A Novel High-Speed Architecture for Integrating Multiple DDoS Countermeasure Mechanisms Using Reconfigurable Hardware

Biet Nguyen-Hoang, Binh Tran-Thanh, Cuong Pham-Quoc, Nguyen Quoc Tuan, Tran Ngoc Thinh
Ho Chi Minh City University of Technology,
Vietnam National University, Ho Chi Minh City, Vietnam.
cuongpham@hcmut.edu.vn

***Abstract*—In this paper, we proposed a novel high-speed architecture to incorporate multiple stand-alone DDoS countering mechanisms. The architecture separates DDoS filtering mechanisms, which are algorithms, out of packet decoder, which is the basement. The architecture not only helps developers to give more concentration on optimizing algorithms but also integrate multiple algorithms to achieve more efficient DDoS defense mechanism. The architecture is implemented on reconfigurable hardware, which helps algorithms to be flexibly changed or updated. We implemented and experimented the system using NetFPGA 10G board with incorporation of Port Ingress/Egress Filtering and Hop-Count Filtering to classify IP spoofing packets. The synthesis results show that the system runs at 118.907 MHz, utilizes 38.99% Registers, and 44.75% BlockRAMs/FIFOs of the NetFPGA 10G board. The system achieves the detection rate of 100% with false negative rate at 0%, and false positive rate closed to 0.16%. The experimental results prove that the system achieves packet decoding throughput at 9.869 Gbps in half-duplex mode and 19.738 Gbps in full-duplex mode.**

***Index Terms*— Distributed Denial of Service (DDoS); FPGA; Hop-Count; Ingress; Egress.**

I. INTRODUCTION

The Internet is growing fast, and it has become an important mechanism to connect people and devices together. For that reason, the number of Internet users is increasing. As of Oct 24, 2015, there are more than 3.2 billion of users joined Internet [1], and it is increasing steadily. This increase will be a good chance for attackers to replicate malicious software (malware), steal personal user information and occupy computers for distributed denial of service (DDoS) attacks.

DDoS is a network attack method to prevent legitimate users from accessing network resources or services. The attacker performs DDoS attacks by consuming network's resources, or by consuming server's resources, or both of them. Attackers can perform attacks from one source (DoS), or from multiple sources (DDoS). Most of DDoS attacks use Internet Protocol (IP) address spoofing technique [2] that allows attackers to forge source IP address of a packet difference from its original address. Spoofer Project showed that 13.5% address space is spoofable [3]. The way routers route a packet is a vulnerability that attackers exploit to perform DDoS attacks. Network routers only check packets' IP destination address to make a routing decision, while source IP address is intact.

In this paper, we proposed a novel architecture to detect and defend against DDoS attack based on IP spoofing technique. The architecture consists of two main components: Base System and DDoS Filtering. The Base System takes responsibility to extract header and store raw packets while waiting for classifying result from DDoS Filtering component. DDoS Filtering component classifies packets based on the header received from the Base System. DDoS Filtering component can include multiple filtering modules. We implemented Port Ingress/Egress Filtering (PIEF) and Hop-Count Filtering (HCF) module to counter DDoS attacks based on theory from [4] and [5].

The main contributions of this paper are as follows:

- High-speed packet decoder: filtering modules are implemented in a Field Programmable Gate Array (FPGA) device. It takes advantage of hardware-based parallel processing, which is faster than software-based implementation.
- A novel model for DDoS's countermeasure mechanism: the proposed architecture separates packet decoder component, namely a basement, from Packet Filtering component, which implements algorithms. This architecture helps developers to implement filtering modules independently based on Packet decoder.
- The Combination of PIEF and HCF for countering DDoS attacks: both implemented filtering modules are DDoS defense mechanisms that prevent IP spoofing attacks. This combination not only consolidates DDoS countering mechanisms but also proves that multiple filtering mechanisms can be incorporated to prevent DDoS attacks.

The rest of the paper is organized as follows; Section II describes DDoS attack and defense mechanisms and related work. Section III presents our proposed DDoS countermeasure. The implementation and experimental results are discussed in section IV. Section V concludes the paper and introduces future work.

II. BACKGROUND AND RELATED WORK

In this section, we present DDoS background and countermeasure methods to prevent DDoS attacks.

