Two Layer Defending Mechanism against DDoS Attacks

Kiruthika Subramanian, Preetha Gunasekaran, and Mercy Selvaraj
Department of Computer Science and Engineering, Thiagarajar College Of Engineering, India

Abstract: *Distributed Denial of Service (DDoS) attackers make a service unavailable for intended users. Attackers use IP spoofing as a weapon to disguise their identity. The spoofed traffic follows the same principles as normal traffic, so detection and filtering is very essential. Hop Count Filtering (HCF) scheme identifies packet whose source IP address is spoofed. The information about a source IP address and its corresponding hops from a server (victim) recorded in a table at the victim. The incoming packet is checked against this table for authenticity. The design of IP2HC table reduces the amount of storage space by IP address clustering. The proposed work filters majority of the spoofed traffic by Hop Count Filter-Support Vector Machine (HCF-SVM) algorithm on the network layer. DDoS attackers using genuine IP is subjected to traffic limit at the application layer. The two layer defense approach protects legitimate traffic from being denied, thereby mitigating DDoS effectively. HCF-SVM model yields 98.99% accuracy with reduced False Positive (FP) rate and the rate limiter punishes the aggressive flows and provides sufficient bandwidth for legitimate users without any denial of service. The implementation of the proposed work is carried out on an experimental testbed.*

Keywords: *DDoS, hop count, IP2HC table, clustering, IP spoofing, testbed.*

Received November 9, 2012; accepted April 29, 2013; published online August 17, 2014

1. Introduction

Distributed Denial of Service (DDoS) attacks remain a serious threat to the reliability of the internet. It takes advantage of internet protocols and the fundamental benefits of the internet delivering data packets from nearly any source to any destination. Huge volumes of packets overwhelm network devices as well as servers, or the packets are deliberately incomplete to rapidly consume server resources. Many of these attacks also use spoofed source IP addresses, thereby eluding source identification. The two most basic types of DDoS attacks are: Bandwidth attacks, application attacks. Bandwidth attacks consume resources such as network bandwidth or equipment by overwhelming with a high volume of packets [23]. Targeted routers, servers and firewalls can be rendered unavailable to process valid transactions and can fail under the load. Adaptive Drop Tail Fuzzy Logic (ADT-FL) [6, 28] regulates the queue size of the router buffers based on the prevailing network traffic during congestion.

Congestion occurs not only due to the increased traffic load and users; but also because of the bogus traffic sent by DDoS attackers. The available bandwidth at the bottleneck link completely utilized by the attack traffic drops the legitimate packets. The most common form of bandwidth attack is a packet flooding attack in which a large number of seemingly legitimate TCP, User Datagram Protocol (UDP) or Internet Control Message Protocol (ICMP) packets are directed to a specific destination [23]. To make detection even more difficult, such attacks might also, spoof the source address to prevent identification. Application attacks use the expected behavior of protocols such as TCP and HTTP to the attacker's advantage by tying up computational resources that prevents from processing transactions or requests. Any computer in the network can be easily compromised by DDoS attacks without the knowledge of being attacked. Sophisticated and automated DDoS attack tools like Trinoo, TFN, TFN2K, Mstream, Stacheldraht, Shaft, Trinity and Knight [2, 5, 7, 8, 9, 10, 11, 15] etc., available in the internet do not require technical knowledge to launch a high rate flooding [14] attack. Popular DDoS attack tools and existing traceback mechanisms in a collaborative environment [21] and their security measures are analyzed in this paper. The victims are surprisingly government agencies, financial corporations, defense agencies and military departments. Popular websites like facebook, twitter, wikileaks, paypal and ebay become DDoS victims that interrupted the normal operation leading to financial loss, service degradation and lack of availability [1, 25].

