

# Analysis of UDP DDoS Cyber Flood Attack and Defense Mechanisms on Windows Sever 2012 and Linux Ubuntu 13

Kiattikul Treseangrat, Samad Salehi Kolahi, Bahman Sarrafpour  
Department of Computing, Unitec Institute of Technology  
re\_vision@live.com  
skolahi@unitec.ac.nz

**Abstract**— Distributed Denial of Service (DoS) attacks is one of the major threats and among the hardest security problems in the Internet world. In this paper, we study the impact of a UDP flood attack on TCP throughputs, round-trip time, and CPU utilization on the latest version of Windows and Linux platforms, namely, Windows Server 2012 and Linux Ubuntu 13. This paper also evaluates several defense mechanisms including Access Control Lists (ACLs), Threshold Limit, Reverse Path Forwarding (IP Verify), and Network Load Balancing. Threshold Limit defense gave better results than the other solutions.

**Keywords:** Cyber Security, UDP DDoS Attack

## I. INTRODUCTION

The Internet has become increasingly important to current society; it has revolutionized our way of communication, doing business, and it has made information publicly and easily accessible. However, with all the advantages of the Internet, there are also some disadvantages. There is no absolute security in the Internet world, and the hackers can use the Internet to launch different types of attacks on a victim, one of which is known as a Distributed Denial of Service (DDoS) attack.

A DDoS Attack is one of the most common and major threat to the Internet in which the goal of the attacker is to consume computer resources of the victim, usually by sending a high volume of seemingly legitimate traffic requesting some services from the victim. As a result, it creates network congestion on the way from a source to the target, thus disrupting its normal Internet operation [1].

In particular, a UDP flood attack occurs when an attacker crafts numerous packets to random destination ports on the victims system. The victim system, on receipt of the UDP packet would respond with appropriate ICMP packets, if the port is closed. A very large number of packet responses would slow down the system or crash [2].

In this paper, we evaluate the impact of a UDP flood attack on a web server with the new generation of Windows or Linux platforms, namely, Windows Server 2012, and Linux Ubuntu 13. This paper also evaluates the existing defense mechanisms such as Access Control Lists [3], Threshold Limit [4], IP Verify [5], and Network Load Balancing [6].

ACLs, Threshold Limit and IP Verify techniques are implemented on the routers, denying unwanted traffic entering the network. ACL is configured to stop the attack by blocking all private IP addresses since these addresses cannot be used on the Internet [3]. Threshold Limit stops the attacks by limiting the traffic rate up to the threshold

for all incoming traffics, in this study, 10,000 packets per second. IP Verify technique enables routers to verify the reachability of source IP addresses before they can enter the network [4]. If the source IP address is not valid, the packet is dropped. Network Load Balancing technique, on the other hand, is implemented on individual servers. It does not stop unwanted traffic entering the network; it can only reduce the impact of attacks by balancing the attack traffic to an additional server using different paths and cables.

The organization of this paper is as follows. In the next section, the related work of DDoS Attacks is discussed. Section three covers the experimental setup and hardware specification. Section four covers information regarding the traffic measurement and data generating tools. Section five covers the evaluation of a UDP flood attack and defenses, and the last sections include the conclusions and future works.

## II. RELATED WORK

Analysis and comparison of DDoS Attack and defense mechanisms on different operating systems has been conducted by a number of researchers.

In 2006, Pack and colleagues [7] investigated the efficiency of Access Control List against the DDoS Attack. The result shows that the number of ACL rules affects the collateral damage (legitimate traffic was dropped unintentionally). With 5 ACL rules, the number of the collateral damage was 45%. However, this number significantly reduced to 15% if 50 ACL rules were used.

In 2009, Lu and colleagues [8] investigated the impact a UDP flood attack on the system by using metrics such as packet loss rate, delay, and jitter. The testbed consists of 9 routers and 14 computers with Intel Celeron 2.1 GHz and 512 KB memory running Linux. Iperf was a primary tool used to generate UDP traffic at 10, 15, 20 and 30Mbps. The result shows that without the attack there was no packet loss, and the delay jitter value was 32.3%. During a UDP flood attack, however, the number of packet loss went up to 14.08% while the jitter slightly decreased to 29.7%.

