PACKET SIMULATION OF DISTRIBUTED DENIAL OF SERVICE (DDOS) ATTACK AND RECOVERY

Author: Sandarva Khanal, Ciara Lynton
Advisor: Dr. Richard A. Dean
Department of Electrical and Computer Engineering
Morgan State University

ABSTRACT

Distributed Denial of Service (DDoS) attacks have been gaining popularity in recent years. Most research developed to defend against DDoS attacks have focused on analytical studies. However, because of the inherent nature of a DDoS attack and the scale of a network involved in the attack, analytical simulations are not always the best way to study DDoS attacks. Moreover, because DDoS attacks are considered illicit, performing real attacks to study their defense mechanisms is not an alternative. For this reason, using packet/network simulators, such as OPNET Modeler, is the best option for research purposes.

Detection of an ongoing DDoS attack, as well as simulation of a defense mechanism against the attack, is beyond the scope of this paper. However, this paper includes design recommendations to simulate an effective defense strategy to mitigate DDoS attacks. Finally, this paper introduces network links failure during simulation in an attempt to demonstrate how the network recovers during and following an attack.

KEY WORDS

Distributed Denial of Service, Network Simulation, OPNET Modeler

I. INTRODUCTION

National infrastructures (such as the financial system, power plants, or power grids, etc.) are becoming automated using computer networks. There is now, more than ever, an increasing risk that the infrastructure in place could become obsolete or even a source of chaos if the server computers are compromised. Figure 1 shows one such national infrastructure, the transportation system. Mishaps are bound to happen if, for example, an attacker was able to overload the server which controls the machine that automates the traffic lights on the road.

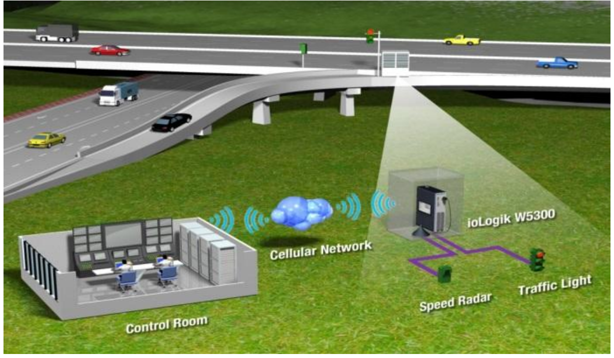


Figure 1: An Example of National Infrastructure

It is dreadful to think what might happen if a similar situation arises in power grids and power plants. National security will be in jeopardy if the computer that controls a nuclear power plant gets compromised. However, it is even more frightful to know that it is not extremely complicated to execute an attack that can do just that. Distributed Denial of Service (DDoS) attacks, a type of Denial of Service (DoS) attack, are simple attacks that can compromise a server within minutes and prevent accessibility to other users.

Distributed Denial-of-Service (DDoS) attacks overwhelm a web server by sending it many more requests for data than it can handle [1]. Recent DDoS attacks around the globe suggest that these attacks are growing stronger, both in the scale of the attack as well as in their level of sophistication [1][2]. A typical DDoS attack normally targets banks and other popular web-servers and peak at around 50 Gbps (gigabits per second) of data pouring in the server [2]. In April 2013, the internet around the world was slowed down because of the largest DDoS attack in history, peaking at 300 Gbps [2][3].Thus, it is a strong indication that DDoS attacks are critical threads to the internet. In this paper, we treat DDoS attack as a major issue in security of national infrastructure, as well as other public and private networks.

II. BACKGROUND

Denial of Service Attacks

In the context of Information Security, availability means that information is readily accessible to authorized and legitimate users. Availability attacks, sometimes called denial-of-service or DoS attacks, are much more significant in networks than in other contexts. A DoS attack occurs when an attacker (or a malicious user) attempts to completely consume all available resources of the target server and, therefore, blocks all services to legitimate users by sending massive amounts of bogus traffic to the victim. A DoS attack typically consumes bandwidth but can also consume memory, CPU cycles, file space, or any other resource that is necessary for normal operation [4]. The victim will become overwhelmed by the overload of traffic and will not be able to respond to legitimate users. Unless countermeasures are taken, the victim will remain at the mercy of the hacker for the entire duration of the attack.