DDoS detection is difficult to predict as illegitimate packets are identical to legitimate packets. DDoS countermeasures are broadly classified as DDoS prevention and DDoS mitigation. Prevention systems filter out all malicious traffic that is not supported by the service or when vulnerable data matches a known signature in a stored bug-database. Mitigating systems repel when an attack takes place. Such systems cause depletion by limiting outgoing bandwidth, dropping requests that do not match predefined regular expressions or by matching threat signatures in a database. Impact analysis of recent DDoS attacks [1, 20] gives an overview, including major factors for causing DDoS attacks and enumerating DDoS incidents in the past. Attack sizes keep increasing year after year and there is no comprehensive solution to defend it successfully. DDoS still remains a powerful threat and a global solution to weed out the attacker completely is a challenging task in the field of information security.

2. Related Work

DDoS attacks are emerging nowadays with matured attack tools [18] in the Internet. Inspite of various detection and defense algorithms [4, 18], DDoS attacks still remains horrendous. These attacks evolve within fraction of seconds with bogus packets making them extremely difficult to combat or trace back the source. Attackers may also use IP spoofing to conceal their identity by making the traceback of DDoS attacks even more complex. Packet filtering technique is one of the methods on lessening the effects of DDoS attacks executed on IP routers. To detect and discard spoofed traffic, various filtering mechanisms are available. Network ingress filtering [13] limits source IP address spoofing. This algorithm removes outgoing traffic that spoofs addresses outside the deploying network's address range thereby preventing random spoofing. The main drawback in this filtering scheme is every ISP should be alerted to implement this scheme. Processing overhead and additional router configuration makes this filtering scheme impossible to deploy widely in ISP's.

Route-Based Filtering (RBF) [24] is an effective spoofing defense algorithm when deployed at a vertex cover of the autonomous system. The algorithm when deployed at the router keeps an incoming table, which links each source IP address with the expected incoming interface. The packets that match the particular interface are accepted whereas packets that arrive on unexpected interface are dropped as spoofed. Any packet with the source address and the destination address that appear in a router is discarded if it doesn't exist in the path. The RBF algorithm fails to explain the construction and maintenance of tables during change in routing infrastructure. Inter Domain Packet Filtering (IDPF) [12] are built from the information implicit in Border Gateway Protocol (BGP) and is deployed in network border routers. Compared to RBF algorithm, it filters only specific spoofed traffic, because they mark multiple incoming interfaces as expected and cannot detect the interface that is used by a source. This algorithm lacks handling subnet spoofing address. Packet marking [22, 35] is based on routers in which a fingerprint is established for an attack packet by the cooperation among the routers which are located on the attack path. The victim examines the fingerprints with the source IP address and then identifies the spoofed packets. Peng *et al*. [26] proposed a packet-filtering scheme on historical packet information. This method is applied on ingress routers. Packet marking and filtering is proposed in the Stack Pi [35] algorithm. This algorithm eliminates the marking holes generated by legacy routers and includes the markings from single legacy routers immediately following Pi-enabled routers in a path. When the packet arrives at its destination, its mark consists of stacked labels placed by markers that have forwarded this packet and can be used as path identifier. Incoming packets associated with the marks stored with the source address are accepted when exact match is found. Packets with incorrect marks are considered as spoofed one and it is discarded. However, to improve the functionality of this algorithm, it is important to place the markers to maximize the distinctiveness between sources. The historical record is used to decide whether to admit or deny an incoming packet. These filtering methods have a number of drawbacks that limit their application in DDoS defense. It needs the cooperation of all the routers on the attack path, which is obviously hard to be fulfilled on the internet. To summarize, all the related work described above lacks in practical deployment, real time applications and require compulsory collaboration with the ISP's.

The proposed methodology incorporates Hop Count Filter-Support Vector Machine (HCF-SVM) model at the victim that requires no co-operation from the routers in the attack path and without third party involvement. In addition, the detection and mitigation is carried out on a realistic (real time) experimental testbed. In this paper, a DDoS mitigation technique is proposed which is a two layer approach using network-based packet filtering and application layer based bandwidth rate limiting to detect and filter high-rate traffic flows with spoofed IPs.