In 2009, Rui and colleagues [9] conducted a study of DDoS prevention based on IP Verify and Threshold Limit. The simulation program in this study was .net 2005 running on Windows Server 2003 system and the total number of IP addresses tested was 12,960,000 IP addresses.

In 2011, Subramani [4] conducted an experiment on TCP and a UDP flood attack and proposed 2 defense

mechanisms namely Access Control Lists and Threshold Limit. The results show that without the attack, the average response time of the server was 0.834 milliseconds while during the attack this number increased to 8.782 milliseconds. After using Access Control Lists, the average response time went down to 1.093 milliseconds, and it reduced to 6.985 milliseconds when using Threshold Limit.

In 2012, Kaur and colleagues [10] conducted an experiment on DDoS Attack using a DETER testbed. The network in this experiment consisted of three computers: an attacker computer, legitimate computer, and FTP server. The purpose of this research was to study the impact of the user throughput between computer nodes during a UDP flood attack. Traffic result shows that the average bandwidth before the attack was around 75Kbps while during the attacks, the average bandwidth has raised around 130Kbps.

In 2014 [11], we studied the TCP SYN DDoS attack and defense prevention mechanisms. We compared various defense mechanisms for preventing potential TCP SYN DDoS attacks. Router based TCP Intercept is found to provide the best defense while Anti DDoS Guardian gave the worst defense.

There has been no work done on testing performance and defense mechanisms between Windows Server 2012 and Linux Ubuntu 13. The lack of available research on impact of DDoS Attack on new generation of Windows and Linux platforms, and the need to develop suitable solution to address the rising cases of DDoS attacks on the computer networks were the main motivation behind this paper.

### III. EXPERIMENT SETUP

The test-bed diagram for site to site is displayed in Figure 1. The test-bed hardware setup remained constant for all experiments conducted. The only exception was the Networking Load Balancing (NLB) in which additional server was added to the switch to configure an IP cluster used as a “shared” IP address between two servers. By using this “shared” IP address, a client automatically connects to the server that has the higher priority first. In this study, the 50:50 rule was used, therefore, when a large number of attack packets entered the network, NLB shared the traffic to both server equally.

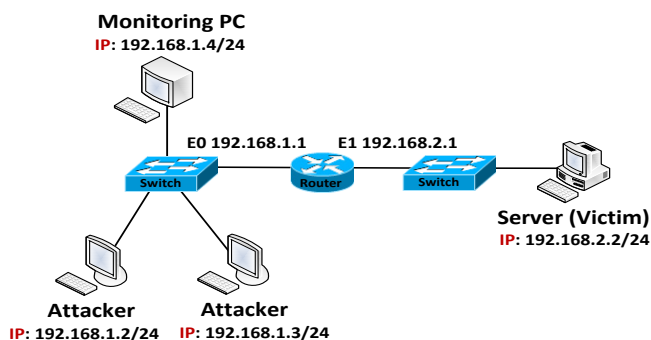


Figure 1: Network testbed.

The network was setup through a direct gigabit Ethernet connection using a standard category 5e cabling between workstations. The router with a 100 Mbps port was used to separate two networks, and used to monitor incoming and

outgoing traffic between networks. There were four types of workstations in the test-bed: Two workstations will act as attackers, one will act as a victim, and another one is used as a monitoring machine.

The workstations where the attackers perform have BackTrack5 R3 installed while the victim machine has Windows server 2012 or Linux Ubuntu 13 installed. The monitoring PC in which Windows 8 installed is where the different varieties of monitoring tools installed to gather data and perform the network testing analysis

The hardware benchmark comprised of an Intel® Core™ i5 2.80 GHz processor with 8.00 GB RAM for the efficient operation of operating systems, Cisco 2811 and Cisco SG 200 were chosen as the network connection devices.