Distributed Denial of Service (DDoS) Attacks

In order for a hacker to overload a victim, the hacker must be able to produce more traffic than the victim can handle. This is often difficult for a single computer to accomplish. To make his/her attacks more effective, the hacker will accumulate many computers to use in the denial of service attack [4]. This is referred to as Distributed Denial of Service (DDoS).

The attack consists of multiple stages [5]. In the first stage, the attacker uses any convenient attack to plant a Trojan horse on a target machine. That Trojan horse does not necessarily cause any harm to the target machine, so it may not be noticed. The attacker repeats this process with multiple targets. Each of these compromised target systems then becomes what is known as a zombie. In the next stage, the attacker sends a signal to all the zombies to launch the attack. The problem is escalated when different zombies are using different attack methods. For instance, some could use Smurf attacks or SYN floods to address different potential weaknesses. Then, instead of the victim's trying to defend against one denial-of-service attack from one malicious host, the victim must try to counter n attacks from the n zombies all acting at once.

In a more complex case, once enough zombies have been collected to launch an attack, the hacker will designate some machines as masters. The master will then be used to give the attack command to the other zombies, creating additional separation between the hacker and the victim.

Network Simulation

Introducing new defense mechanisms against DDoS attacks requires testing and validating the design exposition, which is not feasible by building the network and directly conducting a physical test because of the huge scale of the network involved. Also, this process requires careful attention and possibly research permission from authorities because DDoS attacks are illegal. Furthermore, the high cost associated with such test makes this testing technique almost impossible to follow, to study, and to develop the defense system.

A more attractive solution is to use a network simulation software package to simulate the performance of the network and the server system. Using this method, generating network designs produces safe, fast, reliable, and cost-effective results in order to study, enhance, and modify the design with relative ease [6].

III. KNOWN METHODS OF MITIGATING DDOS ATTACKS

Countermeasures to prevent DDoS attacks include: detection, mitigation, trace-back, and prevention [7]. In this section, some of the more predominant mechanisms to mitigate a DDoS attack are discussed in brief.

Packet Filtering

One mitigation approach filters or rate-limits attack packets [7]. In computer networks, ingress filtering is a technique used to make sure that incoming packets don't have spoofed source IP addresses in their headers. Spoofing is typically done as part of an attack, so that the attacked computer does not know where the attack is really coming from. In ingress filtering, packets coming into the network are filtered based on previous gained information from the originating network that the sending computer is not allowed to send packets with that IP address [8]. The idea is to prevent computers on your network from spoofing (acting as another) [8].

Packet filtering is an effective way to block DDoS attacks, which usually use spoofed source IP addresses. However, it is seldom used by ISPs, as filters degrade ISPs' network performance significantly [10]. Also, filtering will reduce spoofing, but not eliminate the DDoS attack. Therefore this approach, although considered effective against the older style attacks, has been superseded by the advances in denial of service 'technology' [9].

Trace-back Mechanism based on Packet Marking

There are two major methods for packet marking to defend against a DDoS attack: the deterministic packet marking (DPM) and the probabilistic packet marking (PPM) [11]. The PPM mechanism tries to mark packets with the router's IP address information by probability on the local router, so the victim can reconstruct the paths that the attack packets went through [11]. The DPM tries to mark the spare space of a packet with the packet's initial router's information. The PPM strategy can only operate in a local range of the Internet (ISP network), where the defender has the authority to manage. The DPM strategy requires all the Internet routers to be updated for packet marking. However, with only 25 spare bits available in as IP packet, the scalability of DPM is a huge problem [12].