A. Background

Attackers often employ computers or zombies controlled through malwares to form a botnet to perform a DDoS

attack. They are usually motivated by incentives such as financial/economical gain, revenge, ideological belief, intellectual challenge or cyberwarfare [2]. Attacks which are for financial gain are dangerous and hard to mitigate.

a. DDoS Attack Mechanisms

Zargar et al. [2] classified DoS/DDoS attacks into two categories: network/transport-level and application-level flooding attack. Network/Transport-level based flooding attacks are performed by exploiting vulnerabilities of layer 2 to layer 4 in the Open Systems Interconnection (OSI) network model to exhaust victim's network resources. Application-level based flooding attacks exploit application-level vulnerabilities, including protocols and application code, to exhaust victim's server resources.

Network/Transport-level based attacks are also categorized into 4 subcategories [2]; flooding attacks, protocols exploitation flooding attacks, reflection-based flooding attacks, and amplification-based flooding attacks. Flooding attacks often exhaust network resources by consuming bandwidth or overburdening network devices. In protocol exploitation attacks, attacker sends malformed packets, such as TCP SYN flood [6] [7] and TCP SYN/ACK flood [8], to confuse victim. In reflection and amplification based attacks, the attacker sends spoofed packets in which source address is victim's IP address to reflectors/amplifiers, then responses are sent to the victim and cause flooding (i.e., Smurf attack, Fraggle attack).

Application-level based attacks exploit vulnerabilities of application protocol and application code. Attackers often exploit stateless protocols for this kind of attack, such as DNS, NTP. DNS amplification DDoS had been researched [9] [10] and recorded an attack with 300Gbps [10] [11]. NTP amplification DDoS sets a new record with 400Gbps in 2014 [12] [13].

Attacker (i.e., Master) starts DDoS attacks by sending control message to bots (i.e., Agents) in a botnet to perform an attack. Attacker can control the botnet through Internet Relay Chat (IRC) or HTTP-based command.

b. DDoS Defense Mechanisms

DDoS defense mechanisms are classified into network-level and application-level [2]. Network-level based defense mechanism is deployed to mitigate DDoS attacks under network layers. It is also categorized into source-based, network-based, destination-based and hybrid mechanisms based on deployment location. PIEF method [4] can be deployed as a source-based or destination-based mechanism. In the destination-based mechanism, Management Information Base (MIB) [14] can be used to monitor traffic to detect DDoS attacks, HCF method [5] can filter out spoofed packet based on the number of routers a packet traversed. The paper [2] also introduces hybrid methods, such as Stop-It and Active Internet Traffic Filtering (AITF), which incorporate multiple components across network systems to counter DDoS attacks. Application-level based defense mechanism is deployed to detect application vulnerabilities attacks. CAPTCHAR [15] is an application-based method to differentiate DDoS flooding bots from the human. It helps server to classify bot-based packets and filter it.

B. Related Work

This section introduces several methods for detecting spoofed packets, such as PIEF and HCF.

Ferguson et al. [4] proposed Port Ingress/Egress Filtering method to filter spoofed packets. The ingress and egress name depends on its deployment position. Ingress filter is deployed to filter inbound traffic. If an incoming packet is spoofed, it is blocked. Egress filter filters outbound traffic to ensure that malicious packets will never leave internal network.

Wang et al. [5] proposed a method named Hop-Count Filtering to filter spoofed packets based on the number of hops that packets traverse before arriving at the victim. Each packet traveling on the network has its own Time-To-Live (TTL). When a packet traverses a router (hop), its TTL value is decreased by one before forwarding to next hop. Therefore, packet's hop-count could not be spoofed. Hop-count value is calculated by comparing initial TTL to final TTL value when the packet arrives at the destination. Packets whose TTL is equal to zero are dropped. While not being attacked, IP address and hop-count are collected and stored in IP-to-Hop-Count (IP2HC) tables. When DDoS attack occurs, packets' IP and hop-count value are compared to IP2HC records. If it does match an IP2HC record, it is a legitimate packet; otherwise, it is a spoofed packet and is dropped. The paper claimed that HCF can identify 90% of spoofed packets.

Wang et al. [16] also proposed a distributed HCF (DHCF) model, which is implemented at intermediate routers. This method not only protects host but also protects the intermediate network from malicious packets and traffic congestion. Experimental results showed that DHCF achieved better performance than conventional HCF but maintained user's access.

