International Journal of Future Generation Communication and Networking
Vol. 10, No. 10 (2017), pp.51-64
http://dx.doi.org/10.14257/ijfgcn.2017.10.10.05

Coordinator-Agent Based Distributed Defense against DDoS Attacks in Transit-Stub Networks

Karanbir Singh*[1], Kanwalvir Singh Dhindsa[2] and Bharat Bhushan[3]

[1]Research Scholar, IKG Punjab Technical University, Kapurthala (Punjab), India
[2]Professor, Dept. of CSE, Baba Banda Singh Bahadur Engineering College, `Fatehgarh Sahib (Punjab), India
[3]Associate Professor, Dept. of Computer Science, Guru Nanak Khalsa College, Yamunanagar (Haryana), India
[1]karan_nehra@yahoo.co.in, [2]kdhindsa@gmail.com, [3]bharat_dhiman@hotmail.com
**Corresponding author*

Abstract

A distributed denial of service (DDoS) attack can disrupt the normal functioning of Internet services of any organization. There exist many defense systems developed in the past, but they suffer from some disadvantages. Here, we proposed a distributed defense mechanism which detects and mitigate DDoS attacks by monitoring traffic on the edge routers of stub networks. The defense mechanism can be deployed in the form of agents and coordinator on the edge and gateway routers. The entropy based detection will monitor the traffic passing through edge routers and identify if any suspicious flow exists or not. The identified suspicious flow is further inspected to confirm whether it belongs to the legitimate flow or attack flow. The attack related information extracted from the attack packets is then passed to the coordinator. The coordinator shares this information with the neighboring coordinators so that they can instruct their agents to monitor and rate limit the traffic. The effectiveness of the defense system can be measured using some performance metrics through experiments.

***Keyword**: DoS, DDoS, Defense, Coordinator, Agent, Entropy, Transit, Stub*

1. Introduction

The attacker exploits existing internet services to perform DDoS attacks against a victim [1-6]. These kinds of attacks prevent legitimate users from accessing a particular network service by sending large unwanted traffic to victim machine/network. The growth of the Internet in the recent years left many systems vulnerable to the attackers. Attackers use those vulnerable machines to launch a coordinated attack against any target/network. A DDoS attack [7-8] is a coordinated, large-scale attack performed against the services of a target machine or network. The attack can be launched through a huge number of intermediate compromised machines on the internet. An attacker is a mastermind behind the attack which can generate floods of attack traffic through a large number of botnets to consume bandwidth and resources of a specific target. DDoS attacks put major challenges to organizations like the Internet and hosting providers, which can suffer blows to bandwidth, reputation, and bottom line. The attacker performs a DDoS attack by initially scanning some vulnerable machines on the Internet and gains their access. The attacker takes their control by inserting some malicious code or executing attack commands using some hacking tools. There can hundreds or thousands of

Received (July 19, 2017), Review Result (September 11, 2017), Accepted (September 18, 2017)

compromised machines and these machines are usually called as ‘zombies.’ These zombies’ machines collectively form a group called as the ‘botnet.’ The strength of attack will merely depend on the size of the botnet. The disaster caused by the attack can be intensified by increasing the botnet size, larger the botnet size more effective will be the output of DDoS attack.

In [9], we have compared various DDoS defense mechanisms based on their deployment locations and the outcome shows that distributed DDoS defense system can more effectively control the flood of attack traffic. In [10], we also compared centralized and distributed DDoS defense mechanism and proved that distributed defense is much better as compared to centralized defense. This motivation helps us to work on a defense mechanism which works in distributed environment. So here, we are going to propose a defense method which can identify and drop the attack traffic at distributed points of the Internet. The defense system can be put in the form of agents on the edge routers of the stub networks. An agent is a software program, which can work on the behalf of ISPs (Internet Service Providers) and perform a specific task assigned to them [11, 12]. The agents monitor the traffic heading towards a particular network/destination. The agents will observe the traffic passing through the edge routers and identify the happening of a DDoS attack. The packets related to DDoS attack will be identified and dropped. The agents will also communicate with each other to work as a team with a common goal of defending the victim from various kinds of DDoS attacks. The agent's shares attack related information with each other through the coordinator using secure messaging. In this way, the attack can be detected and controlled at the early stages by the cooperation of ISPs. The performance of the defense system can be evaluated by conducting experiments in the presence and absence of attacks.

The flow of this paper is organized as follows. The existing Internet architecture and strategy for the deployment of defense method is discussed in section II. Section III highlights the detailed process of attack detection and defense mechanism. Section IV describes the experimentation of defense system in which the whole simulation environment is explained. The result against some related performance metrics is evaluated and reviewed in Section V. Sections VI concludes with future research work.

2. Internet & Defense Model

The Internet topology is the arrangement of how autonomous systems, routers, and hosts are connected to each other. The Internet can be divided into connected sub-networks which are under the control of different administrative authorities. These sub-networks are called domains or autonomous systems (AS). The AS are usually classified into three categories depending on the way they manage transit traffic [13]. The categories of AS are:

- **Stub AS**: A stub AS is connected to only one other autonomous system. Stubs may have other private connections, but publicly appear to have only one connection to the rest of the Internet.
- **Multi-homed AS**: A multi-homed AS is connected to two or more autonomous systems and maintains its connection to the Internet even if one AS connection fails. It is unable to carry transit traffic.
- **Transit AS**: A transit AS links one AS to another and allows communication to pass through it. ISPs, for example, offer their customer networks to access other networks and the Internet via transit AS. It can carry both local and transit traffic.