Trace-back Mechanism based on Entropy Variation

Another more recent publication on DDoS attack and mitigation uses the trace-back technology to accurately measure the location and number of zombies that are present in the attack network [11]. In the cited paper, the authors propose a trace-back method for DDoS attacks based on entropy variations between normal and DDoS attack traffic.

This method is fundamentally different from commonly used packet marking techniques. The proposed method in the paper is robust against packet pollution, because it does not use packet marking. It is not memory intensive, it is efficiently scalable, and it is independent of attack traffic patterns. Experiments using the proposed methods were shown to accurately trace-back thousands of zombies within 20 seconds (approximately) in a large-scale attack network.

IV. DESIGN RECOMMENDATIONS TO SIMULATE DEFENSE AGAINST DDOS

The following table summarizes the design recommendations that should exist in a proposed defense mechanism, and that would be easily implemented within OPNET architecture:

Criteria	Recommendations/ Requirements
Packets Marking	Avoid Probabilistic or Deterministic Packet Marking (PPM, DPM)
Processing	Perform minimal processing at the victim node Most processing work for defense will be done by the control station over the network
Routing	No change in existing routing software
Defense Mechanism	Should archive real-time trace-back to attackers Should implement global traffic monitoring and control mechanism to distribute the trace-back workload and make is efficient
Trace-back Method	Should be initiated by the target server and independent of packet pollution, attack pattern, attack network topology, and defense network topology
OPNET Deployment	Must be able to implement easily by adding an independent module

Table 1: Design Recommendations to Effeciently Simulate Defense Against a DDOS Attack

V. PROJECT #1- TYPICAL TCP/IP LAN WITH ROUTER CONNECTED TO INTERNET

A vast and complicated network is needed to be simulated prior to simulating a DDoS attack and defense mechanisms. This network is obviously not pre-configured in OPNET and therefore preliminary simulation of them had to be done. The following assumptions are made to reduce the complexity of the simulation studies, and to eradicate any initial confusion during experiments.

- There is only one server, but two legitimate users and hundreds of zombies
- Legitimate users HTTP packets inter-arrival time = 10 seconds
- Zombies' HTTP packets inter-arrival time = 0.3 seconds
- Filtering is studied separately. No filtering is done with the DDoS simulation projects.
- DDoS attack lasts for 5 minutes, typical to real-life average DDoS attack duration.
- At the server, the datagram forwarding rate of the IP processor (queue) is reduced to 5000 packets/sec (from 10^6 p/s) to make the server slower
- At the server, the memory size of the IP processor (queue) is reduced to 8 MB (from 256 MB) to see the packets get dropped after queue is filled up.
- All the processing required for decision, and mitigation of DDoS attack is done by the centralized control station, not the victim server.

The first project was to be able to successfully simulate a small-scale TCP/IP network. This would present a smaller version of the client-server communication model that we would need later in the full-scale DDoS simulation which is illustrated in Figure 2.

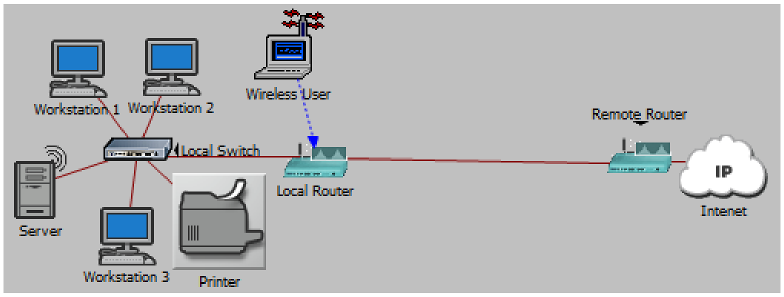


Figure 2: OPNET Project #1- Typical TCP/IP LAN with Router Connected to Internet

Once a simple communication model was established, several possibilities arose. Since ingress filtering and egress filtering are effective ways to block DDoS attacks, which usually use spoofed source IP addresses, packet filtering could be tested in OPNET [7][9][11].