### IV. DATA GENERATION AND TRAFFIC MEASUREMENT TOOL

TCPing [12] was the primary tool used to investigate the latency of the web server during the attack. Latency is a measure of time delay experienced in a system. By using TCPing we can measure the response times and hence we have calculated the latency of the victim computer.

Iperf [13] was selected as the tool to measure the user throughput and packet loss during the attack. Iperf has a client and server functionality, and can measure the throughput between the two ends, either unidirectional or bi-directionally. It is open source software and runs on various platforms including Linux, and Windows.

Hping3 [14] was chosen as an attacker generator, which is a built-in tool that is offered with Linux Back Track R3. HPing3 allows users to generate different types of DDoS attacks including UDP, TCP, and Smurf attack.

Webstress Server Tool [16] was used to generate legitimate traffic. It is software for load and performance testing of a webserver. Webstress Server Tool is designed to simulate multiple users accessing to a website.

All performance evaluation tests were run for 5 minutes, which generated the attack traffic at approximately 3 million packets per run. The attack rate was set to 17000 packets per second while the packet size was 512 Bytes per packet. The legitimate traffic was generated by Webstress Server Tool, which generate the connection request from users to the webserver assuming on average 10 users per second. To ensure high data accuracy, each test was repeated at least 30 times and data average and runs continued until standard deviation of results was below 0.07% of the average.

### V. EXPERIMENTAL RESULTS

The experiments were conducted to evaluate and compare TCP throughputs, round-trip time and CPU utilization before and during the attack on a web server with Windows Server 2012 or Linux Ubuntu 13. This section also evaluates four defense mechanisms, namely, Access Control Lists, Threshold Limit, IP Verify, and Network Load Balancing.

#### A. Impact of a UDP flood attack on Windows Server 2012 and Linux Ubuntu 13.

Figure 2 presents the TCP throughput results of Windows Server 2012 and Linux Ubuntu 13. On the whole, Linux Ubuntu 13 outperformed Windows Server 2012 in terms of throughput values before and during the attack.

The result shows that the TCP throughput value before the attack on the Linux platform was constant at 94 Mbps, which was higher than Windows at about 85 Mbps. During the attack, the TCP throughput value on Linux platform significantly dropped to 0.45 Mbps, while the TCP throughput on Windows reduced to 0.24Mbps.

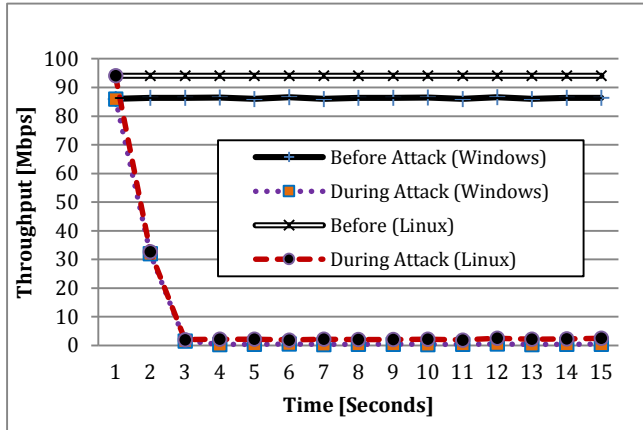


Figure 2: TCP throughput comparison between Linux Ubuntu 13 and Windows Server 2012

A plausible speculation that could explain the reason why Linux Ubuntu 13 outperformed Windows 2012 is probably due to the way kernel network buffers are allocated and used by Linux platforms [15]. That is, Linux has a pre-allocation of fixed-sized memory buffers so that when a network application transmits data, these buffers are used to avoid the overhead associated with buffer allocations.

