servers for quiet some time. The first known distributed denial of service attack occurred in 1996 when Panix, now one of the oldest internet service providers, was knocked offline for several days by a SYN flood. We dived into the techniques about how these attacks are being stopped. DDoS attacks has a wide variety so it is difficult to identify and mitigate. In 2022, network layer DDoS attacks increased by 109% YoY. Activity increased steadily from 731 attacks per day on average in April to 845 in may, to 1195 in June. June 20 and 21 were the most turbulent days with 1815 and 1735 attacks[2]. Cisco predicts that the total number of DDoS attacks will double from the 7.9 million seen in 2018 to something over 15 million by 2023[13]. DDoS Protection Services enable organizations to keep their applications/ websites available round the clock by monitoring traffic and preventing distributed denial of service attacks. DDoS Protection helps organizations to be well-equipped for the lurking threat of DDoS attacks[14].

1.2 Objective of the Project

The purpose of this project is to complete a literature survey on DDoS attacks on different network layers. The purpose is to find the optimal way to prevent these attacks. The goal was to study the various ways the DDoS attacks are dealt with after it has already reached the target server. The target was to find a technique that is able to separate and to get rid of the attack traffic as quickly as possible, so that the server does not lose its functionalities for a long amount of time. Making sure that the server is back and working at its full potential soon after a DDoS attack and thus making sure that the service provided by the server is mostly uninterrupted.

CHAPTER 2
LITERATURE SURVEY

2.1 An Efficient system to stumble on and Mitigate DDoS attack in cloud Environment

K Manju Mohan et. al. [4] have proposed a system to mitigate DDoS attacks in cloud environment proposed framework plans to identify DDoS assault in cloud conditions. And furthermore to mitigate them. In the proposed method there are mainly two phases. In the first phase the IP address of the packet is captured using packet capturing tool such as Wireshark. The second phase is the classification phase. In this stage the Ip address is contrasted with the generally prepared classifier with recognize whether it is a typical or a went after packet. If it's a normal packet, sent to its objective else it is given to the avoidance stage. In the proposed framework the first stage is putting away the records in a SQL server and afterward information is disregarded the network between a few client frameworks and server machines. Then the technique is finished. SVM[4] characterization technique is utilized for preparing and testing the information. DDoS assault alleviation strategy additionally executed in this proposed work. In the proposed strategy the client getting to for the cloud to download a document the client needs to get an arbitrarily produced code from the owner of the file. For this the client needs to send a request to the owner. Assuming the owner does not allow then the client can't get to it. The qualities are put away in scrambled structure. Consequently, attack can be moderated.

2.2 A study on DDoS attacks, danger and its prevention

Chakaraborty S. et. al. [1] have proposed a system to mitigate DDoS attacks for ISPs. According to the findings, local ISPs should take the following steps to strengthen security against DDoS attacks in a cost-effective way. Firstly, they should assign usernames and passwords to each user who uses the router to ensure that each passive connection's interface has an RPF. Secondly, use virtual terminal filters to disable Telnet on virtual terminals and only allow SSH-based connections, as well as TACACS [1] for password authentication, to prevent public routers from receiving responses from your router. Finally, we can set up security laboratories, and if that's not an option, keep at least one extra router and server around to test out new services rather than deploying them. Reduce the total number of transit providers by partnering with other regional ISPs to take advantage of benefits like renting scrubbing center, managing out-of-band, and perhaps creating better security facilities.

2.3 DDoS Attack Detection Method and Mitigation Using Pattern of the Flow

Sanmorino A. , Yazid S. et. al. [3] have proposed a technique to mitigate DDoS attacks using pattern of the flow. The solution is to use the flow patterns that the server has been supplied. We can tell whether a packet is the product of a DDoS assault using this flow pattern. The pattern of flow entries is applied to the router or switch that functioned as a conduit between the botnet[3] and the server that was the assault's target. To ascertain a flow's pattern, data from all incoming flows may be extracted. the data gleaned from a stream, including the transfer protocol, flow size, and packet count as well as the source IP address, source port, destination IP address, and destination port. Packets. One of the DDoS attack flow section patterns is the size of the usual number of packets per flow over a set amount of time. By identifying the flow entry pattern of the request made by the DDoS traffic, we can recognize and mitigate the DDoS attack. The next step is to reject packets originating from the attacker. This level includes support for firewalls as a dealing system. We used two-layer firewall in the replicas. The first firewall architecture includes the vast majority of network layers[8]. The layers process the packets, transmit packets and packet data to the primary firewall engine, and retain and transmit packet settings to succeeding levels. The primary firewall engine analyses the packet data to at least one introduced channel to demonstrate to the layers how to treat the packet. We may thus conclude that a packet can only be sent if it comes from an authorized user.

2.4 Towards Mitigation of Low and Slow Application DDoS Attacks

Shtern M. et. al. [5] have proposed a system to mitigate DDoS attacks for Low and Slow application. This section proposes the solution architecture for mitigating LSDDoS attacks aimed at applications running on SDI. This gives a system that can identify and resist LSDDoS attacks a standardised design paradigm. Malicious traffic may be detected by the LSDDoS Detection Sensor, which may also determine whether an LSDDoS assault is taking place. The Shark Tank is created, and it concentrates unfavourable traffic there. The Automation Controller is also accountable for the application being healed. the software that is resistant to LSDDoS attacks[10]. gathers performance data from every component. Fish Tank: It minimises the effects of DDoS assaults while also enabling the system to record such attacks for later use. Performance-Based Methodology Having a model of the application's behaviour when there isn't an attack (normal behaviour) and comparing it with the application's attack-related characteristics (abnormal behaviour) is one technique to identify the LSDDoS attack and decrease its impact.

2.5 Mitigation using Nfv

Alharbi T. et. al. [7] In this paper a two-stage DDoS mitigation framework by leveraging NFV and SDN is proposed.
This framework consists of a traffic screening stage and a service stage. The traffic screener will analyse the traffic and determine what next-stage processes are needed for a traffic flow. Network-layer security, applicationlayer security, and/or certain network services are applied to a traffic flow according to the screening and analysis. The proposed DDoS mitigation scheme can be deployed in the organization's datacentre at the premises to reduce latency and achieve better privacy and security.

Screening mechanism

In some cases, the increase in incoming traffic is due to special events and seasons, and is not an abnormal flood. The traffic screener inspects the traffic using the algorithms and policies based on traffic pattern and packet features. It makes a decision whether a traffic flow is likely benign or malicious, and generates the outputs . In case of malicious flood, the traffic screener requests instantiation of the required network and/or application layer VSF from the orchestrator. However, if the high volume of the incoming traffic is due to legitimate requests, the traffic screener requests an increase in the capacity of the needed VNF such as DNS server or other functions to handle the traffic.

Resource allocation

To adapt to the changing demand for computing, storage, and bandwidth resources, the resource allocation module will specify the needed resources for any VNF, any VSF, and the traffic screener. Before the traffic screener sends an instantiation request of any virtualized function to the orchestrator, the resource allocation algorithm is invoked to determine the allocated resources based on the demand and availability. If more resources are needed for a VNF or VSF, the resource allocation module will work with the orchestrator to scale up the resource for the virtualized function or generate another instance.

2.6 Anycast and its potential for DDoS mitigation

De Vries B. W. et. al. [2] have proposed a system to mitigate DDoS attacks using Anycast to configure multiple sever with same IP address. With this we can confine the attack to a limited area. So the server might be unavailable to a fraction of its users, rather than becoming totally unavailable. IP any cast reduces data redundancy. and provides high availability. It is widely used in DNS and CDN. As there are multiple server with same If there's no single point of failure. So Anycast is effective against DDoS attack, as it cannot shut down whole server at once. For example, on November 2015, the DNS[2] root servers received so many requests that it saturated some of their network connection, but it was. limited as 11 out of 13 root name servers are Anycasted. The goal is to optimize the Anycast But in a way where nee have maximum security against DDoS attack. The approach will be focused on talking with operators, to understand their procedure to make an optimized, tailor made system for them. An experimental anycast- testbed will be deployed, comparable to PEERING. It will announce and withdraw IP prefixes. RIPE Atlast framework will be used to perform active measurements. Passive measurement data will be provided by BGPmon and RIPE'S RIS[2]. With the obtained data we attempt to find ways of optimizing the placement of the nodes. Sources of DDoS attacks will be analyzed to see if it is coming from a certain area, it will further assist the optimization of the anycast catchment. Sleeping / inactive instances may be added, which will be activated in case of a DDoS attack.