VI. PROJECT #2- PACKET FILTERING BY PACKET TYPE

The goal of this project is to prove the concept of using firewall filtering to filter ingress traffic by traffic type. The idea is to use or expand this concept later to filter certain types of traffic packets from getting into the network.

Here, firewall filtering is done to demonstrate that certain types of traffic can be filtered out from getting into the network or a part of a huge network. The simulation lab required 3 different scenarios in order to compare the performance of the same (topological) network in the following three cases: there is no firewall in the network, there is a firewall in the network and it allows all traffic through, or there is a firewall in the network and it discards web traffic. Figure 3 shows the time-average of the HTTP page response-time (in seconds) for all three scenarios.

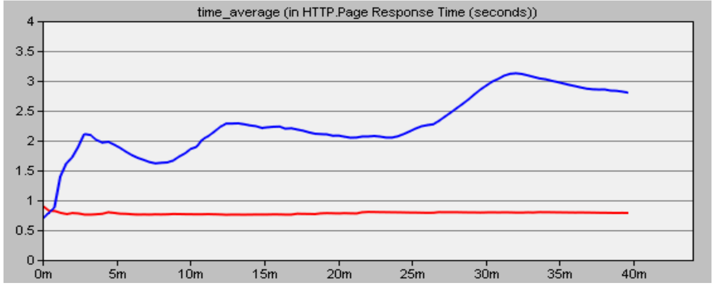


Figure 3: Result for OPNET Project #2, Firewall Filtering Ingress HTTP Traffic

As filtering is done in web traffic in the third scenario, only the first two scenarios allow web traffic to flow through the network. The absence of the third plot (belonging to the third scenario, when firewall filtering/blocking is used) indicates that the firewall blocked all the incoming web traffic in that scenario.

VII. OPNET PROJECT #3: PACKET FILTERING BY SOURCE ADDRESS

The goal of this project is to prove the concept of using firewall filtering to filter ingress traffic by traffic source. The idea is to expand this concept later to filter traffic from certain machines and thus prevent them from getting into the network. The simulation scenario makes use of TCP/IP network similar to the one shown in OPNET project #1 in section V.

Here, firewall filtering is done to demonstrate that certain traffic sources (nodes) can be filtered out from the rest of the network or a part of a huge network. The simulation lab required 2 different scenarios in order to compare the performance of the same (topological) network.

The first scenario makes use of a router to connect the LAN to the IP cloud. In this scenario, there are 5 attackers but no firewall in the network.

The second scenario makes use of a firewall to connect the LAN to the IP cloud. In this scenario, firewall filters the traffic sent by the attackers. Figure 4 shows the HTTP traffic sent (in packets/seconds) by one of the attackers in both the scenarios.

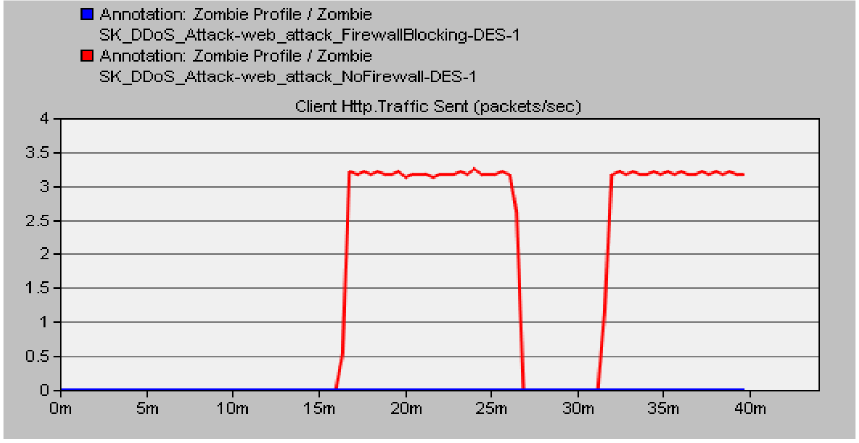


Figure 4: Result for OPNET Project #3- HTTP Traffic Filtered from the Attack Node

As filtering is done in by traffic source only in the second scenario, the first scenario allows web traffic to flow through the network as shown by the red plot in the figure. When firewall filtering/blocking is used, the firewall blocked all the incoming web traffic generated by the attack sources in that scenario.