Our defense system is based on this model because the majority of existing research is based on Internet topology that focuses on the autonomous systems. So our aim is to develop a distributed defense model which provide defense against DDoS attack in source network (Stub AS). Stub autonomous systems are chosen for the deployments of defense components because most of the attack traffic is originated from these places. The traffic originated from a customer network will initially be processed by the edge router. The edge router then forwards it to the gateway node which further passes it to the core router. The core routers belong to the backbone network which carries information between different stub networks. The stub autonomous system is the best place for early detection and filtering of attack traffic and prevents it from reaching the victim. Figure 1 shows the placement of defense agents and coordinators in transit-stub based Internet topology.

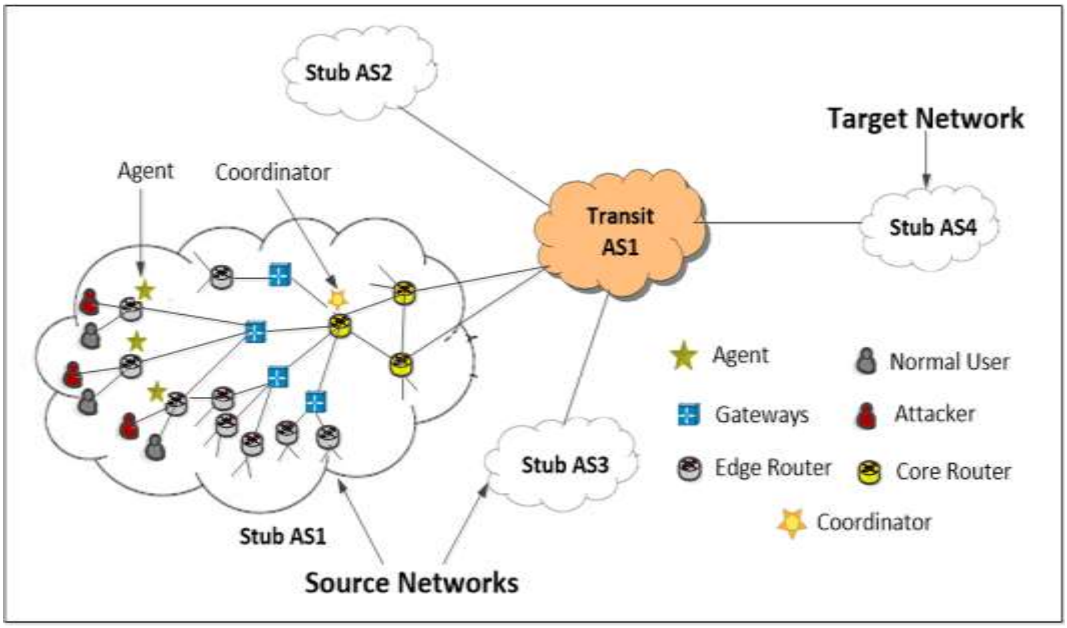

Figure 1. Internet Model with DDoS Defense Deployment

The defense system is distributed in nature as the defense components can be deployed in various locations. There are mainly two entities *i.e.* agent and coordinator, which are responsible for performing the distributed defense. The agents are specialized modules which work on behalf of ISP or particular destination network. They can implement various algorithms or procedures and can communicate with other modules like a coordinator. The other entity of the defense system is a coordinator who manages various agents in the stub network and passes attack related information as received from the agents to the neighboring coordinators. The detection algorithm can be placed in the form of agents on the edge routers of the stub network.

3. Defense Process

The defense system can be deployed in the form of agents on the edge routers of the stub networks. The agents hosting detection algorithm will continuously monitor the incoming traffic passing through the edge router. The entropy feature is used to measure the randomness in the flow [14]. The entropy of a flow and router remains stable in the absence of an attack. When a DDoS attack happens, it results in the decrease of flow as well as router entropy. The entropy decreases because any one of the flow will start dominating the traffic passing by the edge router. The detection algorithm running on the edge router will observe and calculate normalized router entropy. Some parameters like time interval, time window, and threshold values are predetermined and used in the detection algorithm. If the value of normalized router entropy becomes less than a particular threshold, then the traffic is considered as suspicious traffic. This traffic may contain one or more flows which are dominating the router traffic. The next step is to identify the suspicious flow among the flows passing through the router. The packet rate of all the flows is calculated and compared against a threshold value to know about the suspicious flow. Once the suspicious flow is identified, the next step is to confirm whether the flow is the part of the flash event or attack flow. The entropy rate of the suspicious flow is calculated. The gateway router also calculates the entropy rate of suspicious flow for the adjacent edge routers. If the difference between the entropy rates for the suspicious flow at edge and gateway router is less than a threshold value then flow will be treated as legitimate flow otherwise, it will be treated as attack flow (DDoS attack). The flow which is identified as attack flow can be dropped. Figure 2 shows the process used to carry out distributed defense against DDoS attacks.