2.7 A Collaborative DDoS Mitigation Solution Based on Ethereum Smart Contract an RNN-LSTM

Wani S. et. Al. [6] has presented a strategy to reduce DDoS attack using block chain. Participants of the collaborative DDoS defence (ASs and consumers) initially need to develop the four forms of smart contracts that are instantly associated with a register dased type of smart contracts(EVM- in case of Ethereum blockchain). As a result, when web servers are overloaded by attackers, the customer/AS that is being attacked stores the attackers' IP addresses in the Black IP[6] smart contract. To execute an effective traffic filtering and because of the constraint of the memory space of switches (Flow rules may expire after a certain period), thus, we design a Deep learning detection system that enables the functionality of SDN[9] controller to manage with attacks. All parties must join the multi-sig contract to build the off-chain on top of the main blockchain in the Ethereum network, or what we refer to as the lightning network, which permits the exchange of information in under a millisecond, in order to alleviate the issue of the prolonged waiting time. The subscribed autonomous systems receive the most recent lists of addresses to block and authenticate the legitimacy of the attack in less than a second by examining traffic statistics and validating the victim's address. Depending on the security policies and systems in place, several mitigation measures can be activated after obtaining the list of new attackers. Moreover, it can stop harmful traffic close to its origin. In the Blockchain-based joint DDoS defence, all participants pool their resources when the attack target is unable to defend against the attack on its own. This helps prevent harm to the resources and speeds up the defensive process.

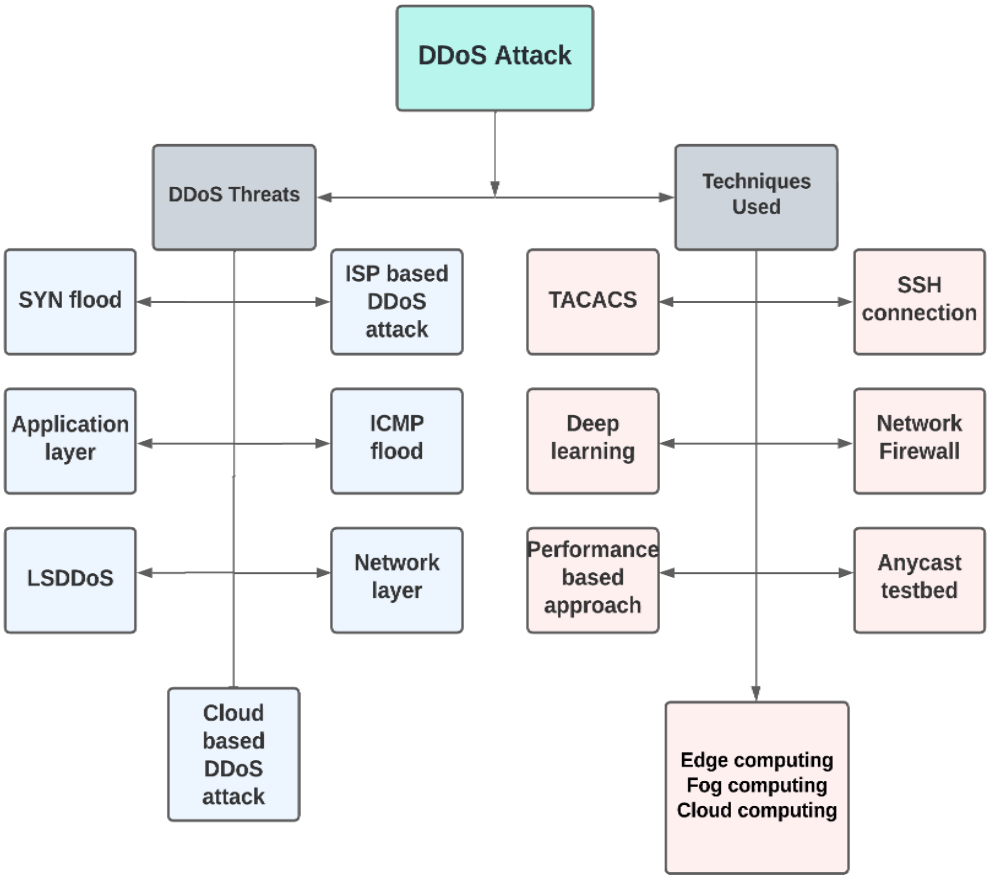


Figure 1:Networking Environment

In figure 1; a graphical representation of the DDoS attacks dealt and techniques used for mitigation in the surveyed paper has been shown.

In table 2; a comparative study has been shown in order to give an holistic view of different mitigation techniques available.

Table 1: Comparative Analysis

Reference	Types of DDoS attacks	Techniques used	Proposed methods	Research Gap
	ISP Based DDoS Attack	Disabling telnet on vtys and allowing only SSH connection. TACACS	Users should be provided usernames and passwords in order to access the router. Possibly one must work together with other regional ISPs to decrease the number of transit providers in order to profit from things like renting a scrubbing center, out- of-band management, and perhaps establishing better security facilities.	There is very little information about DDoS attacks publicly available because of organizations reluctance to reveal it. It is difficult to deploy ingress egress filtering universally.
	SYN flood and ICMP flood	Network Firewall	This method includes the support of firewall as a dealing with system. In the reproductions we utilized double layer firewall. The first firewall design incorporates a majority of network layers	Signal degradation, channel congestion, and faulty networking hardware
	Application layer	Deep learning	It Strengthen the capacity of the protection by sharing the resources among members Participants can get multiple countermeasures if an attack is highly sophisticated. It can block the attacker traffic near to the origin, by blocking the attack traffic near the source, it reduces the total cost of forwarding packets which mostly consist of massive useless traffic in case of DDoS attack	The success of the deep learning model would depend entirely on the dataset used to create the model, which would constitue a research gap.

Reference	Types of DDoS attacks	Techniques used	Proposed methods	Research Gap
	Network Layer	Anycast Testbed	We can configure multiple server with the same IP address with anycast, making it difficult for attackers to target one specific server. No single point of failure with multiple server with same IP address, so the server cannot shut down all at once.	Lack of routing agreements and lack of available data to place the anycast nodes at the right places.
	LSDDoS	SDI capabilities are used and suspicious requests are redirected for further filtering.	This provides a standardized blueprint for a system that will detect and mitigate LSDDoS threats. It provisions the Shark Tank and then redirects malicious traffic to it. The Automation Controller is also responsible for healing the application.	Higher resource usage than expected
	Cloud based DDoS attacks	Multilevel DDoS mitigation framework	It is done on three different layers Edge computing level Fog computing level Cloud computing level	Sensitive information stored in the cloud server are at a risk of leakage.