Ayman et al. [17] proposed an upgraded version of HCF, by storing multiple hop-count values according to multiple routes. This modified HCF method can increase true positive rate because a packet's hop-count values may vary if it travels through multiple routes. However, this method suddenly increases false negative rate, because it increases the chance to the attacker to bypass the detector.

Maheshwari et al. [18] combined probabilistic and round trip time in Distributed Probabilistic HCF-Round trip time (DPHCF-RTT). Packets are checked once by intermediate DPHCF-RTT routers (nodes) and then they are forwarded to the victim. The larger number of intermediate routers implemented, the higher detection rate of malicious packets is. The paper claimed that detection rate is up to 99.33%.

III. METHODOLOGY

In this section, we describe our proposed DDoS countering architecture as shown in Figure 1. The architecture consists of two main components: Base System and DDoS Filtering component.

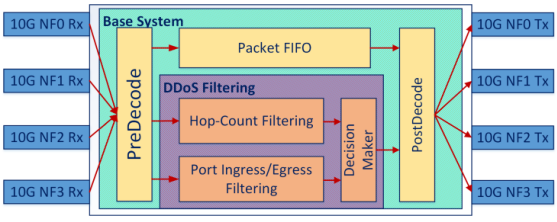

Figure 1: The architecture of DDoS countering system

A. Base System

a. PreDecode

This module decodes and extracts the IP header of incoming packets. Packets' header is transferred to filtering modules for classifying. Raw packets are stored in Packet FIFO while waiting for classifying results from filtering modules. Those packets' header includes source IP, destination IP and TTL value.

b. Packet FIFO

There are two approaches to process packets. The first approach, a packet is de-encapsulated into header and payload. Then, the header is sent to filtering modules to classify. Finally, if the packet is legitimate, header and payload are encapsulated and the packet is sent out to the network. If the packet is classified as DDoS, the packet is not encapsulated. This is time-consuming approach because encapsulation takes time. The second approach is to use a buffer to store full raw packets. This approach helps to reduce system latency. In this work, we implemented the second approach and named Packet FIFO.

c. *PostDecode*

The PostDecode module receives packets from the Packet FIFO module and waits for decisions from the Decision Maker module. The PostDecode module determines whether the packets are forwarded or dropped depending on the feedback of DDoS Filtering modules. If packets are legitimate, they are sent out to the network. Otherwise, they are dropped.

B. DDoS Filtering

In Section 2.4, several DDoS defense mechanisms are discussed. Each of those approaches only counters a specific DDoS attack. Therefore, those mechanisms do not completely classify DDoS attack packets in stand-alone mode. In this section, we present a combination of PIEF and HCF modules. This combination helps DDoS Filtering to classify packets deeper and wider than the stand-alone mode.

Table 1
Global and Specialized Address Blocks

Address Block	Present Use
0.0.0.0/8	"This" network
10.0.0.0/8	Private-Use Networks
127.0.0.0/8	Loopback
169.254.0.0/16	Link Local
172.16.0.0/12	Private-Use Networks
192.0.0.0/24	IETF Protocol Assignment
192.88.99.0/24	6to4 Relay Anycast
192.168.0.0/16	Private-Use Networks
198.18.0.0/15	Network Interconnect Device Benchmark Testing
198.51.100.0/24	TEST-NET-2
203.0.113.0/24	TEST-NET-3
224.0.0.0/4	Multicast
240.0.0.0/4	Reserved for Future Use
255.255.255.255/32	Limited Broadcast

a. Port Ingress/Egress Filtering

In computer networking, ingress filtering is a technique used to make sure incoming packets are actually from their original network. Any router that implements ingress filtering method checks source IP address of traversing packets. The router drops the packets if its source IP address is not in the range of address that the router's interface is connecting. Table 1 shows IP address blocks for special use. Those IP addresses do not either appear or exist on the Internet as usual. Therefore, they should also be blocked in PIEF module.