VIII. OPNET PROJECT #4: MODELING OF THE INTERNET ARCHITECTURE

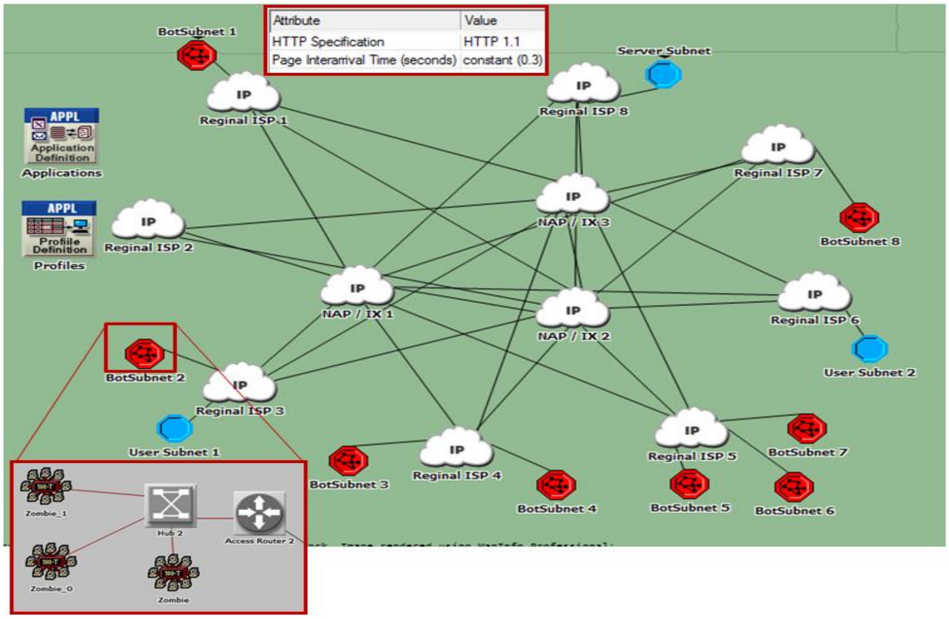
The goal of this project is to create a large scale network that represents a complex network of networks, called Internet. The idea is to make this model as realistic as possible so that DDoS attacks can be studied in simulation environment. Because realistic modeling of the Internet is crucial to being able to simulate DDoS attacks over internet for this paper, a good understanding and proper replica is needed to complete the work most effectively. The network developed has two legitimate users (clients) and one server, each connected to the internet through different ISPs. The simulation results for this network are presented and analyzed later.

IX. OPNET PROJECT #5: DDOS ATTACK IN INTERNET

The goal of this project is to simulate a DDoS attack by flooding the target server. The simulation scenario makes use of the internet architecture developed in OPNET project #4.

There are 8 different subnets where attackers are located (shown in the figure as “Bot Subnet” in red color). Each of these subnets comprise of three different LANs, each with 13 bots, making a total of 3*13*8 bots = 312 bots. The bots start to send attack traffic randomly at around 15 minutes and DDoS attack will last for 5 minutes.

Both the legitimate users and the bots send HTTP traffic to the server. However, the traffic generation is controlled using two different Application Configurations and Profile Configuration. A detailed illustration of this network can be found in Figure 5 and simulation results for this network are also presented and analyzed later.