CHAPTER 3
Proposed Methodology

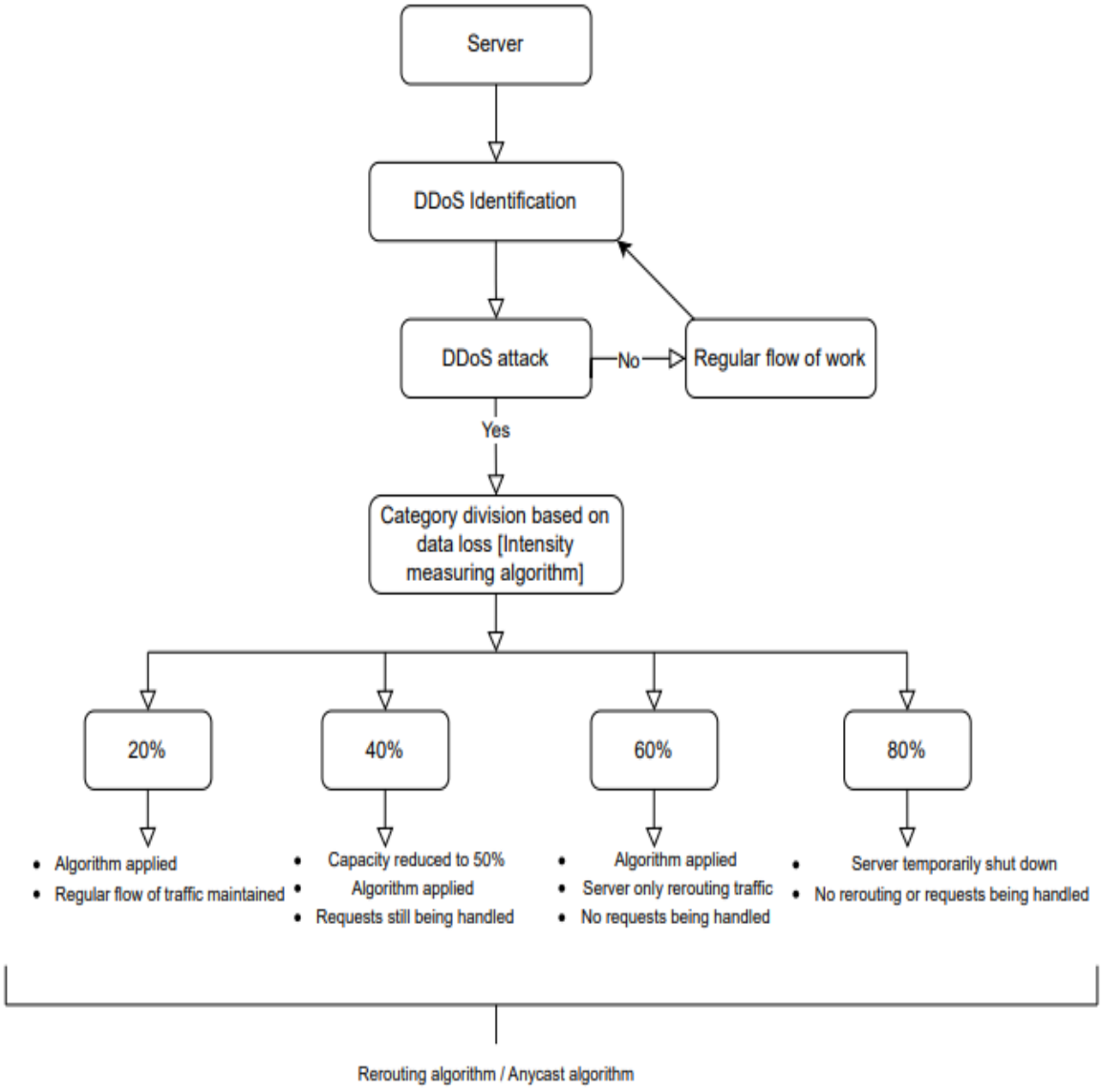


Figure 2: Networking Environment

The algorithm begins by checking the traffic flow on the server to determine if it is experiencing a DDoS attack or if it is handling regular traffic flow.

If it is determined that the server is not under a DDoS attack, it proceeds to maintain regular flow of work. Requests from users continue to be handled normally.

If the server is identified as experiencing a DDoS attack, the severity of the attack is assessed based on the amount of data loss caused by the attack. The data loss is measured as a percentage, indicating the proportion of traffic that is lost or disrupted using utility calculation mention in paper An adaptive learning routing protocol for the prevention of distributed denial of service attacks in wireless mesh networks by Sudip Misra , P. Venkata Krishna and team.

The severity analysis step categorizes the DDoS attack into different levels based on the data loss percentage. The flowchart provides divisions at 20%, 40%, 60%, and 80% data loss.

If the data loss is determined to be 20%, anycast algorithm is applied to mitigate the attack. As a measure, the capacity of the server is reduced to 50%. This reduction in capacity helps the server handle the attack more effectively.

For a DDoS attack with a data loss of 40%, anycast algorithm is applied to further mitigate the attack. The details of this algorithm are not specified in the flowchart but can be implemented based on the specific needs and capabilities of the server.

If the data loss reaches 60%, the server takes a different approach to mitigate the attack. Instead of reducing capacity, the server reroutes the incoming traffic to other servers. By doing so, it aims to maintain regular flow of traffic while still actively mitigating the ongoing DDoS attack.

In the case of an attack with a severe data loss of 80%, the flowchart recommends temporarily shutting down the server. This is done to protect the server and its resources until the attack can be fully mitigated.

If the server is temporarily shut down due to the severity of the DDoS attack, there will be no rerouting of traffic or handling of requests until the attack has been fully addressed.

On the other hand, if the server is not experiencing a DDoS attack, the algorithm ensures that regular flow of traffic is maintained, and requests from users are being handled without disruption.

3.1 Attack and Defence Utility Calculation Model

The given text describes a calculation method for measuring the effect of attack and defence in AL-DDoS (Active Learning-based DDoS) attacks. The method involves comparing the compound metric value with a threshold that is set based on stable data fluctuation within 1-5 hours after an attack.

In this method, the attack and defence effects are measured by calculating the change rate of the node status value caused by the attack and defence measures. The attack strength is present to 1, and the defence base strength is preset to 0 for simulation purposes. The defence effect is calculated by accumulating the defence measures' change amount.

To ensure the universality of the calculation results of attack and defence effects, the average attack effect within a certain period of time is obtained by analysing the attack effect at each moment. The peak and minimum values of the attack effect are eliminated to fully describe the attack effect.

The table 1 in the text lists the change value of the attack effect caused by the change of the attack strength and the change amount of the defence effect caused by the accumulation of defence measures. The attack and defence utility value can be used to describe the total attack on the network system during the attack and the total defence capability during the attack and defence.

However, the expression of the attack effect can only obtain the attack size at a certain point, and it cannot reflect the impact of the attack in the entire attack process due to the variability of attack and defence behaviours and limited influence of network traffic.
The passage discusses the importance of selecting appropriate metrics to measure and analyses the impact value of DDoS attacks. Several existing metrics are mentioned, including effective network throughput, average access failure time, average response time, volume of traffic, successful transaction rate, average service rate, and request rate. However, the passage argues that a more comprehensive approach is needed, taking into consideration network traffic, hardware performance, and other related indicators. To that end, the passage proposes six metrics for measuring the impact of DDoS attacks:
Network throughput rate: the total number of data packets received and sent by the network card in the server during the attack.
TCP data segment transmission rate: the number of TCP segments at any time during the attack.
IP datagram transmission rate: the rate at which IP datagrams are transmitted during the attack.
Transaction failure rate: the ratio of the number of failed accesses to the total number of accesses at any time.
Average traffic arrival time: the time it takes to successfully access the server.
Server CPU utilization: the occupation of resources by the attacker affects the performance of the server hardware. These metrics are selected based on their ability to provide a comprehensive picture of the impact of DDoS attacks, taking into account the characteristics and purposes of AL-DDoS attacks.
On the basis of threshold value, the size of the attack effect at each moment can be obtained, and then the attack utility can be used to illustrate the impact value caused by the attack type during

the entire attack process. In other words, the utility is the sum of the effects and the cumulative amount of the attack effect changing with time. If the attack effect size is F, when setting and selecting the threshold, the attack effect is 0, and the attack effect is also 0. If an attack occurs, set t_1 attack effect to be F_1, t_2 attack effect to be F_1, and t_n attack effect to be F_n. Based on the threshold setting, the calculation method of attack utility E is

$$E = \sum_{F_{n1}}^{F_{aq}} F.$$

The attack scale and intensity may change during the attack, but when the defence measures are attacked, it is Network system Network system Defence effect Attack effect Change the network system status Figure 1: Attack and defence behaviour effect in network system. Time Utility Attack Defence Secure interval Insecure interval Figure 2: Utility criteria for network system security. Changed-status network system Normal-status network system Defence effect Attack effect Attack and defence utility (system change volume) Figure 3: Attack and defence utility in attack and defence process. 4 Security and Communication Networks necessary to know whether the defence effect is constant. /e defence effect can be obtained by comparing different attack effects. In this case, the average attack force will be calculated.