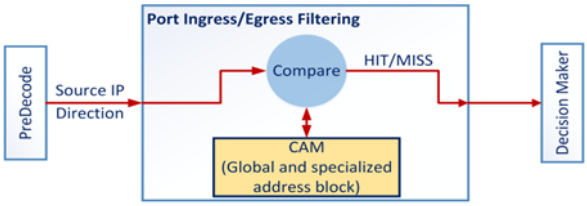

Figure 2: The architecture of Port Ingress/Egress Filtering module

Figure 2 describes a diagram of PIEF module. When the packet's source IP address is sent to PIEF, The PIEF searches the IP in Content Addressable Memory (CAM). If a MISS signal returns (no record found in CAM), the packet is legitimate. Otherwise, the packet is illegitimate. After that, PIEF forwards MISS/HIT signal to Decision Maker.

b. Hop-Count Filtering

Although DDoS attackers can forge any field in the packets' header, they cannot falsify the number of hops that a packet traversed to reach its destination. The number of traversed hops of a packet, named hop-count, is calculated by subtracting the final TTL from the initial TTL. TTL is an 8-bit field [19] in the IP header which originally introduced to specify the maximum lifetime of an IP packet on the Internet. The final TTL is the value when the packet reaches the destination. The initial TTL values are 30, 32, 60, 64, 128, and 255 according to OS where the packet is made. Figure 3 shows the complete HCF algorithm.

```
for each packet:
    extract the final TTL Tf and IP Address S;
    infer the initial TTL Ti;
    compute the hop-count Hc = Ti – Tf;
    index S to get the stored hop-count Hs;
    If (Hc # Hs)
        packet is spoofed;
    else
        packet is legitimate;
```

Figure 3: The algorithm of Hop-Count Filtering module

IP2HC table is important because it helps HCF to detect whether a packet is spoofed or not. CAM is used to store IP address in most FPGA-based system networking because of its fast query response. However, CAM can return index and an HIT/MISS signal only. Therefore, We store hop-count in Register Array instead of CAM. Furthermore, the incorporation of CAM and Register Array improves query time. The index of CAM and Register Array is direct one-to-one mapping. If source IP address of a packet is stored in CAM at index k, the returned value of Register Array at index k is the hop-count value of that packet. Table 2 show how we implement IP2HC. When we look for IP address 134.170.188.221, CAM returns an index of 1. Based on that index, Register Array returns 10 as a hop-count value of the IP address.

Figure 4 describes the architecture of the HCF module. The module receives packets' source IP and final TTL from

the PreDecode module. This module calculates the hop-count value, lookups IP address in CAM and gets the hop-count value stored in the Register Array. If the returned hop-count is equal to the calculated hop-count, the packet is legitimate. Otherwise, the packet is spoofed. For the first time the packets, whose source IP do not exist in CAM, come to the system, CAM returns a MISS signal and stores the packets' IP and hop-count value. However, the initial TTL can be forged. It means that the packet is spoofed, but HCF could not recognize that. That is the main disadvantage and limitation of HCF mechanism. Therefore, we consider combining PIEF and HCF.

Table 2
IP to Hop-Count table

CAM		Register Array	
Index	IP Blocks	Index	Hop-count value
1	134.170.188.0/24	1	10
2	69.171.230.0/24	2	20
...	...	...	...
n	216.58.221.0/24	n	8

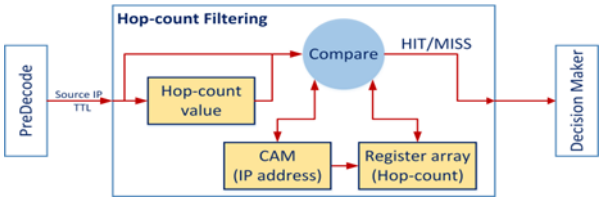

Figure 4: The architecture of Hop-Count Filtering module

c. Decision Maker

Decision Maker module gives a decision to the PostDecode module depended on the outputs of the PIEF and HCF modules. Drop signal alerts the PostDecode module if either PIEF or HCF realizes a sign of DDoS attack. Otherwise, bypass signal is sent to PostDecode to allow the packet to go to the network.