Attribute	Value
HTTP Specification	HTTP 1.1
Page Interarrival Time (seconds)	constant (0.3)

Figure 5: OPNET Project #5-Eight Subnets (Total of 312 Bots) involved in a DDoS Attack

OPNET PROJECT #6: LINK FAILURES TO STUDY NETWORK RECOVERY

The goal of this project is to simulate the network recovery following a DDoS attack. The idea is to use results from this simulation as a benchmark to refer back when we successfully achieve all the mitigation steps: attack suspicion, communication, decision and response. The simulation scenario is built upon the scenario presented in OPNET Project #5.

This project uses link failure and link recovery sequence to cut off the links from where attack traffics are getting into the internet during the attack, and to restore them after the attack. A “Failure” configuration node is added automatically that can control all the links and nodes in the project. This configuration node fails (disconnects) all the links that connect the BotSubnets to their ISPs at exactly 1000 simulation seconds (about 2 minutes into the DDoS attack) and then recover at exactly 1800 simulation seconds (about 10 minutes after the DDoS attack stops).

There are some crucial assumptions made in this project. First, we are assuming that the defense mechanism is in place so that we do not have to worry about detecting and mitigating an ongoing attack. As stated earlier, mitigation of DDoS attack is beyond the scope of this paper. Second, by failing specific links close to the attackers at 2 minutes into the DDoS attack, we assume that all the attackers will be traced-back within 2 minutes from the start of the attack. Recent studies and publications have indicated that the attackers involved in a DDoS attack can be traced back within 20 seconds [11]. Third and most importantly, we assume that link fail and recover automatically. In our studies, we recommended that there exist a centralized control station that can monitor traffic flows. Simulation results for these networks are presented and analyzed next.

X. RESULTS AND ANALYSIS

This section shows results for the final three project simulations. Figure 6 shows the number of active TCP connection throughout the simulation. In the figure, the blue plot is for the first scenario, when no attackers are placed in the network. Since only two legitimate users exist to send packet to the server at any given time, the incoming packets get served quickly. Therefore the active TCP connection counts at the server fluctuated between 0 and 2.

The red plot is for the second scenario that consists of 312 bots. The bots send huge amount of attack traffic to the target server, randomly starting between 14th and 16th minutes. The server cannot handle the demand, and starts to get overwhelmed. Therefore, the red plot begins to swell from 14th minutes. The bots are programmed to stop randomly after about 4-6 minutes. Therefore, starting from around 18th minute, there is a fall in the red plot from 18th. By 20 minutes, almost all the bots stop sending attack packets. However, the plot indicates that the network recovers slowly, requiring roughly 12 more minutes to fully recover from the attack.

The green plot is for the third scenario, which uses link failure and link recovery sequence. When all the incoming attack traffic is blocked 2 minutes into the DDoS attack, there is only sporadic traffic coming in from legitimate users. Lower influx of traffic rapidly reduces the active TCP connection counts at the server and therefore the network recovers much faster, as shown by the steep downfall of the green plot.