Table 2:Attack and defence parameter simulation calculation

Duration	DDoS Intensity	Defence Strength	Node Availability	DDoS Effect	Defence Effect
1	1	0	1	1	0
1	1	1	0	1	1
1	2	0	2	2	0
1	2	1	1	2	1

3.2 Utility Calculation

The given passage describes the concept of attack and defence utility value. It represents the cumulative value of attack and defence effects in the entire process of attack and defence. The role of attack and defence means changes continuously during the process, and hence, the evaluation process must consider the utility value of both the effects.

The attack utility value (E_A) and defence utility value (E_D) can be calculated using the following equations:

$$\mathbf{E_A = \overline{A_X} \cdot t,}$$

$$\mathbf{E_D = \overline{D_X} \cdot t.}$$

where t is the attack duration, X is the attack intensity, Y is the defence intensity, A_X is the average attack force, and D_X is the average defence force.

The attack utility value (E_A) is calculated by multiplying the attack duration (t), attack intensity (X), and average attack force (A_X). Similarly, the defence utility value (E_D) is calculated by multiplying the defence duration (t), defence intensity (Y), and average defence force ($\overline{DX}$).

The attack and defence utility values provide a quantitative measure of the effectiveness of attack and defence mechanisms in a cybersecurity context. These values can be used to compare different attack and defence strategies and help in decision-making to improve cybersecurity posture.

3.3 Ip-Anycast

IP anycast is a networking technique that allows multiple hosts or servers to share the same IP address. Unlike unicast, where a single IP address corresponds to a single host, anycast IP addresses can be associated with multiple hosts located in different geographic locations. When a client sends a request to an anycast IP address, the routing infrastructure determines the nearest host based on factors such as network topology or routing metrics. The client's traffic is then routed to the closest host providing the anycast service, improving efficiency and reducing latency. Anycast is commonly used for high availability, load balancing, and content distribution, as it enables the distribution of services across multiple locations while appearing as a single IP address to clients.

Anycast rerouting using OSPF (Open Shortest Path First) can be employed as an effective strategy to mitigate the impact of a DDoS (Distributed Denial of Service) attack. By leveraging OSPF's dynamic routing capabilities, traffic can be automatically rerouted from targeted anycast servers to alternative ones that are not under attack. This redistribution of traffic helps distribute the attack load and minimizes the impact on the targeted network. Upon detecting a DDoS attack, OSPF routing configurations can be modified to increase the cost or manipulate metrics associated with the affected anycast IP address. OSPF routers then recalculate the shortest path, updating their routing tables to reroute traffic away from the targeted anycast servers. By continuously monitoring the network and adapting the OSPF configuration, anycast rerouting using OSPF aids in maintaining service availability and effectively mitigating the impact of DDoS attacks.

BGP (Border Gateway Protocol) plays a crucial role in determining the topologically closest anycast instance presented to users accessing an anycast IP address. BGP calculates the shortest path between networks (autonomous systems) to direct traffic from one network to its destination. This routing mechanism ensures that users are directed to the nearest anycast instance, reducing latency and improving performance.

While reduced latency is a significant benefit of anycast, it requires careful configuration and planning to achieve. Properly configuring BGP routing and anycast instances is necessary to ensure traffic localization and optimal routing for users. With the replication of services across multiple anycast instances, IP-anycast also provides scalability and resilience, as traffic can be distributed among multiple servers or locations.

One prominent application of IP-anycast is in the Domain Name System (DNS). The DNS protocol, being stateless and typically requiring a single request and response, is well-suited for anycast. In fact, twelve out of the thirteen root DNS operators (A-M) employ IP-anycast to enhance the performance, scalability, and resilience of the DNS infrastructure.

Overall, IP-anycast, with the support of BGP and careful configuration, offers benefits such as reduced latency, scalability, and resilience, making it a valuable technique for distributing services across multiple locations and improving network performance.

3.3.1 Catchment Measurement

- The methods to determine the catchment of an anycast service can be roughly divided in two categories:

    1. Passive Measurements:
       Passive methods register all source IP-addresses reaching each site at the site itself. This results in the catchment among all IP addresses that connected to the anycast service and is almost always incomplete or biased towards the user population of the anycast service.
    2. Active Measurements:
       Active methods use some kind of vantage points (VPs) to actively send some message to the anycast prefix and infer the anycast site reached from the response.

- Another method to measure the catchment of an anycast service is by sending ICMP echo request with anycast prefix as the source and listening for the responses on the anycast instances.

3.3.2 Controlling Catchment

With BGP, anycast operators have few options for influencing the capture of anycast instances. Choosing the upstream AS and entering into peering agreements can have an impact on the catchment. or by artificially lengthening the path to a particular instance using, for instance, AS-path prepending. However, due to their trial-anderror nature, these strategies frequently result in unanticipated outcomes. We map the attack traffic on anycast instances to determine the impact of a DDoS attack on a specific anycast deployment configuration. Each instance's catchment, which maps a significant portion of the IPv4 address space to instances, is combined.

3.3.3 Anycast Testbed

Utilizing ICMP Echo requests and responses, or "ping," the testbed is able to determine the catchment of each of its instances. The anycast prefix will be used as the source IP address each time one of the instances sends an ICMP Echo request to a particular IP address. BGP will route the response to the "closest" anycast instance and address it to the anycast prefix. The ICMP Echo response will be sent to the instance that is closest, and the source IP address of the response will be determined to belong to this instance's catchment.

3.3.4 Measurements

The measurements aim to determine the most effective AS-path prepending mitigation strategies for various DDoS attack combinations and anycast deployment scenarios. We evaluate the effects of various AS path-prepending strategies for each combination, select those that have a positive effect, and possibly refine them.

3.3.5 Anycast Deployments

The following properties are taken into consideration when defining anycast deployments: the anycast service's objectives and guidelines, the number of anycast sites, and each site's capacity.

For anycast services, we define two types of policies:

1. Maximum accessibility for everyone, with no discrimination against users, and the service ought to be made available to as many people as possible
2. Minimum required performance, or a certain level of performance, is valued more than user accessibility.

- This study's DDoS attacks are captured by services known as Booter or Stresser . Without requiring any technical expertise in DDoS attacks, these DDoS for hire services can carry out massive DDoS attacks.
- The testbed's total number of anycast sites can only support so many instances. The DDoS attacks, on the other hand, are not actually carried out against the testbed; rather, they are simulated using the mapping of source IP addresses on the anycast testbed's catchment.

CHAPTER 4
Real Time Implementation

To achieve real-time implementation, various factors need to be considered, including:

- Hardware: The choice of hardware plays a vital role in real-time systems. High-performance processors, dedicated hardware accelerators, and specialized peripherals can be used to handle time-sensitive tasks efficiently.

- Operating System: Real-time operating systems (RTOS) are designed to provide predictable and deterministic behaviour. They prioritize tasks, manage interrupts, and provide mechanisms for synchronization and communication.

- Software Design: Designing real-time software involves careful consideration of task scheduling, resource management, and data processing. Techniques like priority-based scheduling, pre-emption, and efficient algorithms are used to ensure timely execution.

- Timing Analysis: Timing analysis is performed to guarantee that all tasks and operations can be completed within their specified deadlines. It involves estimating worst-case execution times, analysing task dependencies, and verifying timing constraints.

- Testing and Validation: Rigorous testing and validation are essential to ensure that the real-time system behaves as expected. This includes testing for worst-case scenarios, stress testing, and performance analysis under different conditions.

Real-time implementation requires a combination of hardware, software, and system-level considerations to meet the timing requirements of the application. It is a complex and specialized field that requires expertise in real-time systems design and development.

4.1 Implementation of Attack Utility Function in Python

The code provided calculates various metrics related to DDoS (Distributed Denial of Service) attack intensity based on a network traffic dataset. Here's a breakdown of what the code does:

1. It imports the pandas library, which is commonly used for data manipulation and analysis.

2. The network traffic dataset is loaded from a specified path using the pd.read_csv() function and stored in a DataFrame called df.