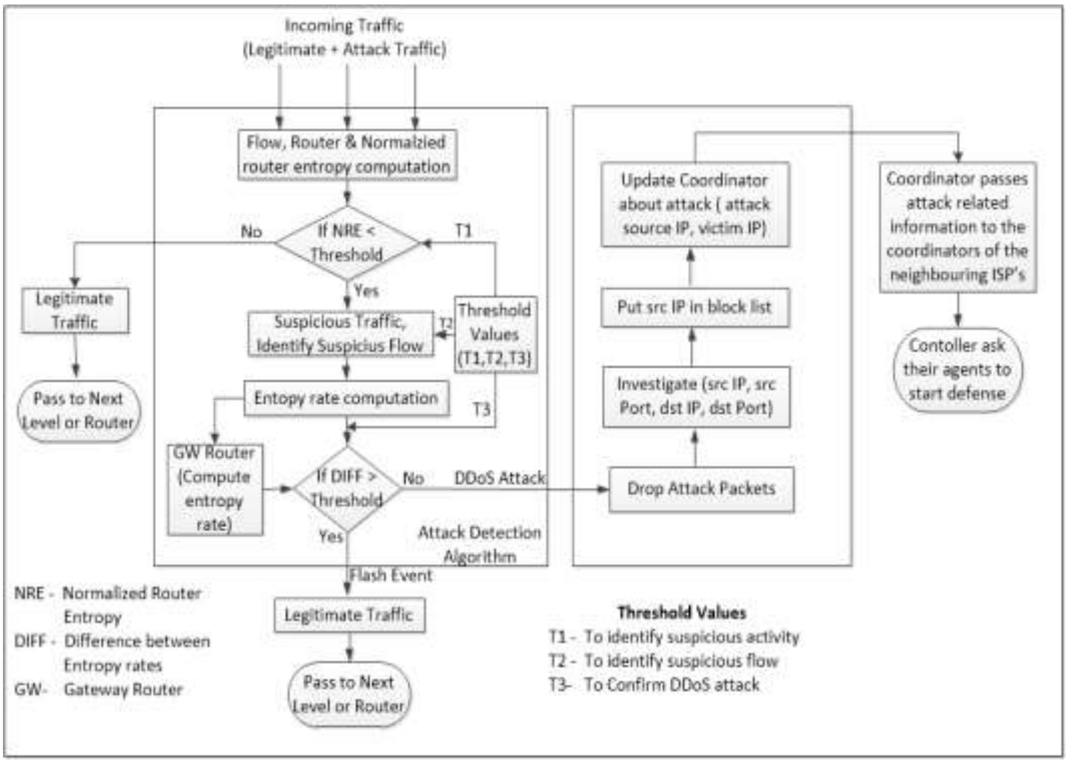
Incoming Traffic
(Legitimate + Attack Traffic)
Flow, Router & Normalzied router entropy computation
No
If NRE < Threshold
T1
Yes
Legitimate Traffic
Suspicious Traffic, Identify Suspicius Flow
T2
Threshold Values (T1,T2,T3)
Pass to Next Level or Router
Entopy rate computation
T3
GW Router (Compute entropy rate)
If DIFF > Threshold
No
DDoS Attack
Yes
Attack Detection Algorithm
Flash Event
Legitimate Traffic
Pass to Next Level or Router
NRE - Normalized Router Entropy
DIFF - Difference between Entropy rates
GW- Gateway Router
Threshold Values
T1 - To identify suspicious activity
T2 - To identify suspicious flow
T3- To Confirm DDoS attack
Update Coordinator about attack (attack source IP, victim IP)
Put src IP in block list
Investigate (src IP, src Port, dst IP, dst Port)
Drop Attack Packets
Coordinator passes attack related information to the coordinators of the neighbouring ISP's
Contoller ask their agents to start defense

Figure 2. The Defense Process

The next part of the defense process is to inform the neighbouring ISPs about the attack information. The attack related information like src IP, dest IP, src Port, and dst Port is passed to the coordinator by the agent where the attack is detected. The coordinator further updates neighbouring coordinators about this attack related information. The neighbouring coordinators then ask their agents to monitor and rate limit traffic heading towards the particular destination. The whole communication between agents and coordinator within ISP and between the coordinators of neighbouring ISPs is encrypted to protect it from attackers. The effectiveness of the defense method can be increased by the increasing the participation from ISPs.

4. Experimentation of the Proposed Defense System

The goals of simulation experiments are: illustrating the efficiency of the proposed scheme, evaluating the performance of legitimate users during attacks in the absence and presence of the defense system, and measuring collateral damages of the defense system during attack detection and prevention. The different entities involved in a DDoS attack are to be modeled in the simulation, including attackers, normal hosts, stub and transit domains. The target of the simulation is to find out whether the proposed agent-based distributed defense system can be able to detect and defend DDoS attack successfully. The efficiency of defense system can be evaluated by simulating a varying number of nodes and different scales of attacks.

4.1. Simulation Environment

The simulation environment of the proposed defense system contains the following components: OMNet++, INET framework, and ReaSE [15-17]. OMNeT++ is a discrete event simulation framework, which is commonly used for the simulation of various kinds of queuing and communication networks. OMNeT++ along with INET is well suited for simulating large networks supporting realistic Internet topologies. A simulation in OMNeT++ is composed of hierarchically structured modules, which contain the functionality of simulation. The actual functionality of TCP/IP protocol is implemented through simple modules by using one or more C++ classes. The simple modules can be connected with each other via gates and combined to form compound modules. So, a compound module can be defined, which reflects the whole working of a router or standard host system. The compound modules itself can be connected to other modules by using incoming and outgoing gates, also called a channel. A particular bandwidth or packet delay can be assigned to each channel connecting different modules. Figure 3 shows the simple model structure of OMNeT++ modules.