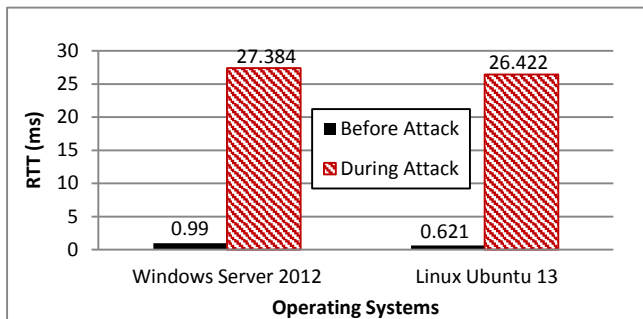


Figure 3: Comparison of RTT results between two operating systems before and during attack.

Figure 3 shows the round-trip time comparison between Windows Server 2012 and Linux Ubuntu 13. On the whole, the result shows that Microsoft Server 2012 had higher delay values than Linux Ubuntu 13. Without the attack, the average RTT of Windows platform was 0.99ms, while the average RTT of Linux platform was 0.62ms. During the attack, the RTT of Windows Server 2012 went up significantly from 0.99ms to 27.38ms, while the RTT of Linux Ubuntu 13 increased from 0.62ms to 26.42ms.

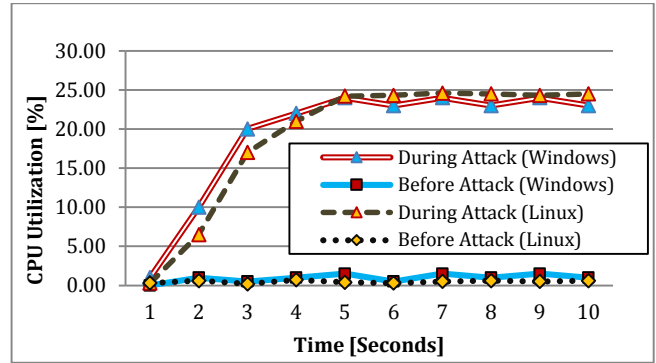


Figure 4: Comparison of CPU Utilization between two operating systems before and during attack.

Figure 4 illustrates the CPU utilization before and during a UDP flood attack. In terms of Windows Server 2012, the CPU usage before the attack was constant at 1% to 2%. During the attack, however, the CPU utilization went up approximately 10% within 2 seconds and increased to 22% in 4 seconds. Afterwards, it fluctuated between 19% and 24%.

In terms of Linux Ubuntu 13, the result shows that the CPU usage before the attack was slightly lower than for Windows, which was around 0.3 to 0.7%. During the attack, Linux Ubuntu 13 demonstrated the better performance in terms of stability; the CPU utilization increased at approximately 6% within 2 seconds. Afterwards, it remained steady at 23.5% to 24.9%.

### B. Evaluation of DDoS defenses on Windows Server 2012 and Linux Ubuntu 13

Figure 5 shows the impact of the UDP flood attack on TCP throughputs using Windows Server 2012 after using defenses, namely, ACLs, Threshold Limit, IP Verify, and Network Load Balancing.

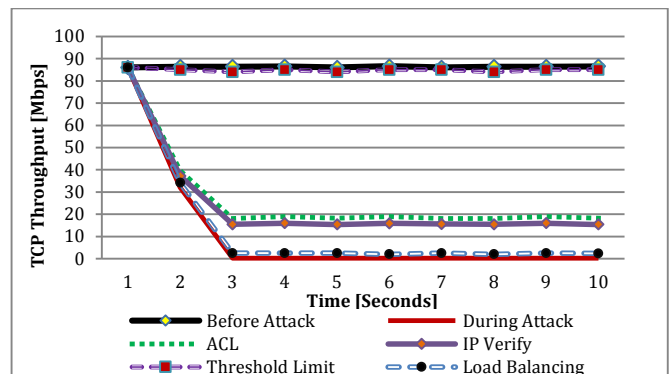


Figure 5: TCP throughput on Windows Server 2012 after using defenses mechanisms.