3. The DataFrame df is cleaned by dropping any rows with missing values using the df.dropna() function.

4. The code calculates the following metrics:

    - Packet Rate: The rate at which packets are sent, calculated by dividing the packet count by the duration.

    - Bandwidth Consumption: The average amount of bytes transferred per second, calculated by dividing the total bytes by the duration.

    - Connection Rate: The rate at which unique connections are made per second, calculated by dividing the number of unique connections by the duration.

    - dst_host_same_srv_rate: The average rate of connections to the same service among all connections.

    - wrong_fragment_count: The total count of wrong fragments in the dataset.

    - dst_host_same_src_port_rate: The average rate of connections to the same source port among all connections.

5. The calculated metrics are printed to the console as DDoS Attack Intensity Metrics.

6. The attack effects, which are the calculated metrics, are stored in a list called attack_effects.

7. The code initializes variables n and m to determine the range for calculating the attack utility.

8. A loop iterates over the range from n to m+1, calculating the attack utility as the sum of attack_effects from index i-1 to m.

9. The attack utility value is printed to the console.

10. The maximum possible value for the attack utility is assumed to be 700,000.

11. The attack utility percentage is calculated by dividing the attack utility by the maximum value and multiplying by 100.

12. The attack utility percentage is printed to the console as Attack Utility (Percentage).

The code assumes that the network traffic dataset is in a CSV file format and is located at the specified path. It also assumes that the dataset has been pre-processed to remove any missing values before running the calculations.

4.2 Snippets of the Code

```
import pandas as pd

# Load the network traffic dataset
dataset_path = '/kaggle/input/dataset-1/Kddcup99_csv.csv'
df = pd.read_csv(dataset_path)
df=df.dropna()

# Calculate packet rate
start_time = df['duration'].min()
end_time = df['duration'].max()
duration = end_time - start_time

packet_count = len(df)
packet_rate = packet_count / duration

# Calculate bandwidth consumption
total_bytes = df['src_bytes'].sum()
bandwidth_consumption = total_bytes / duration

# Calculate connection rate
unique_connections = df['service'].nunique()
connection_rate = unique_connections / duration


# Calculate dst_host_same_srv_rate
dst_host_same_srv_rate = df['dst_host_same_srv_rate'].mean()
```

Figure 3:Attack Utility Code

```
# Calculate wrong_fragment
wrong_fragment_count = df['wrong_fragment'].sum()
# Calculate dst_host_same_src_port_rate
dst_host_same_src_port_rate = df['dst_host_same_src_port_rate'].mean()

# Print results
print("DDoS Attack Intensity Metrics:")
print("Packet Count:", packet_count)
print("Duration:", duration, "seconds")
print("Packet Rate:", packet_rate, "packets per second")
print("Bandwidth Consumption:", bandwidth_consumption, "bytes per second")
print("Connection Rate:", connection_rate, "connections per second")
print("dst_host_same_srv_rate:", dst_host_same_srv_rate)
print("wrong_fragment count:", wrong_fragment_count)
print("dst_host_same_src_port_rate:", dst_host_same_src_port_rate)
attack_effects = []
attack_effects.append(dst_host_same_src_port_rate)
attack_effects.append(wrong_fragment_count)
attack_effects.append(dst_host_same_srv_rate)
attack_effects.append(connection_rate)
attack_effects.append(bandwidth_consumption)
attack_effects.append(packet_rate)
attack_effects.append(duration)
attack_effects.append(packet_count)
n=0
m=len(attack_effects)
n=0
m=len(attack_effects)
print("Attack Effect List:", attack_effects)
for i in range(n, m+1):
    attack_utility = sum(attack_effects[i-1:m])  # Calculate the attack utility


print("Attack Utility:", attack_utility)
# Assuming you have maximum possible value
maximum_value = 700000

attack_utility_percentage = (attack_utility / maximum_value) * 100

print("Attack Utility (Percentage):", attack_utility_percentage)
```

Figure 4: Attack Utility Code

4.3 Output:

```
DDoS Attack Intensity Metrics:
Packet Count: 494020
Duration: 58329 seconds
Packet Rate: 8.469543451799277 packets per second
Bandwidth Consumption: 25625.584014812528 bytes per second
Connection Rate: 0.0011315126266522656 connections per second
dst_host_same_srv_rate: 0.7537812234322496
wrong_fragment count: 3178
dst_host_same_src_port_rate: 0.6019359742520545
Attack Effect List: [0.6019359742520545, 3178, 0.7537812234322496, 0.0011315126266522656, 25625.584014812528,
8.469543451799277, 58329, 494020]
Attack Utility: 494020
Attack Utility (Percentage): 70.57428571428571
```

Figure 5:Code Output

In the given DDoS attack scenario, several metrics have been provided to measure the intensity and effect of the attack. Let's break down the metrics and their interpretation:

1. Packet Count: 494,020
   This metric indicates the total number of packets sent during the attack. In this case, 494,020 packets were observed.

2. Duration: 58,329 seconds
   Duration represents the length of time the attack lasted. In this scenario, the attack spanned 58,329 seconds.

3. Packet Rate: 8.469543451799277 packets per second
   The packet rate measures the average number of packets sent per second during the attack. Here, the average packet rate was 8.47 packets per second.

4. Bandwidth Consumption: 25,625.584014812528 bytes per second
   This metric represents the average amount of bandwidth consumed by the attack, measured in bytes per second. In this case, the attack consumed an average of 25,625.58 bytes per second.

5. Connection Rate: 0.0011315126266522656 connections per second
   Connection rate indicates the average number of new connections established per second during the attack. Here, the average connection rate was 0.00113 connections per second.

6. dst_host_same_srv_rate: 0.7537812234322496
   This metric represents the percentage of connections to the same destination host and same service among all connections made during the attack. In this case, 75.38% of connections were made to the same destination host and service.

7. wrong_fragment count: 3,178
   The wrong_fragment count measures the total number of wrong fragments observed during the attack. Here, 3,178 wrong fragments were detected.

8. dst_host_same_src_port_rate: 0.6019359742520545
This metric indicates the percentage of connections originating from the same source port among all connections made during the attack. In this case, 60.19% of connections had the same source port.

The "Attack Effect List" represents a list of the above metrics in a specific order: [dst_host_same_src_port_rate, wrong_fragment count, dst_host_same_srv_rate, connection rate, bandwidth consumption, packet rate, duration, packet count].

The "Attack Utility" value of 494,020 represents an overall measure of the attack's utility or impact. It appears to be derived from the packet count metric, as it has the same value. This metric indicates the scale or magnitude of the attack.

The "Attack Utility (Percentage)" value of 70.57% represents the attack utility as a percentage of the maximum possible utility. It suggests that this particular attack achieved 70.57% of its maximum potential utility.

Please note that these metrics provide quantitative information about the attack's intensity and effects. However, the specific interpretation and implications may depend on the context and the target system being attacked.

4.4 Implementation Of Anycast routing algorithm using OSPF in GNS3

Anycast is a network routing technique that allows multiple servers to share the same IP address. When a client sends a request to the anycast IP address, the routing infrastructure determines the closest server based on the routing protocol, such as OSPF (Open Shortest Path First). Here's a high-level overview of how you can implement anycast using OSPF:

1. Design your network topology: Set up a network infrastructure with multiple servers located in different geographical locations. These servers will host the same services or content.

2. Configure OSPF: OSPF is a link-state routing protocol that dynamically calculates the shortest path between routers. Configure OSPF on each router within your network. Ensure that all routers are running OSPF and are properly connected.

3. Assign the anycast IP address: Choose an IP address that will be used for anycast. This IP address will be shared among all the servers providing the same service. Assign the anycast IP address to the loopback interface of each server.

4. Advertise the anycast IP address: Configure OSPF to advertise the anycast IP address and its associated network. Each server will announce the anycast IP address as a loopback network in OSPF updates. This way, all routers within the OSPF domain will learn about the anycast IP and its associated network.

5. OSPF route calculation: OSPF routers will receive OSPF updates from all servers announcing the anycast IP address. The routers will calculate the shortest path to reach the anycast IP based on the OSPF metrics, such as bandwidth or cost. Each router will store the route to the anycast IP address in its routing table.