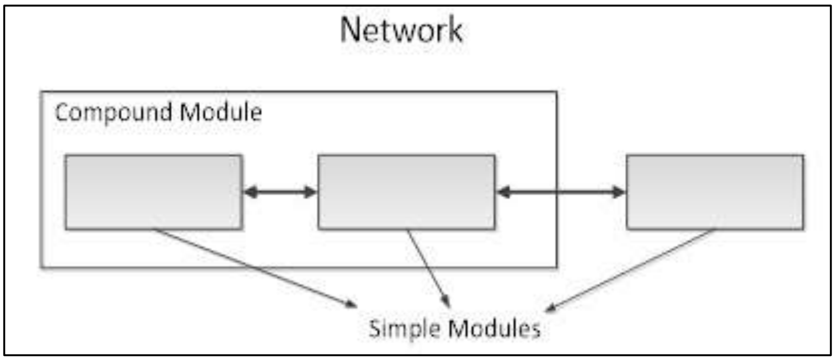

Figure 3. OMNeT++ Model Structure

To simulate the internet like networks, OMNeT++ is used along with INET. The INET is an extension of OMNeT++, which uses the similar idea of modules, which communicates by message passing. The hosts, switches, routers and other network devices are represented by OMNeT++ compound modules. These compound modules are assembled from simple modules that represent applications, protocols, and other functional components. ReaSE is an extension of INET framework, which permits us to produce realistic simulation topologies keeping multiple aspects of real world network such as some nodes, traffic patterns, link bandwidth and attack traffic, *etc*. ReaSE can also generate topology on both AS level and router level.

4.2. Simulation Scenario

To perform simulation, firstly we need realistic Internet topologies. ReaSE has been developed as an extension for OMNeT++ and is based on the protocols implemented by the INET framework. ReaSE offers own topology generation through GUI-based tools as well as traffic generation during simulations based on different network services and traffic types. It can generate realistic Internet topology of varying sizes along with background traffic based on the defined parameters. Here, we first discuss the details of network topology and then the generation of background traffic to be used in the simulation.

4.2.1. Network Topology

The network topology is simulated on two levels. In the first tier, the network topology of AS is simulated. The Positive-Feedback Preference (PFP) method is used to simulate the Internet on AS level [18]. Figure 4 shows the initial topology chosen for the simulation of the proposed defense model, which is generated by ReaSE extension for INET. The generated network topology consists of three AS (AS level topology). The chosen parameters for initial level network topology used in the experiments are given in Table 1. Figure 4 shows AS level network topology, containing 6 transit AS named tas2, tas3, tas4, tas6, tas9, tas10 and 14 stubs AS named sas0, sas1, sas5, sas7, sas8, sas11, sas12, sas13, sas14, sas15, sas16, sas17, sas18, sas19. The globe represents transit AS and stub AS is represented by clouds.

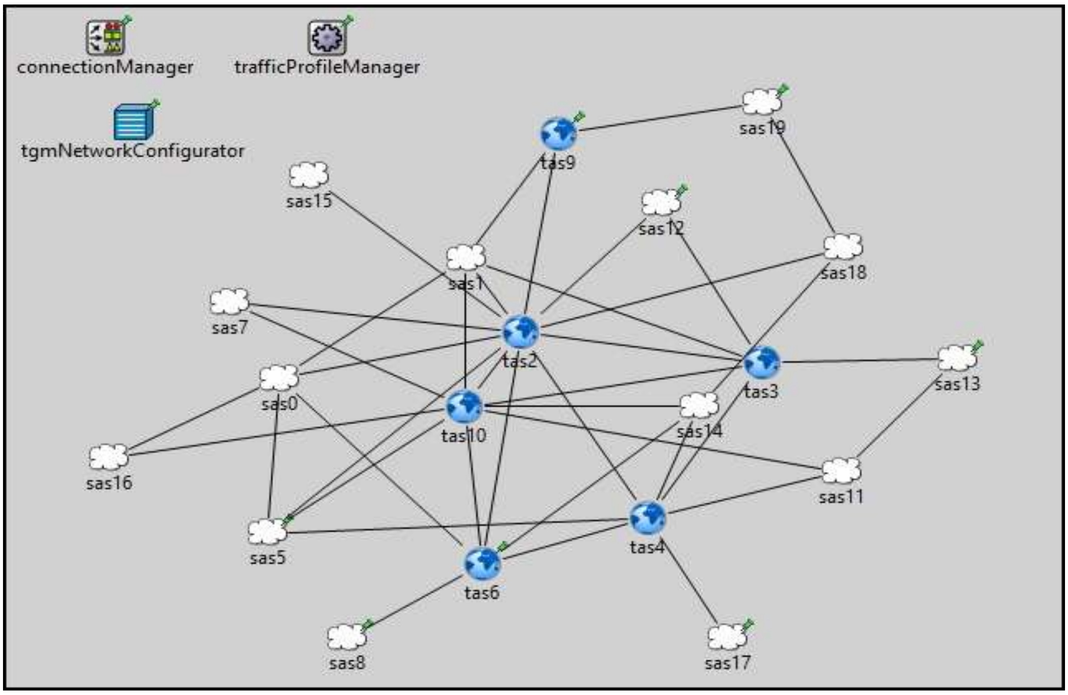

Figure 4. AS Level Network Topology

On the second level, the router-level topology is constructed. The network at router level is built on a layered approach. The first layer in each domain consists of a varying number of hosts and servers. Some hosts which turn into DDoS zombies are present at this level. The second layer contains some edge routers, which provides connectivity services for its customer hosts/networks to the external networks. The edge routers provide a means of communication between gateways and end user hosts/networks. The gateways are present on the third layer of the network. In each domain, multiple edge routers are connected with a single gateway, which results in converging several LANs at a gateway. The fourth level of the network contains core routers, which provides inter-domain connectivity. Figure 5 shows the router level topology of sas7. The other transit and stub AS also have the same kind of structure.