C. Experiments

In this section, we implement and test the system using a NetFPGA 10G board.

a. System Implementation

The NetFPGA 10G [20] board is used to develop DDoS countermeasure architecture. The board includes four SFP+ ports, Xilinx Virtex-5 TX240T. Four SFP+ ports are suitable to build network applications. Besides, Xilinx Virtex-5 TX240T provides powerful hardware to handle huge traffic on the Internet. We use HDL to develop the following modules:

- The PreDecode module: This module receives raw packets from network interface via AIX interface [21]. After that, this module provides the decoded fields to DDoS Filtering modules and forwards the original packets to the Packet FIFO module.
- The Packet FIFO module: This module functions as a FIFO, which is provided by Xilinx. The configuration of Packet FIFO is 256-bits in width and 1024 entries in depth. With that configuration, the Packet FIFO module stores minimum 21 packets in 1500 bytes of size and maximum 512 packets in 64 bytes of size.
- The PostDecode module: The functionality of this module is to send packets to 10G NIC TX through AXI interface. The PostDecode module receives control signals from the Decision Maker module. Depending on the signals, the PostDecode module decides whether the packets are dropped out from the system or forwarded to the network.
- The Port Ingress/Egress Filtering module: This module consists of two components: CAM and Comparator. The CAM consists of sixteen ranges of IP address as describe in Figure 3. The Comparator compares the IP address of incoming packets with CAM.
- The Hop-Count Filtering module: This module consists of three components: CAM, Comparator, and Register Array as describe in Figure 4. According to the limitation of NetFPGA 10G board's resources, we only build two versions of CAM, 128 and 256 entries.
- The Decision Maker module: This module gives final decisions to the Base System component based on the decision signals from the PIEF and HCF modules. Either the PIEF or HCF sends DROP signal, the Decision Maker instantly turns on DROP signals to the PostDecode module. Otherwise, the Decision Maker sends BYPASS signals to the PostDecode module.

The system is synthesized by ISE 13.4 without any manual optimization. Table 3 shows hardware resources usage of the system. The system, which implements 128 entries of CAM, runs at 118.907 MHz, utilizes 38.99% Registers, and 44.75% BlockRAMs/FIFOs.

Table 3
Device utilization summary of the system

Module	Max Clock Frequency (MHz)	Registers	BlockRAM/FIFO (KBytes)
System	118.907	58,384	5,220
PreDecode	262.522	1061	36
Packet FIFO	170.023	423	468
PIEF	247.452	111	108
HCF	120.043	1933	4,608

b. Experimental Setup

To measure the accuracy and throughput of the system, we deployed a testing model as shown in Figure 5. NetFPGA 10G boards and Open Source Network Tester (OSNT) [22] were used for both throughput and accuracy testing model. OSNT is a flexible tool. It can generate and capture packets of any size at the line-rate speed of 10Gbps. In the accuracy testing model, three computers functioned as zombies to attack the system, another one was a normal user.

The throughput testing model was used to test decoding speed. We prepared TCP/UDP packets with various sizes from 64 to 1500 bytes. The packets were sent out at the maximum speed of OSNT Generator. We used OSNT Monitor to measure the throughput.

In the accuracy testing model, we tested the combination of two filters: PIEF and HCF. We prepared TCP/UDP packets with various sizes. Some of those packets were real and collected from the Internet to test HCF. Some packets were generated to test PIEF. Classified packets were captured to evaluate and find out the detection rate (DR), false positive

rates (FPR) and false negative rates (FNR). The NetFPGA_01 classifies packets, bypasses legitimate packets to NetFPGA_02 and forwards spoofed packets to NetFPGA_03.

D. Experimental Results

Figure 6 shows throughput values of the experimental system. The vertical axial shows throughput value in Gbps. The horizontal axial shows packet size of test cases in Byte. For each test case of packet size, the left, middle and right columns show packet decoding speed in half-duplex mode, full-duplex mode, and the packet generating speed of OSNT respectively. Based on the results in Figure 6, the throughput of the system nearly reaches the maximum speed of packet generator in half-duplex mode and achieves 9.869 Gbps. In full-duplex mode, the system in stable condition achieves twice the throughput of half-duplex mode and up to 19.738 Gbps, except test case of 64-Byte packets. The low throughput value in the full-duplex mode of the 64-Byte test case may be due to the experience of the developer in dealing with small packet size.

Table 4 shows the statistical values of accuracy test of the system. The results were collected by evaluating captured classified packets. We repeated the test case twice in each version of CAM. In all test cases, the DR is up to 100%, the FNRs are 0%. In 128 entries CAM test cases, FPRs are 0%. In 256 entries CAM test cases, FPRs are close to 0.16%.