6. Client request handling: When a client sends a request to the anycast IP address, the nearest router in terms of OSPF metrics will receive the request. The router will forward the request to the server with the anycast IP address based on the stored route in its routing table.

7. Load balancing and failover: Anycast can provide load balancing and failover capabilities. By advertising the same anycast IP from multiple servers, traffic can be distributed across them based on the OSPF-calculated paths. If one server becomes unreachable, the OSPF routing protocol will update the routing table, directing traffic to the next nearest server.

By leveraging OSPF's routing capabilities, anycast implementation can ensure efficient routing and optimal server selection based on network topology and OSPF metrics. This approach allows you to provide high availability and efficient distribution of services or content to clients across multiple locations.

```
R1#
R1#conf ter
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#int f0/0
R1(config-if)#ip add 10.10.10.1 255.255.255.0
R1(config-if)#no shut
R1(config-if)#
*May 16 15:38:31.947: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state t
o up
*May 16 15:38:32.947: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthern
et0/0, changed state to up
R1(config-if)#
R1(config-if)#int lo1
R1(config-if)#
*May 16 15:39:00.411: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback1,
 changed state to up
R1(config-if)#
R1(config-if)#ip add 1.1.1.1 255.255.255.0
R1(config-if)#router ospf 1
R1(config-router)#network 10.10.10.0 0.0.00.255 area 0
R1(config-router)#
*May 16 15:44:07.655: %OSPF-5-ADJCHG: Process 1, Nbr 20.20.20.2 on FastEthernet0
/0 from LOADING to FULL, Loading Done
```

Figure 6: Router R1 Configuration

```
R2#
R2#conf ter
Enter configuration commands, one per line.  End with CNTL/Z.
R2(config)#int f0/0
R2(config-if)#ip add 10.10.10.2 255.255.255.0
R2(config-if)#no shut
R2(config-if)#
*May 16 15:39:59.239: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state t
o up
*May 16 15:40:00.239: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthern
et0/0, changed state to up
R2(config-if)#
R2(config-if)#int f6/0
R2(config-if)#ip add 20.20.20.2 255.255.255.0
R2(config-if)#no shut
R2(config-if)#
*May 16 15:40:38.547: %LINK-3-UPDOWN: Interface FastEthernet6/0, changed state t
o up
*May 16 15:40:39.547: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthern
et6/0, changed state to up
R2(config-if)#
R2(config-if)#router ospf 1
R2(config-router)#network 10.10.10.0 0.0.00.255 area 0
R2(config-router)#
*May 16 15:41:57.379: %OSPF-5-ADJCHG: Process 1, Nbr 1.1.1.1 on FastEthernet0/0
from LOADING to FULL, Loading Done
R2(config-router)#
R2(config-router)#network 20.20.20.0 0.0.00.255 area 0
R2(config-router)#
R2(config-router)#int lo1
R2(config-if)#
*May 16 15:49:57.211: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback1,
 changed state to up
R2(config-if)#ip add 2.2.2.2 255.255.255.0
R2(config-if)#
*May 16 15:54:09.879: %OSPF-5-ADJCHG: Process 1, Nbr 3.3.3.3 on FastEthernet6/0
from LOADING to FULL, Loading Done
R2(config-if)#
```

Figure 7: Router R2 Configuration

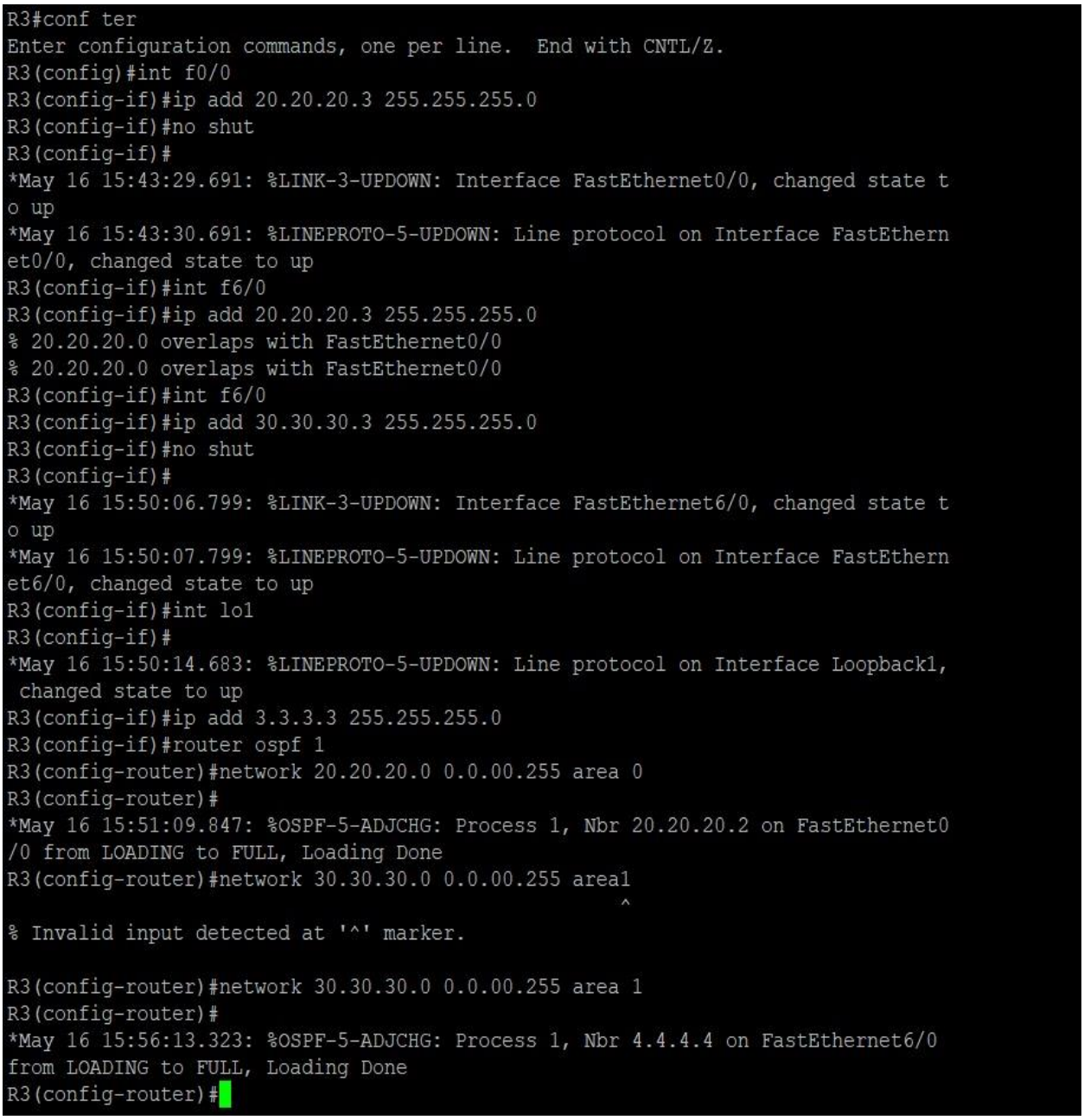


Figure 8: Router R3 Configuration

```
R4#
R4#conf ter
Enter configuration commands, one per line.  End with CNTL/Z.
R4(config)#int f0/0
R4(config-if)#ip add 30.30.30.4 255.255.255.0
R4(config-if)#no shut
R4(config-if)#
*May 16 15:48:23.511: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state t
o up
*May 16 15:48:24.511: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthern
et0/0, changed state to up
R4(config-if)#int f6/0
R4(config-if)#ip add 40.40.40.4 255.255.255.0
R4(config-if)#no shut
R4(config-if)#
*May 16 15:49:01.939: %LINK-3-UPDOWN: Interface FastEthernet6/0, changed state t
o up
*May 16 15:49:02.939: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthern
et6/0, changed state to up
R4(config-if)#int lo1
R4(config-if)#
*May 16 15:49:11.531: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback1,
 changed state to up
R4(config-if)#ip add 4.4.4.4 255.255.255.0
R4(config-if)#router ospf 1
R4(config-router)#network 30.30.30.0 0.0.00.255 area 1
R4(config-router)#
*May 16 15:50:22.975: %OSPF-5-ADJCHG: Process 1, Nbr 3.3.3.3 on FastEthernet0/0
from LOADING to FULL, Loading Done
```