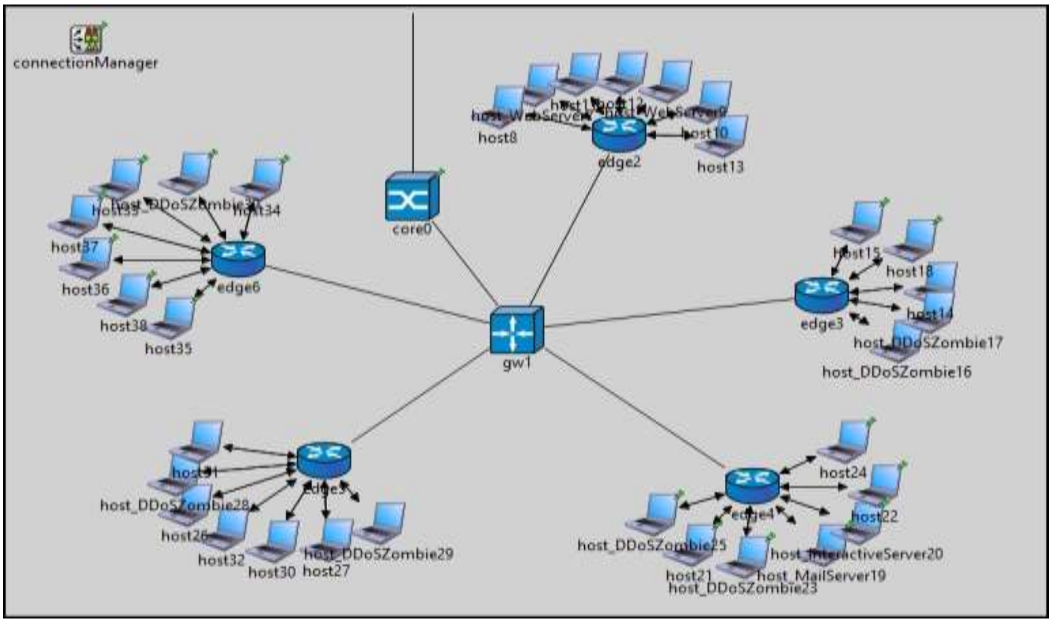

Figure 5. Router Level Topology of Stub AS7

The network scenario simulated for performance evaluation consists of 20 AS (6 transit and 14 stubs) which further contains 20 core routers, 31 gateways, 175 edge routers and 1270 end systems. Table 1 shows the various configuration parameters used in the simulation.

Table 1. Configuration Parameters for AS Level and Router Level Topology

Level	Parameters	Description	Value
AS Level	Transit Node Thresh	Minimum node degree to be a transit AS	20
	Nodes	Number of autonomous systems generated	20
	Parameter - P	PFP Growing Parameter which represents number of new node connections	0.4
	Parameter -D	PFP Growing Parameter, it characterizes the nodes preference level, depending on their connectivity after the addition of new node to the network	0.04
Router Level	R-Node-Max	Maximum router nodes per router topology	13
	R-Node-Min	Minimum router nodes per router topology	8
	Core-Ratio	Percentage of core router nodes per router topology	5.0
	Core-Cross-Link-Ratio	Percentage of core cross-link ratio	20.0
	Hosts-Per-Edge-Max	Maximum number of host per edge router	10
	Hosts-Per-Edge-Min	Minimum number of host per edge router	5

The different level of router operates at different speeds. The core routers of various domains are connected to each other through very high-speed links. Similarly, core router within each AS connects to few gateway routers via high-speed links and multiple edge routers are connected to the gateway through medium speed. Lastly, the edges routers connect to many hosts with low-speed links. Table 2 shows the link properties of the chosen topology.

Table 2. Link Properties used in ReaSE

Router Level	Link Speed	Delay
Core to Core	2.5 Gbps	1 ms
Core to Gateway	1 Gbps	1 ms
Gateway to Edge	155 Mbps	1 ms
Edge to Server	10 Mbps	5 ms
Edge to Host	0.768 Mbps	5 ms
Host to Edge	0.128 Mbps	5 ms

4.2.2. Traffic Generation

After the creation of appropriate topology, it is important to ensure that the traffic to be generated by hosts and other nodes must show resemblance with realistic Internet traffic to get accurate and meaningful results. The traffic patterns can be realistic if they show self-similar behavior [19], which requires the proper mixture of multiple types of traffic. ReaSE [17], D-ITG [20], TrafGen [21], and BonnTraffic [22] are some popular traffic generators, which can generate self-similar traffic patterns. One possible solution to attain self-similar traffic behavior is by using multiple sources of traffic which can be switched off and on based on heavy tailed intervals [23]. The other possible method is to create traffic at packet level by reproducing suitable stochastic processes for both packet sizes random variables and inter-departure time [20]. We choose ReaSE for the generation of realistic internet traffic because it combines both techniques mentioned above (*i.e.* packet level modification and multiple traffic sources) and accepts a suitable combination of different protocols based on TCP, UDP, AND ICMP to produce eight different traffic profiles and allocates a selection probability to each one these profiles. Table 3 shows the various traffic sources, protocol, and flow percentage.

Table 3. Link Properties used in ReaSE

Traffic Source	Protocol	Flow (%)
HTTP	TCP	45.5
FTP	TCP	18.5
Telnet	TCP	9.5
Interactive Traffic	TCP	10.5
Streaming Traffic	UDP	2.4
Ping Traffic	ICMP	3.6
Mail traffic	TCP	4.5
Backup Traffic	TCP	5.5

4.3.3. Attack Traffic

To test any DDoS defense mechanism, we need some malicious nodes which can produce realistic attack traffic. The Tribe Flood Network [24] is a real tool; it can be used to generate DDoS attacks. ReaSE integrates TFN (Tribe Flood Network), a real attack tool that is used to perform a DDoS attack on any host. TFN works by randomly choosing and replacing some normal host with DDoS zombies. This compound module DDoSZombie is made with a simple module TribeFloodNetwork with some other INET modules which are necessary to accomplish the functionality of an attacking system. The module TribeFloodNetwork implements the real functionality of producing attack packets as per the parameters configured in the simulation. These packets are then directly sent to the IP layer of INET framework. In our simulation, a total 238 DDoSZombies are placed across various autonomus systems, at simulation time 10th sec, the zombies start the DDoS attack based on TCP SYN packets & 92% of the zombies jointly launch the attack by sending a fixed rate TCP SYN packets to the victim Webserver62 which is in sas17.