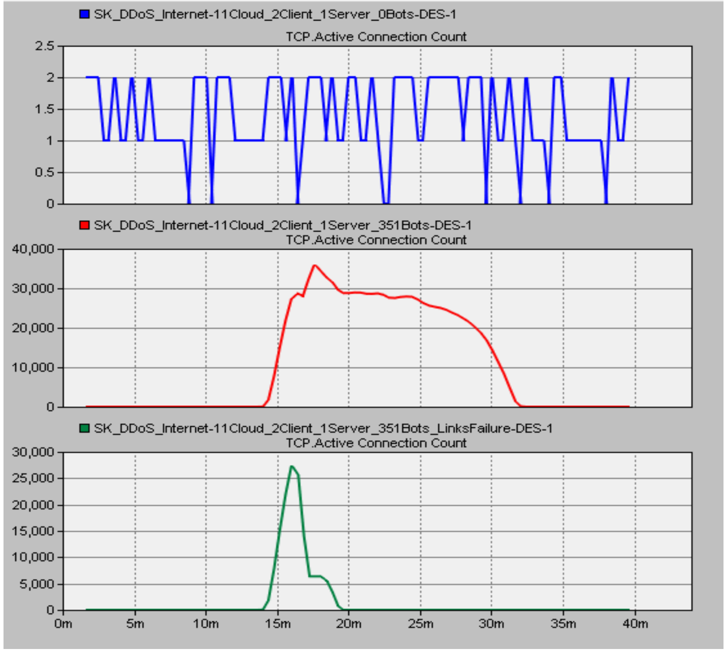


Figure 2: Results for OPNET Project #4-6-- Active TCP Connection Count at the Server

Figure 7 shows the results of the CPU Utilization at the server. It is critical to understand that the CPU utilization greater than 80% is considered menacing, and will account to increase in waiting time at the server's queue. The blue line which indicates the CPU utilization of the server for the first simulation (no attack) is well below 5%, and is therefore not an issue.

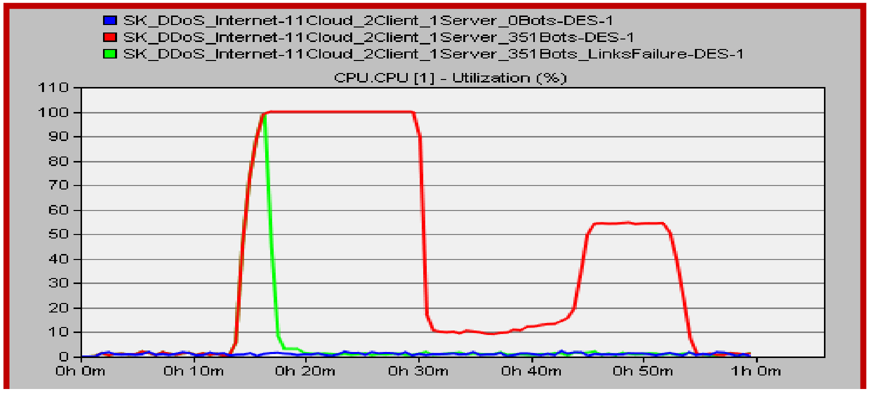


Figure 7: Results for OPNET Project #4-6— CPU Utilization at the Server

During the DDoS attack, a huge amount of traffic gets to the server and CPU utilization of the server increases dramatically to 100%, as shown by the red plot. The CPU utilization of the server remains 100% for about 15 minutes even though the attack lasted only for about 5 minutes. This is because a huge amount of incoming traffic is waiting for service in the queue of the server's IP processor and it takes some time to drain the fully consumed target server.

The green plot is for the link failure and link recovery sequence. When all the incoming attack traffic is blocked 2 minutes into the DDoS attack, there is only sporadic traffic coming in from legitimate users. Lower influx of traffic rapidly reduces the CPU utilization of the server and therefore the network recovers much faster, as shown by the steep downfall of the green plot.

Figure 8 shows the IP packets dropped (in packets/second) at the server throughout the simulation. The red line represents the IP packets dropped at the server for the second simulation (with attack). During a DDoS attack huge amount of traffic gets to the server, and start waiting for service in the queue of the server's IP processor, because the server's resources are fully utilized. The forwarding memory free size eventually drops down to 0 bytes during the attack, as the memory (server's queue) gets filled up. Once the memory is filled up, any more incoming traffic will get dropped at the server node. Therefore, we see packet getting dropped only few minutes after the attack begins and last a few minutes after the attack stops.