We also did statistic calculation based on packet size. Figure 7 shows the DR, FPR, and FNR of the experimental system based on packet size and CAM version. The packets are classified into 6 groups: 0-64 Bytes, 64-128 Bytes, 128-256 Bytes, 256-512 Bytes, 512-1024 Bytes and 1024-1500 Bytes. In all test cases, the DRs are 100%, FNRs are 0%. In 128 entries CAM version, the FPRs are 0%. In 256 entries CAM version, the FPRs vary, but they are downtrend on uptrend of the packet size; the values are 0.33%, 0.74%, 0.19%, 0.17%, 0.13%, 0.12% according to the increase of packet sizes.

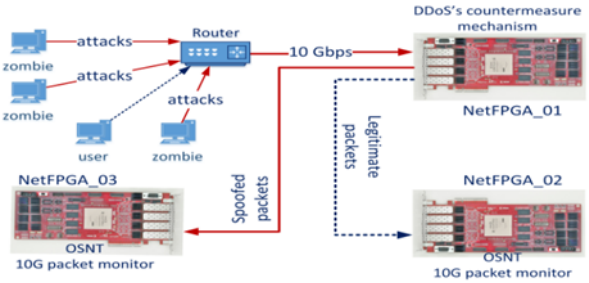

Figure 5: The accuracy testing model

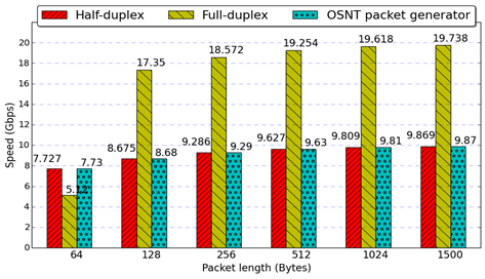

Figure 6: The throughput experimental results

Table 4
The packet classification statistic

CAM version	Test number	Expected result		Classified result	
				Legitimate	Spoofed
128 entries	1	Legitimate	286,162	286,162	0
		Spoofed	13,838	0	13,838
	2	Legitimate	286,162	286,162	0
		Spoofed	13,838	0	13,838
256 entries	1	Legitimate	266,757	266,319	438
		Spoofed	33,243	0	33,243
	2	Legitimate	266,757	266,285	472
		Spoofed	33,243	0	33,243

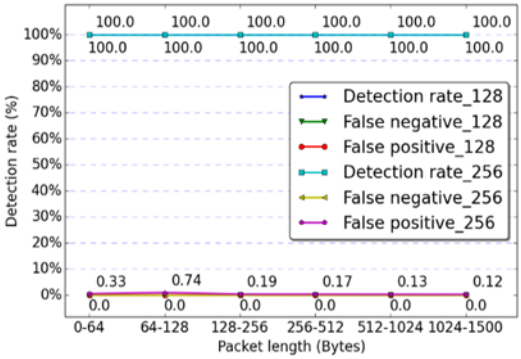

Figure 7: The packet classification statistic ratio

IV. CONCLUSIONS

In this paper, we proposed a novel architecture to defend against DDoS attacks using reconfigurable hardware. The architecture, which separates algorithms from the base system, helps developers to focus on optimizing filtering modules. The Base System component provides high-speed packet decoder and helps reducing system latency. Our DDoS countermeasure mechanism combines PIEF and HCF modules. With our approach, the packet decoding speed of the system reaches to 9.869 Gbps in half-duplex mode and 19.738 Gbps in full-duplex mode. Moreover, the combination of PIEF and HCF improves the DR up to 100%, higher than 90% of [5] and 99.33% of [18], with FNR close to 0% and FPR close to 0.16%.

ACKNOWLEDGEMENT

This research is funded by Vietnam National University Ho Chi Minh City (VNU-HCM) under grant number C2015-20-09. We would like to thank Nguyen Bao Quoc, Tran Thi Thuy Chau, and Do Minh Chien for assistance and incorporation in implementing and experimenting the system.

REFERENCES

[1] Internet Live Stats: retrieved October, 25 2015 from http://www.internetlivestats.com/.