5. Results & Discussions

To evaluate the performance of the proposed solution, we conducted a simulation experiment using different simulation parameters. The various connection related parameters like bottleneck links, delay, various traffic sources and their flow percentage are already mentioned the above section. A legitimate user sends a randomly chosen request with steady traffic rates in the range [2Kbps, 20Kbps]. If the request reaches at the server successfully, then the server will initiate the result within a certain processing delay. The legitimate and attack users are randomly distributed in different stub domains. A number of experiments are carried out by varying the number of attackers & attack intensities. The simulation will run for 25 seconds and results will be collected in the form of scalar and vector output values. The scalar has a single output value (*e.g.* the number of packets received) but vector stores series of time-value pairs during the simulation period. These statistics can later be analyzed with the data of interest. The effect of DDoS attack and defense mechanism on the performance of legitimate traffic is explained below.

5.1. Throughput

Throughput is the rate of which a packet will be successfully delivered to a destination over a communication channel. Throughput can be used to check the performance and network efficiency in a way that a high throughput offers high network performance and vice versa. The throughput is usually measured in terms of the total number of packets delivered to the destination. When an attack is launched, legitimate and attack traffic, both use the bottleneck link. So throughput is defined as a number of legitimate packets received at the destination per second. Throughput can also be measured in terms of goodput and badput respectively. Goodput is defined as the number of bytes per second of legitimate traffic that is received at the server, and badput is defined as the number of bytes per second of attack traffic that is received at the server. The throughput is measured in terms of evaluating the number of legitimate packets delivered to the destination in the following three cases.

1. In the absence of attack & defense mechanisms,
2. In the presence of attack but in the absence of defense system, and
3. In the presence of both attack & defense mechanisms.

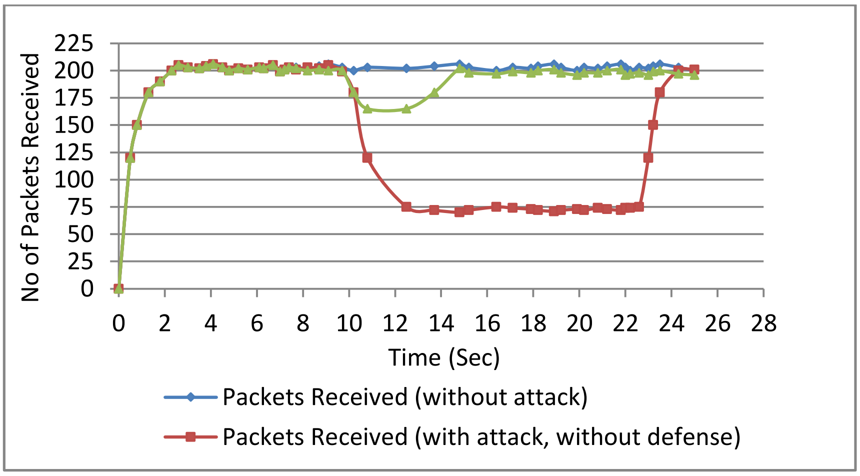

Figure 6. Throughput Variations

The attack starts at 10th seconds after the start of legitimate traffic and ends at 22nd seconds. Finally, we calculated the number of packets delivered to the destination in the cases mentioned above. Figure 6 shows the performance of legitimate packets in above mentioned three situations. The X-axis represents time intervals in seconds, and the Y-axis represents the number of legitimate packets delivered to the destination in different situations.

5.2. Response Time

In the second set of experiments, the effect of DDoS attack and defense system on the response time taken by legitimate packets during transmission is measured. Response time is the measure of the amount of time required for packets to travel across a network path from a sender to a receiver. It is the combination of time taken by a packet to travel from client to server, server delay and the time required for a packet to reach to client from server. Here we record and calculate the response time taken by legitimate packets in the cases mentioned above.