The TCP throughput value before the attack was stable at around 86 to 86.6 Mbps. During the attack, it significantly dropped to 0.19 Mbps. In this study, Network Load Balancing technique was the worst performer, barely improving the throughput value from 0.19 to 0.251 Mbps. This was due to the fact that, in contrast to other three techniques, NLB did not stop unwanted traffic entering the network; it balanced the attack traffic to an additional server, therefore, allowing attacker to consume entire

network bandwidth by sending more attack packets. NLB was also the most expensive defense solution as it required additional server and cables. Comparing the three defenses implemented on the router, Threshold Limit offered the most effective defense, in which the number of throughput values before the attack and after using the defense was almost the same, about 86.60 Mbps, outperforming other defense techniques by large margin. ACLs came in second, increasing the TCP throughput value from 0.19 Mbps to 18.48 Mbps, followed closely by IP Verify which increased TCP throughputs to 15.67 Mbps.

The huge gap in performance between the three defenses, implemented on the router, was due to the fact that Threshold Limit allows all packets to enter the network but limits the traffic rate up to a flexible threshold, therefore maintaining a steady flow of packets and the traffic fluctuations. The main disadvantages of this defense technique is the fact that both legitimate and attack packets could be dropped when the packet rate exceed the upper rate, resulting in a higher collateral compared to other defense techniques.

ACLs, and IP Verify are both effective defense when attackers use private IP addresses, or unreachable IP addresses respectively, but they cannot block attacking traffic if attackers use legitimate IP addresses not on the black list.

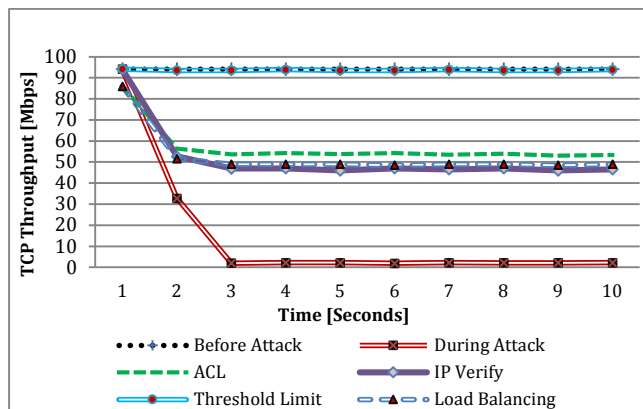


Figure 6: TCP throughput on Linux Ubuntu 13 after using defenses mechanisms.

Figure 6 presents the TCP throughput values after using defenses on Linux Ubuntu 13. The result shows that the TCP throughput value before the attack was stable at 94.1 Mbps. During the attack, the TCP throughput significantly dropped to 0.36 Mbps.

The most effective defense in this study was also Threshold Limit in which the number of throughput values before the attack and after using solutions were almost the same, at 86.60 Mbps. ACLs came in second, and increased the throughput value from 0.36 Mbps to 53.37 Mbps, while Network Load Balancing and IP Verify increased the throughput to 47.39 Mbps and 46.93 Mbps, respectively. Overall, similar trend compared to Windows Server 2012 but surprisingly both ACLs, and IP Verify, offered about three times better performance on Linux Ubuntu 13 compared to Windows Server 2012.

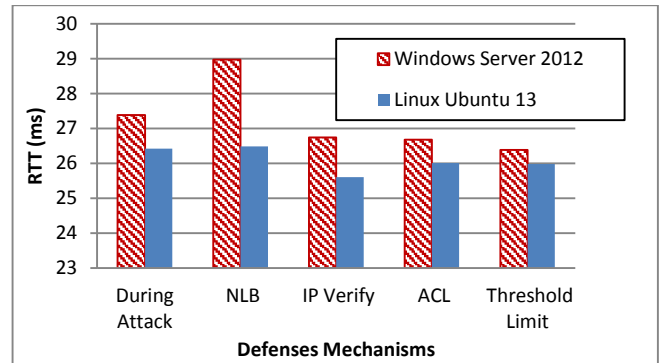


Figure 7: RTT comparison after using DDoS defenses.