Figure 9: Router R4 Configuration

```
R4(config-router)#
R4(config-router)#router ospf 1
R4(config-router)#redistribute eigrp 100 subnets
R4(config-router)#router eigrp 100
R4(config-router)#no auto-summary
R4(config-router)#network 40.40.40.0
R4(config-router)#redistribute ospf 1 metric 1000 1 10 255 1500
R4(config-router)#
*May 16 15:55:56.939: %DUAL-5-NBRCHANGE: EIGRP-IPv4 100: Neighbor 40.40.40.5 (Fa
stEthernet6/0) is up: new adjacency
R4(config-router)#
```

Figure 10: External Routing in R4

```
R5#
R5#conf ter
Enter configuration commands, one per line.  End with CNTL/Z.
R5(config)#int f0/0
R5(config-if)#ip add 40.40.40.5 255.255.255.0
R5(config-if)#no shut
R5(config-if)#
*May 16 15:49:34.835: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state t
o up
*May 16 15:49:35.835: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthern
et0/0, changed state to up
R5(config-if)#int lo1
R5(config-if)#
*May 16 15:49:43.623: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback1,
 changed state to up
R5(config-if)#ip add 5.5.5.5 255.255.255.0
```

Figure 11: Router R5 Configuration

```
R5(config-if)#router eigrp 100
R5(config-router)#no auto-summary
R5(config-router)#network 40.40.40.0
R5(config-router)#
*May 16 15:53:56.871: %DUAL-5-NBRCHANGE: EIGRP-IPv4 100: Neighbor 40.40.40.4 (Fa
stEthernet0/0) is up: new adjacency
R5(config-router)#redistribute ospf 1 metric 1000 1 10 255 1500
R5(config-router)#
```

Figure 12:External Routing in R5

4.5 OSPF Neighbour

Neighbour ID: The OSPF router ID of the neighboring router.

Priority: The OSPF priority of the neighboring router in the designated router (DR) election process.

State: The current state of the OSPF adjacency with the neighbor (e.g., Full, 2-Way, ExStart, Exchange, Loading, etc.).

Dead Time: The time remaining until the OSPF adjacency with the neighbor is considered dead.

Interface: The interface through which the OSPF adjacency is formed with the neighbor.

Interface IP Address: The IP address assigned to the interface.

Area: The OSPF area to which the neighbor belongs.

```
R1(config-router)#
R1(config-router)#do show ip ospf neigh

Neighbor ID     Pri   State           Dead Time   Address         Interface
20.20.20.2        1   FULL/BDR        00:00:30    10.10.10.2      FastEthernet0/
0
R1(config-router)#
```

Figure 13: OSPF Neighbour Nodes

4.6 OSPF Data

show ip ospf data command retrieves and displays the OSPF database information for a specific OSPF process running on a router. The output includes details about the OSPF topology, such as the OSPF neighbors, network links, and routing information.

The output includes the OSPF router link states, network link states, and summary network link states for Area 0.0.0.0. Each row provides information about a specific link or network, such as the Link ID, Advertising Router, Age, Sequence Number, and Checksum.

```
R1(config-router)#do show ip ospf data

            OSPF Router with ID (1.1.1.1) (Process ID 1)

                Router Link States (Area 0)

Link ID         ADV Router      Age         Seq#       Checksum Link count
1.1.1.1         1.1.1.1         1436        0x80000002 0x0010DB 1
3.3.3.3         3.3.3.3         650         0x80000003 0x00F8A1 1
20.20.20.2      20.20.20.2      704         0x80000004 0x00DF08 2

                Net Link States (Area 0)

Link ID         ADV Router      Age         Seq#       Checksum
10.10.10.1      1.1.1.1         1436        0x80000001 0x001FB5
20.20.20.2      20.20.20.2      704         0x80000001 0x00571C

                Summary Net Link States (Area 0)

Link ID         ADV Router      Age         Seq#       Checksum
30.30.30.0      3.3.3.3         641         0x80000001 0x00FCD8

                Summary ASB Link States (Area 0)

Link ID         ADV Router      Age         Seq#       Checksum
4.4.4.4         3.3.3.3         159         0x80000001 0x0072AC

                Type-5 AS External Link States

Link ID         ADV Router      Age         Seq#       Checksum Tag
40.40.40.0      4.4.4.4         164         0x80000001 0x00C94E 0
R1(config-router)#
```

Figure 14: Router Data

4.7 OSPF Route

show ip route command retrieves and displays the routing table entries. Each entry represents a specific network or prefix and includes details about the next hop, outgoing interface, administrative distance, and metric. The routing table contains information about both directly connected networks and learned routes from dynamic routing protocols or static routes.
The output includes several routing table entries. Each entry includes information such as the network prefix, the routing source or code (C for connected, O for OSPF, S for static, etc.), the administrative distance, the next hop IP address or exit interface, and other relevant details.

```
R1(config-router)#do show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       + - replicated route, % - next hop override

Gateway of last resort is not set

      1.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C        1.1.1.0/24 is directly connected, Loopback1
L        1.1.1.1/32 is directly connected, Loopback1
      10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C        10.10.10.0/24 is directly connected, FastEthernet0/0
L        10.10.10.1/32 is directly connected, FastEthernet0/0
      20.0.0.0/24 is subnetted, 1 subnets
O        20.20.20.0 [110/2] via 10.10.10.2, 00:25:47, FastEthernet0/0
      30.0.0.0/24 is subnetted, 1 subnets
O IA     30.30.30.0 [110/3] via 10.10.10.2, 00:13:45, FastEthernet0/0
      40.0.0.0/24 is subnetted, 1 subnets
O E2     40.40.40.0 [110/20] via 10.10.10.2, 00:05:38, FastEthernet0/0
R1(config-router)#
```

Figure 15: Routing Paths

Conclusion

Utilizing OSPF as the routing protocol in anycast deployments offers several advantages for mitigating DDoS attacks. OSPF's dynamic nature enables real-time rerouting of traffic, redirecting it away from targeted nodes and minimizing the impact of the attack. Additionally, OSPF supports load balancing across multiple anycast nodes, ensuring efficient resource utilization and preventing congestion. The fast convergence capabilities of OSPF enable quick adaptation to changing network conditions, facilitating prompt rerouting of traffic during DDoS attacks. Moreover, OSPF's scalability allows for the seamless addition of anycast nodes to handle increased traffic volumes. Collectively, these features make OSPF a valuable component in the defense against DDoS attacks when combined with anycast deploymentszing OSPF (Open Shortest Path First) as the routing protocol for anycast deployments can effectively contribute to the mitigation of DDoS (Distributed Denial of Service) attacks. OSPF is a dynamic routing protocol commonly used in IP networks, and when combined with anycast, it offers several advantages for DDoS mitigation.

Dynamic rerouting is a key advantage of OSPF as the routing protocol in anycast deployments for mitigating DDoS attacks. OSPF operates by continuously exchanging routing information among network routers, allowing it to dynamically adapt to real-time network conditions. In the event of a DDoS attack, OSPF can swiftly reroute traffic based on the changing attack patterns. By redirecting traffic away from targeted nodes, OSPF effectively mitigates the impact of the attack and ensures uninterrupted service availability. This dynamic rerouting capability enhances the resilience of the network and significantly contributes to the overall effectiveness of DDoS mitigation in anycast deployments utilizing OSPF.

Load balancing is a significant advantage of OSPF as the routing protocol in anycast deployments for DDoS mitigation. OSPF enables the distribution of traffic across multiple anycast nodes based on factors like link cost and available bandwidth. This load balancing mechanism ensures the efficient utilization of network resources. By preventing congestion on specific nodes, which are often primary targets of DDoS attacks, OSPF helps maintain the stability and availability of network services. Additionally, load balancing ensures that traffic is evenly distributed across the anycast group, optimizing the overall performance and resilience of the network. In summary, OSPF's load balancing capabilities enhance the effectiveness of DDoS mitigation in anycast deployments by efficiently managing traffic and mitigating the impact on targeted nodes.