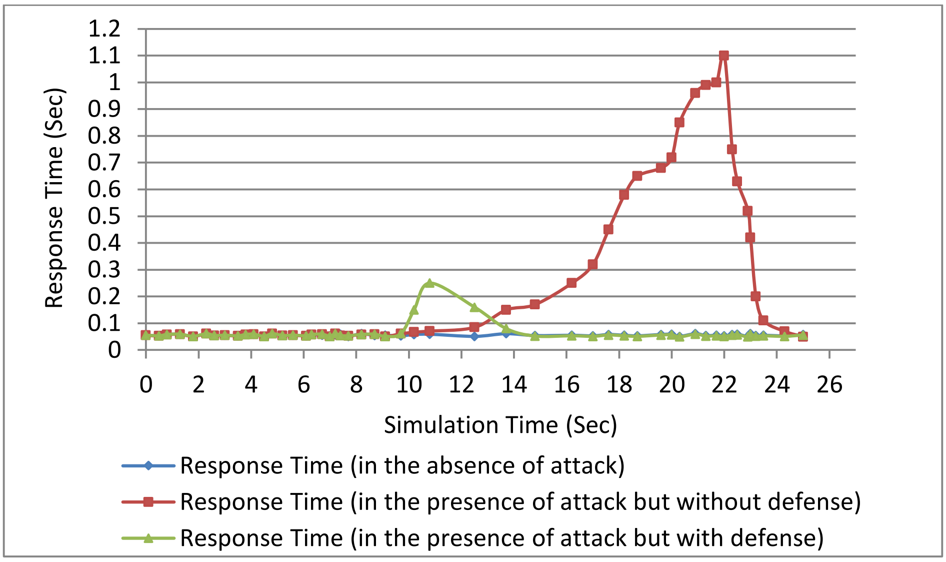

Figure 7. Response Time Variations

Initially, the response time is evaluated when there is no attack. The response time remains stable *i.e.* less than 0.1 sec, in the absence of an attack. The attack is launched at the 10th second; the response time starts increasing with the increase in attack packets strength. Figure 7 shows that the maximum response time can touch even 1.1 seconds during the attack. In the third case, we identify the effect of defense method on the performance of response time. The response time will start increasing as soon as the attack is launched during the 10th second but soon it will be controlled by the invocation of defense system at the 12th second.

5.3. Deployment

In the third set of experiments, the benefits of increased deployment are investigated. The effectiveness of the defense system can be increased if the defense system can be deployed on more number of edge routers. The edge routers of stub networks where defense system needs to be deployed are directly under the control of ISPs. So if they agree to participate in the defense process, the overall effectiveness of defense can be increased.

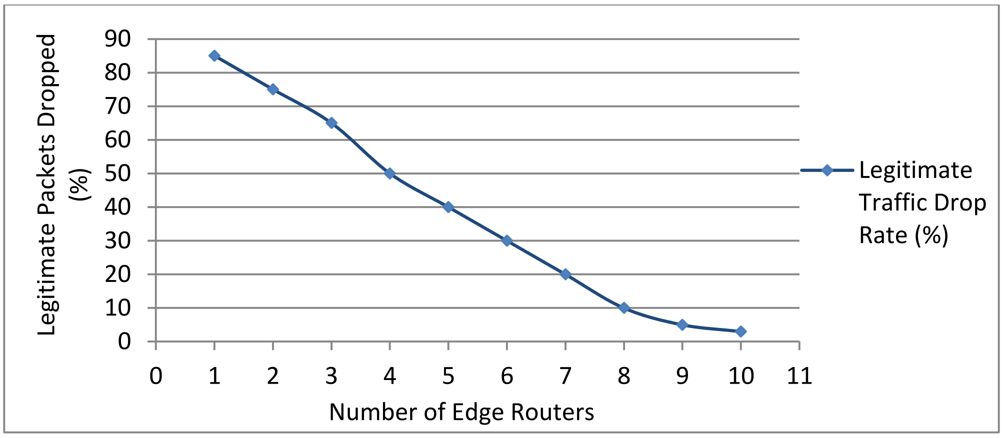

Figure 8. Edge Router Variations

Figure 8 displays the number of legitimate packets dropped due to false alarm rate will decrease gradually with the increase in the edge routers which joins the defense system. By implementing the defense system on a sufficient number of edge routers, the attack traffic can be identified and dropped more efficiently, and the number of the attack packets reaching the victim server will get decreased.

6. Conclusion & Future Work

The main focus of this paper is to propose a defense systems which detect and filter attack traffic in distributed environment. The transit-stub model of Internet topology is used for the implementation of the defense system. The agents holding detection and filtering mechanism can be placed on various edge routers of the stub autonomous system. The agents monitor the traffic passing through edge routers and identify for suspicious traffic. The suspicious traffic can further be investigated to confirm whether it is creating a DDoS attack or not. The packets which belong to attack traffic can be dropped and further investigated. The attack related information extracted from packet header will be exchanged with the coordinator. The coordinator then further shares this information with the neighboring coordinators. The nearby coordinators then ask their agents to protect the identified destination against the DDoS attack. The effectiveness of defense system will depend on two factors.

- Firstly, the threshold values used in the detection algorithm will decide the percentage of false positive and false negative which in effect control the collateral damage
- Secondly, the participation of ISPs will determine the effectiveness of defense system; more participation will ensure high legitimate packet delivery ratio

The future work is to test the performance of defense system by increasing more number of transit-stub domains in an incremental fashion. The effectiveness of defense system will also be compared against the same kind of defense mechanisms.

References

[1] T. Peng, C. Leckie and K. Ramamohanarao, “Survey of network-based defense mechanisms countering the DoS and DDoS problems”, ACM Computing Survey, vol. 39, no. 1, **(2007)**, pp. 3:1-3:42.

[2] V. Chandola, A. Banerjee and V. Kumar, “Anomaly detection: A survey. ACM Computing Survey”, vol. 41, no. 3, **(2009)**, pp. 15:1–15:72.

[3] G. Loukas and G. Oke, “Protection against denial of service attacks: A survey”, The Computer Journal, vol. 53, no. 7, **(2010**), pp. 1020–1037.

[4] M. Bhuyan, D. Bhattacharyya and J. Kalita, “Surveying port scans and their detection methodologies”, The Computer Journal, vol. 54, **(2011)**, pp. 1565–1581.

[5] H. Kashyap and D. Bhattacharyya, "A DDoS attack detection mechanism based on protocol specific traffic features", Proceedings of the Second International Conference on Computational Science, Engineering and Information Technology, Coimbatore, India, **(2012)**.

[6] S. Lin and T. Chiueh, "A Survey on Solutions to Distributed Denial of Service Attacks", Tech. Rep. TR201, Department of Computer Science, State University of New York, Stony Brook, **(2006).**

[7] P. Criscuolo, "Distributed Denial of Service, Tribe Flood Network 2000, and Stacheldraht CIAC-2319", Department of Energy Computer Incident Advisory Capability (CIAC). (Tech. Rep. UCRLID - 136939, Rev. 1), Lawrence Livermore National Laboratory, **(2000)**.