Figure 7 shows the comparison of round-trip time after using DDoS defenses on Windows Server 2012 and Linux Ubuntu13. On the whole, the result shows that Linux Ubuntu 13 produced the delay values lower than Windows Server 2012.

In terms of defenses mechanisms, IP Verify was the most effective defense among the other defenses, which reduced the RTT of Windows from 27.384ms to 26.746ms, and from 26.422ms to 25.604ms for Linux. Threshold Limit came in second, which reduced RTT of Windows to 26.39ms and 25.98ms for Linux Ubuntu 13.

Access Control List came in third, which reduced the RTT of Windows to 26.677ms and 26.007ms for Linux Ubuntu 13. Interestingly, Network Load Balancing resulted in the highest RTT, which was 28.973ms for Windows and 26.487ms for Linux. It can be noted that this number was even higher than the number of RTT during the attack. The reason behind is that the load-balancing solution requires system resources to examine incoming packets and make load-balancing decisions, and thus impose an overhead on network performance [6].

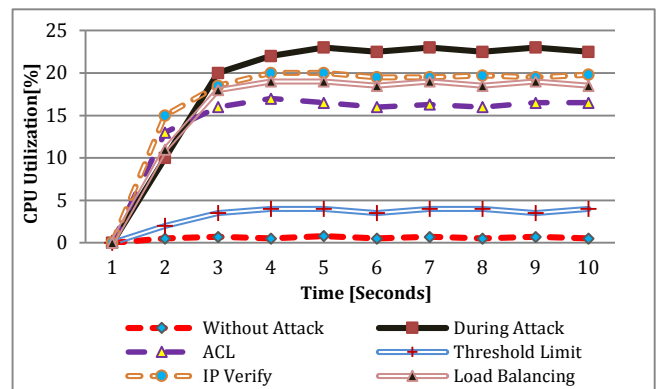


Figure 8: CPU utilization on Windows Server 2012 after using defenses mechanisms.

Figure 8 illustrates the average CPU utilization before and after using defenses on Windows Server 2012. The result shows that the CPU usage before the attack was stable at around 1%. During the attack, the CPU usage fluctuated between 19% and 24%.

The most effective defense in this study was the Threshold Limit, which decreased the CPU utilization from 24% to 3%. ACLs came in second, and reduced the CPU usage from 24% to 16%. Network Load Balancing decreased the CPU usage to approximately 18%. This figure was similar to IP Verify, which reduced the server's CPU to 20%.

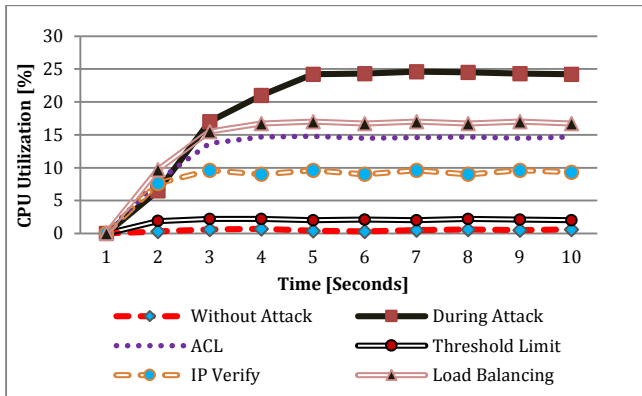


Figure 9: CPU utilization on Linux Ubuntu 13 after using defenses mechanisms.

Figure 9 illustrates the average CPU utilization after using defenses on Linux Ubuntu 13. The results show that the CPU usage before the attack was stable at around 0.3 to 0.7%. During the attack, the CPU usage of Linux platform fluctuated between 23% and 25%.

The most effective defense in this study was Threshold Limit, which decreased the CPU utilization to 3%. IP Verify came in second, and reduced the CPU usage from 25% to 10%. In terms of ACLs, the CPU usage went down from 25% to 15%, while Network Load Balancing reduced the server's CPU usage to 18%.