[2] Zargar, S.T. and Joshi, J. and Tipper, D. 2013. A Survey of Defense Mechanisms against Distributed Denial of Service (DDoS) Flooding Attacks. *IEEE Communications Surveys Tutorials*. 15(4): 2046-2069.

[3] Spoofer Project: State of IP Spoofing: retrieved January, 2015 from http://spoofer.caida.org/summary.php.

[4] P, Ferguson and D, Senie. 2000. Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing. Internet RFC2827.

[5] Wang, Haining and Jin, Cheng and Shin, Kang G. February, 2007. Defense against Spoofed IP Traffic Using Hop-Count Filtering. IEEE/ACM *Transaction On Networking*. 15(1): 40-53.

[6] Haining Wang and Danlu Zhang and Shin, K.G. 2002. Detecting SYN flooding attacks. INFOCOM 2002. *Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings*. 3: 1530-1539.

[7] Wei Chen and Dit-Yan Yeung. 2006. Defending Against TCP SYN Flooding Attacks under Different Types of IP Spoofing. *International Conference on Networking, International Conference on Systems and International Conference on Mobile Communications and Learning Technologies*. 38-43.

[8] Mirkovic, J. and Reiher, P. 2004. A Taxonomy of DDoS Attack and DDoS Defense Mechanisms. ACM SIGCOMM *Comput. Commun.* Rev. 34(2): 39-53.

[9] Kambourakis, G. and Moschos, T. and Geneiatakis, D. and Gritzalis, S. 2007. A Fair Solution to DNS Amplification Attacks. *The 2nd International Workshop on Digital Forensics and Incident Analysis*. 2007. 38-47.

[10] Fachkha, C. and Bou-Harb, E. and Debbabi, M. March, 2014. Fingerprinting Internet DNS Amplification DDoS Activities. *The 6th International Conference on New Technology, Mobility and Security* (NTMS). 1-5.

[11] When spammers go to war: Behind the Spamhaus DDoS: retrieved March, 2013 from http://arstechnica.com/security/2013/03/when-spammers-go-to-war-behind-the-spamhaus-ddos/.

[12] Arbor Network. 2015. Worldwide Infrastructure Security Report. *Technical Report.* 10: 1-128.

[13] Technical Details Behind a 400Gbps NTP Amplification DDoS Attack: retrieved February, 2014 from https://blog.cloudflare.com/technical-details-behind-a-400gbps-ntp-amplification-ddos-attack/.

[14] Cabrera, J.B.D. and Lewis, L. and Xinzhou Qin and Wenke Lee and Prasanth, R.K. and Ravichandran, B. and Mehra, R.K. 2001. Proactive detection of Distributed Denial of Dervice attacks using MIB traffic variables-a feasibility study. IEEE/IFIP International Symposium on Integrated Network Management Proceedings. 609-622.

[15] Ahn, L. and Blum, M. and Hopper, N.J.and Langford, J. May, 2003. CAPTCHA: Using Hard AI Problems for Security. *Advances in Cryptology* — EUROCRYPT. 294-311.

[16] Wang, X. and Li, M. and Li, M. December, 2009. A Scheme of Distributed Hop-Count Filtering of Traffic. *IET International Communication Conference on Wireless Mobile and Computing* (CCWMC 2009). 516-521.

[17] Ayman, M. and Imad, E. and Ayman, K. and Ali, C. May, 2014. IP Spoofing Detection Using Modified Hop Count. IEEE *28th International Conference on Advaned Information Networking and Applications*. 512-516.

[18] Maheshwari, R. and Krishna, C.R. and Brahma, M.S. February, 2014. Defending network system against IP spoofing based distributed DoS attacks using DPHCF-RTT packet filtering technique. *International Conference on Issues and Challenges in Intelligent Computing Techniques* (ICICT). 206-209.

[19] Internet Protocol: retrieved 2012 from https://tools.ietf.org/pdf/rfc791.pdf.

[20] NetFPGA 10G: retrieved January, 2013 from http://netfpga.org/site/#/systems/3netfpga-10g/details/.

[21] AXI Reference Guide: retrieved January, 2012 from http://www.xilinx.com/support/documentation/ip-documentation/axi-ref-guide/v13-4/ug761-axi-reference-guide.pdf.

[22] The Open Source Network Tester: retrieved 2012 from http://osnt.org/.