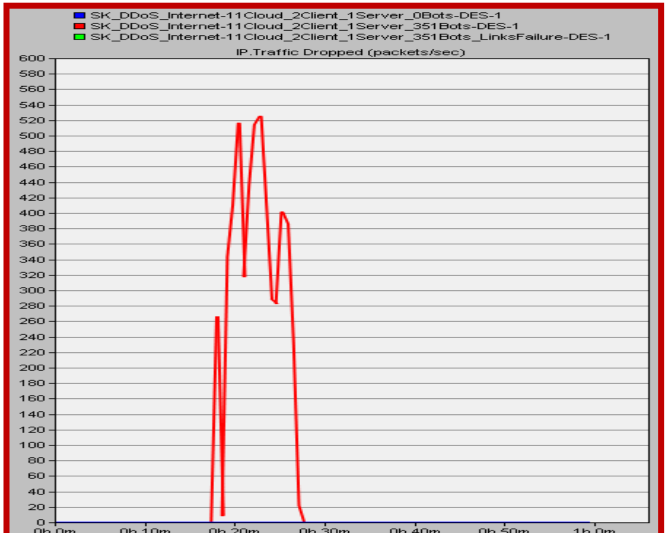


Figure 8: Results for OPNET Project #4-6— IP Packet Dropped at the Server

The absence of other two plots (belonging to the first scenario where no attack takes place, and the third scenario, when link failure is used) indicates that the packets did not get dropped when we disconnected the bots. In other words, all legitimate users were served when the attack traffics are not introduced into the network.

XI. CONCLUSION

Our simulation modeled a flooding attack by sending large numbers of TCP packets destined for the victim. The combined zombies managed to send over 8 MB worth of bogus traffic per second, more than enough to completely consume the victim's available bandwidth. Because of this, packets were buffered at the server. Once the available buffers were completely filled, packets destined for the victim were dropped. This project offers a starting point for future work in distributed denial of service research. By presenting the in-depth result analysis, this paper attempts to provide a benchmark to refer back to, for any future DDoS simulation studies.

XII. ACKNOWLEDGEMENTS

We would like to thank our advisor, Dr. Richard A. Dean, for his support leading to this paper and OPNET Technologies Inc. for providing software license to carry out the simulations.

XIII. REFERENCES

[1] Kalman K. K. Wan and Rocky K. C. Chang. "*Engineering of a Global Defense Infrastructure for DDoS Attacks*". IEEE 2002.

[2] Dave Lee. *Global internet slows after 'biggest attack in history'*. BBC News. 27 March, 2013. <http://www.bbc.co.uk/news/technology-21954636>

[3] *Dutchman arrested over huge web attack*. BBC News. 26 April, 2013. <http://www.bbc.co.uk/news/technology-22314938>

[4] John A. Hamilton, Jr., Ph.D., W. Chatam, and J. Rice. "*Using Simulation to Analyze Denial of Service Attacks*". Auburn University.

[5] "Security in Computing". 4th Edition (International Ed.). Charles P. Pfleeger and Shari Lawrence Pfleeger. Prentice Hall. 2006.

[6] Issariyakul, Teerawat; Hossain, Ekram. "*Introduction to Network Simulator NS2.*" New York: Springer; 2009.

[7] Paul C. Hershey, Ph.D. and Charles B. Solio, Jr., Ph.D. "*Procedure for Detection of and Response to DDOS Cyber Attacks on Complex Enterprise Systems*", IEEE 2012.

[8] Veronika Durcekova, Ladislav Schwartz and Nahid Shahmehri. *"Sophisticated Denial of Service Attacks Aimed at Application Layer"*, IEEE 2012.

[9] D. W. Gresty, Q. Shi, M. Merabti. "*Requirements for a General Framework for Response to Distributed Denial of Service*", 2001.

[10] Kalman K. K. Wan and Rocky K. C. Chang. "*Engineering of Global Defense Architecture for DDOS Attacks*", IEEE 2002.

[11] Shui Yu and Robin Doss. "*Traceback of DDOS Attacks Using Entropy Variations*", IEEE 2011.

[12] M.T. Goodrich, "*Probabilistic Packet Marking for Large-Scale IP Traceback*," IEEE/ACM Trans. Networking, vol. 16, no. 1, pp. 15-24, Feb. 2008.