[8] B. Todd, "Distributed Denial of Service Attacks", Available: http://www.linuxsecurity.com/resource files/intrusion detection/ ddos–whitepaper.html, **(2000)**.

[9] K. Singh, N. Kaur and D. Nehra, "A comparative analysis of various deployment based DDoS defense schemes", Proceedings of 9th International Conference on Quality, Reliability, Security and Robustness in Heterogeneous Networks, Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol. 115, **(2013)**, pp. 606-616.

[10] K. Singh, K. Dhindsa and B. Bhushan, "Distributed Defense: An Edge over Centralized Defense against DDos Attacks", International Journal of Computer Network and Information Security, vol. 9, no. 3, **(2017)**, pp. 36-44.

[11] SFlow, "Traffic monitoring using sFlow", Available: http://www.sflow.org/sFlowOverview.pdf, **(2003).**

[12] M. Shafiq, A. Ali, E. Ahmad, H. Ahmad and H. Suguri, "Detection and Prevention of Distributed Denial of Services Attacks on Wide Area Networks by Collaborative Effort of Software Agents", Proceedings of International Conference on Parallel and Distributed Computing and Networks, Innsbruck, Austria, **(2005)**.

[13] Y. Rekhter and P. Gross, "Application of the Border Gateway Protocol in the Internet", Request for Comments 1772, Internet Engineering Task Force, **(1995)**.

[14] T. Cover and J. Thomas, "Elements of Information Theory", Second Edition, John Willey & Sons, **(2007)**.

[15] A. Varga and R. Horing, "An overview of the OMNeT++ simulation environment", Proceedings of the 1st International Conference on Simulation Tools and Techniques for Communications, Networks and Systems & Workshops, Marseille, France, **(2008)**.

[16] INET Framework for OMNeT++, manual, Available: https://omnetpp.org/doc/inet/api-current/inet-manual-draft.pdf.

[17] T. Gamer and M. Scharf, "Realistic simulation environment for IP-based networks", Proceedings of 1st International Conference on Simulation Tools and Techniques for Communication and Systems & Workshops, Marseille, France, **(2008)**.

[18] S. Zhou and R. Mondragon, "The positive-feedback preference model of the AS-level Internet topology", Proceedings of IEEE International Conference on Communications, Seoul, South Korea, **(2005)**.

[19] M. Crovella and A. Bestavros, "Self-similarity in World Wide Web traffic: evidence and possible causes", Journal of IEEE/ACM Transactions on Networking, vol. 5, no. 6, **(1997)**, pp. 835-846.

[20] S. Avallone, D. Emma, A. Pescap and G. Ventre, "A Practical Demonstration of Network Traffic Generation", Proceedings of International Conference 8th IMSA, Kauai, Hawaii, (**2004)**.

[21] I.Dietrich, "OMNeT++ Traffic Generator", Available: http://www7.informatik.uni-erlangen.de/~isabel/omnet/modules/TrafGen/, **(2006)**.

[22] B. Roemer, "BonnTraffic: A modular framework for generating synthetic traffic for network simulations", Available: http://web.informatik.uni-bonn.de/IV/bomonet/BonnTraffic.htm, **(2015)**.

[23] W. Willinger, M. Taqqu, R. Sherman, D. Wilson, "Self-similarity through high-variability: statistical analysis of ethernet LAN traffic at the source level," IEEE/ACM Transactions on networking, vol. 5, no. 1, **(1997)**, pp. 71-86.

[24] D. Dittrich, "The "Tribe Flood Network" distributed denial of service attack tool", Available: https://staff.washington.edu/dittrich/misc/tfn.analysis, **(1999)**.

Authors

Karanbir Singh, he is doing his Ph.D. in the field of Network Security from IKG Punjab Technical University, Kapurthala (Punjab). He obtained his MCA degree from Kurukshetra University, Kurukshetra (Haryana), India. He has a teaching and research experience of more than 13 years. He is the member of various professional bodies like IAEME, UACEE, and IACSIT. He has authored more than 7 papers in various international journals & the proceedings of reputed national and international conferences. His research interests are in the fields of Computer Networks, Network Security, and Adhoc Networks.

Kanwalvir Singh Dhindsa, he is working as Professor in the Department of CSE at Baba Banda Singh Bahadur Engg. College, Fatehgarh Sahib (Punjab). He obtained his Ph.D. in Computer Engg. (In the field of Mobile Computing & Information Systems) from Punjabi University Patiala. He has been awarded the 'Best Ph.D. Thesis Award' in International Conference held in association with Computer Society of India (CSI) at Roorkee (Uttarakhand) in Nov. 2014. He has guided many M.Tech. students & is currently guiding 7 Ph.D. scholars. He has authored more than 70 publications in various esteemed international referred journals & proceedings of reputed national and international conferences. His research interests are in the fields of Cloud Computing, Big Data, IoT, Mobile Computing, Database & Security, and Web Engineering.

Bharat Bhushan, he is employed as Head and Associate Professor in the Department of Computer Science & Applications, Guru Nanak Khalsa College, Yamunanagar (Haryana). He has done Ph.D. in Computer Science & Applications from Kurukshetra University, Kurukshetra, India. His qualification also includes MCA and Master of Science (Physics). He has teaching and research experience of more than 26 years. He is professional member of various reputed national and international associations. He has more than 30 research papers to his credit in various referred international journals and reputed international conferences. His research interests are in the fields of Software Quality and Mobile Networks.