## VI. CONCLUSION

In this paper, we study the impact of a UDP flood attack on Windows Server 2012 and Linux Ubuntu 13. The result shows that Linux Ubuntu 13 outperformed Windows Server 2012 in terms of TCP throughput values, the RTT, and CPU utilization. In terms of TCP throughput, the result shows that Linux produced 8Mbps higher throughput values than Windows (before attack) and 0.2Mbps more bandwidth during the attack. The RTT result shows that Windows had higher delay values than Linux before attack (0.99ms vs 0.62ms) and during attack (27.38ms vs 26.44ms). The CPU usage result shows that Windows OS utilized the CPU higher than Linux OS. Before the attack, Linux OS utilized the CPU at 0.7%, while Windows OS utilized the CPU at 2%. During the attack, the CPU utilization of Windows and Linux OS went up to 24% and 23%, respectively. Among four defense mechanisms studied, IP Verify and Threshold Limit gave better results than the other solutions. They could effectively increase the bandwidth almost to the pre-attack level. On the other hand, Network Load Balancing had the lowest results in both studies.

## REFERENCES

- [1] B. Gupta, C. Joshi, and M. Misra. "Distributed Denial of Service Prevention Techniques" *IJCEE*, vol. 2, no. 3, 2010, pp. 268-276.
- [2] A. Singh and D. Junefa. "Agent Based Preventive Measure for UDP Flood Attack in DDoS Attacks" *IJEST*, vol.2, no. 8, 2010, pp. 3405-3411.
- [3] Y. Rekhter. Address Allocation for Private Internets. RFC 1918, February 1996.
- [4] R. Subramani, "Denial of Service Attacks and Mitigation Techniques: Real Time Implementation with Detailed Analysis," The SANS Institute, 2011.
- [5] F. Baker, and P. Savola. "Ingress Filtering for Multihomed Networks" RFC 3704, March 2004.
- [6] Microsoft (2014). Network Load Balancing Technical Overview. Available: <http://technet.microsoft.com/en-us/library/bb742455.aspx>
- [7] G. Pack, J. Yoon, E. Collins, and C. Estan., "On Filtering of DDoS Attacks Based on Source Address Prefixes," presented at the Securecomm and Workshops, Baltimore, 2006.
- [8] W. Lu, W. Gu, S. Yu.. "One-Way Queuing Delay Measurement and Its Application on Detecting DDoS Attack," *Journal of Network and Computer Applications*, vol.32, no.2, 2009, pp. 367-376.
- [9] X. Rui, M. Li, and Z. Ling. "Defending against UDP Flooding by Negative Selection Algorithm Based on Eigenvalue Sets," in *International Conference on Information Assurance and Security*, Xi'an 2009, pp. 342-345.
- [10] D. Kaur, M Sachdeva, K. Kumar. "Study of DDoS Attacks Using DETER Testbed," in *International Journal of Computing and Business Research*, vol.3, no.2, 2012, pp. 1-13.
- [11] S.S. Kolahi, A.A. Alghalbi, A.F. Alotaibi, S.S. Ahmed, D. Lad, "Performance comparison of defense mechanisms against TCP SYN flood DDoS attack", 2014 6th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (IEEE ICUMT), St Petersburg, 2014, pp. 143-147.
- [12] E. Fulkerson. (2014). Ping Over a TCP Connection. Available: <http://www.elifulkerson.com/projects/tcping.php>
- [13] R. Jones. Netperf 2.4.5 Available: <http://www.netperf.org/netperf/NetperfNew.html>
- [14] S. Sanfilippo. (2014). Hping3. Available: <http://www.hping.org/hping3.html>
- [15] S.S. Kolahi and P. Li, "Evaluating IPv6 in Peer-to-Peer 802.11n Wireless LANs," *IEEE Internet Computing*, Vol. 15 Issue 4, 2011, pp. 70-74.
- [16] Webstress (2014). Website Performance, Stress, and Load Testing. Available: <http://www.paessler.com/webstress>.