Fast convergence is a critical advantage of OSPF as the routing protocol in anycast deployments for mitigating DDoS attacks. OSPF is specifically designed to quickly adapt to network changes, ensuring rapid convergence in response to dynamic traffic patterns. During a DDoS attack, when traffic patterns can fluctuate rapidly, OSPF's fast convergence capabilities efficiently propagate routing updates throughout the network. This swift convergence enables OSPF to promptly redirect traffic away from targeted nodes, minimizing the impact on service availability. By minimizing the time taken to adapt to changes, OSPF enhances the network's resilience and improves its ability to effectively mitigate the impact of DDoS attacks in anycast deployments.

Scalability is a significant benefit of OSPF as the routing protocol in anycast deployments for DDoS mitigation. OSPF is designed to handle large networks comprising numerous routers and subnets. When integrated with anycast, OSPF enables the deployment of multiple anycast nodes across different geographic locations. This scalability allows for the efficient distribution of traffic and the absorption of high-volume DDoS attacks without overwhelming a single node.

Scalability is a significant benefit of OSPF as the routing protocol in anycast deployments for DDoS mitigation. OSPF is designed to handle large networks comprising numerous routers and subnets. When integrated with anycast, OSPF enables the deployment of multiple anycast nodes across different geographic locations. This scalability allows for the efficient distribution of traffic and the absorption of high-volume DDoS attacks without overwhelming a single node.

By leveraging OSPF's scalability, anycast deployments can effectively handle increased traffic loads and distribute them across multiple nodes. This capability helps prevent congestion and ensures that no single node becomes a bottleneck during DDoS attacks. The ability to deploy anycast nodes across various locations also enhances the network's resilience by providing redundancy and diversifying the attack surface.

OSPF brings flexibility and manageability to anycast deployments for DDoS mitigation. It allows network administrators to customize network design, define areas, and set policies to meet specific requirements. By fine-tuning routing metrics, OSPF enables preferential routing for legitimate traffic and effective traffic filtering for malicious traffic, strengthening the overall DDoS mitigation strategy.

However, it's important to note that OSPF should be supplemented with other security measures to provide comprehensive protection against DDoS attacks. Implementing additional measures such as traffic filtering, rate limiting, and intrusion detection systems alongside OSPF enhances the overall defense against DDoS attacks and helps ensure the integrity and availability of network services.

OSPF's flexibility and manageability, when combined with anycast, offer dynamic rerouting, load balancing, fast convergence, scalability, redundancy, and effective DDoS mitigation strategies. These benefits contribute to the resilience and availability of network services during DDoS attacks, but a comprehensive security approach should include multiple protective measures for optimal defense.

References

[1] Chakraborty, S., Kumar, P., & Sinha, B. (2019). A study on ddos attacks, danger and its prevention. *Int. J. Res. Anal. Rev*, *6*(2), 10-15.

[2] de Vries, W. B., Schmidt, R. D. O., & Pras, A. (2016). Anycast and its potential for DDoS mitigation. In *Management and Security in the Age of*

[3] *Hyperconnectivity: 10th IFIP WG 6.6 International Conference on Autonomous Infrastructure, Management, and Security, AIMS 2016, Munich, Germany, June 20-23, 2016, Proceedings 10* (pp. 147151). Springer International Publishing.

[4] Sanmorino, A., & Yazid, S. (2013, March). DDoS attack detection method and mitigation using pattern of the flow. In *2013 International conference of Information and communication technology (ICoICT)* (pp. 12-16). IEEE.

[5] Yan, Q., Huang, W., Luo, X., Gong, Q., & Yu, F. R. (2018). A multi-level DDoS mitigation framework for the industrial Internet of Things. *IEEE Communications Magazine*, *56*(2), 30-36.

[6] Shtern, M., Sandel, R., Litoiu, M., Bachalo, C., & Theodorou, V. (2014, March). Towards mitigation of low and slow application ddos attacks. In *2014 IEEE international conference on cloud engineering* (pp. 604-609)IEEE.

[7] Essaid, M., Kim, D., Maeng, S. H., Park, S., & Ju, H. T. (2019, September). A collaborative DDoS mitigation solution based on ethereum smart contract and RNNLSTM. In *2019 20th Asia-Pacific Network Operations and Management Symposium (APNOMS)* (pp.1- 6).IEEE.

[8] Yaegashi, R., Hisano, D., & Nakayama, Y. (2021, January). Light-weight DDoS mitigation at network edge with limited resources. In *2021 IEEE 18th Annual Consumer Communications & Networking Conference (CCNC)* (pp. 1-6). IEEE.

[9] Deepthi, S., Hemanth, K. S. S., Rajesh, D., & Kalyani, M. (2015). A Novel Approach for DDoS Mitigation with Router. In *Intelligent Computing, Communication and Devices: Proceedings of ICCD 2014, Volume 1* (pp.701-707). Springer India.

[10] Obaid, H. S., & Abeed, E. H. (2020). Dos and DDoS attacks at OSI layers. *International Journal of Multidisciplinary Research and Publications*, *2*(8), 1-9.

[11] Devassy, J. J. (2021). *Detection of Application Layer DDoS Attack Using Logistic Regression* (Doctoral dissertation, Dublin, National College of Ireland).

[12] Yaegashi, R., Hisano, D., & Nakayama, Y. (2021, January). Light-weight DDoS mitigation at network edge with limited resources. In *2021 IEEE 18th Annual Consumer Communications & Networking Conference (CCNC)* (pp. 1-6). IEEE.

[13] Douligeris, C., & Mitrokotsa, A. (2004). DDoS attacks and defense mechanisms: classification and state-oftheart. *Computer networks*, *44*(5), 643-6

[14] Zhao, X., Peng, H., Li, X., Li, Y., Xue, J., Liang, Y., & Pei, M. (2020). Defending application layer DDoS attacks via multidimensional parallelotope. *Security and Communication Networks*, *2020*, 1-11.

[15] Li, T. S., & Li, Z. C. (2013). An Anycast Routing Algorithm Based on the Combination of Genetic Algorithm and Ant Colony Algorithm. In *Applied Mechanics and Materials* (Vol. 239, pp. 1324-1330). Trans Tech Publications Ltd.

[16] Fan, L., & Taoshen, L. (2009, May). Implementation and performance analyses of anycast QoS routing algorithm based on genetic algorithm in NS2. In *2009 Second International Conference on Information and Computing Science* (Vol. 3, pp. 368-371). IEEE

[17] Oki, Eiji; Rojas-Cessa, Roberto; Tatipamula, Mallikarjun; Vogt, Christian (April 24, 2012). *Advanced Internet Protocols, Services, and Applications*. John Wiley & Sons. pp. 102 & 103. ISBN 978-0-470-49903-0. Archived from the original on January 5, 2020.

[18] *C. Partridge; T. Mendez; W. Milliken (November 1993). Host Anycasting Service. Network Working Group. doi:10.17487/RFC1546. RFC 1546. Informational.*

[19] *D. Johnson; S. Deering (March 1999). Reserved IPv6 Subnet Anycast Addresses. Network Working Group. doi:10.17487/RFC2526. RFC 2526. Proposed Standard.*

[20] *R. Hinden; S. Deering (February 2006). IP Version 6 Addressing Architecture. Network Working Group. doi:10.17487/RFC4291. RFC 4291. Draft Standard.* Obsoletes RFC 3513. Updated by RFC 5952, 6052, 7136, 7346, 7371 and 8064

[21] *O. Troan (May 2015). B. Carpenter (ed.). Deprecating the Anycast Prefix for 6to4 Relay Routers. Internet Engineering Task Force. doi:10.17487/RFC7526. BCP 196. RFC 7526. Best Common Practice.* Obsoletes RFC 3068 and 6732

[22] Woodcock, Bill (November 14, 2019). "TCP and Anycast". *NANOG mailing list archive*. North American Network Operators Group.