3. Work Model

Our principal methodology is to construct a table based on the source IP addresses and the relevant hops from the server in normal condition. During the period of DDoS attack, the attacking packets with random spoofing would be filtered for their source IP addresses. A Time-To-Live (TTL) based HCF scheme [27] is being extended with machine learning algorithm and the experimental results are carried out in DDoS testbed. The efficiency and accuracy of the proposed work is improved using machine learning technique. By clustering the address prefixes, the amount of storage space is reduced. Further, the improvement in the network performance metrics is also shown graphically. HCF-SVM method is deployed at victim without the assistance of routers, and is convenient to put into practice. The attackers using genuine IP addresses are subjected to traffic limit [34] at the application layer. The existing HCF mechanism [17] is applied at the network layer where spoofed traffic is classified from normal using SVM and traffic limit at the application layer as shown in Figure 1.

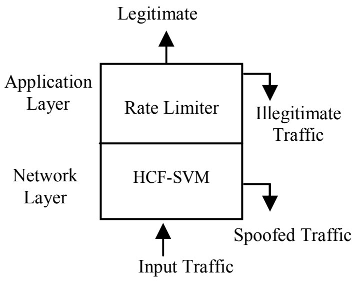

Figure 1. Block diagram of the proposed work.

3.1. HCF Algorithm

Based on the TTL field of the IP header hop count is computed. TTL is an 8 bit field in the IP header, originally introduced to specify the maximum lifetime of IP packets in the Internet. During transmission, each intermediate router decrements the TTL value of an IP packet by one before forwarding it to the next-hop router. The final TTL value when a packet reaches its destination is therefore, the initial TTL subtracted by the number of intermediate hops. The challenge in hop count computation is that a destination only sees the final TTL. Since, the OS for a given IP address may change at any time, a single static initial TTL value for each IP address cannot be assumed.

Fortunately, most modern OSs uses only a few selected initial TTL values 30, 32, 60, 64, 128 and 255 according to [33]. This set of initial TTL values cover most of the popular OS, such as Microsoft Windows, Linux, variants of BSD and many commercial Unix systems. The hop count inspection algorithm [17] as shown in Algorithm 1. Extracts the source IP address and the final TTL value from each IP packet. The algorithm infers the initial TTL value and subtracts it from the final TTL value to obtain the hop count. Then, the source IP address serves as the index into the table to retrieve the correct hop count for this IP address. If the computed hop count matches the stored hop count, the packet will be "authenticated" otherwise, the packet is classified as "spoofed".

Algorithm 1: HCF algorithm

```
for each incoming packet
    Extract the final TTL $T_f$ and Source IP S;
    Infer the initial TTL $T_i$;
    Calculate the hop count $H_c = T_f - T_i$;
    Index S to get the stored hop count $H_s$;
    If ($H_c=H_s$)
        Packet is legitimate;
    else
        Packet is spoofed;
```

3.2. IP2HC Mapping Table

HCF removes spoofed traffic with an accurate mapping between IP addresses and hop counts. Our objectives in building an IP2HC mapping table are: Accurate and up-to-date IP2HC mapping, moderate storage requirement. By clustering address prefixes based on hop counts, an accurate IP2HC mapping table is built and it maximizes HCF's effectiveness without storing the hop count for each IP address. According to the first 24 bits of IP addresses, the hosts are grouped and it is a common aggregation method. The hosts whose network prefixes are longer than 24 bits, may reside in different physical networks in spite of having the same first 24 bits. Thus, these hosts are not necessarily co-located and have identical hop counts. IP addresses are further divided within each 24 bit prefix into smaller clusters based on hop counts as shown in Figure 2.

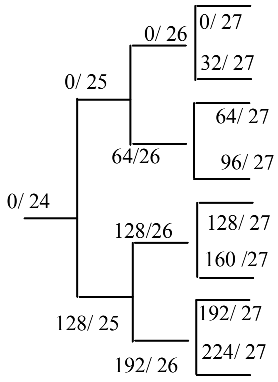

Figure 2. Aggregation with hop count clustering.

To understand whether this refined clustering improves HCF over the simple 24 bit aggregation [3], the filtering accuracies of HCF tables under both aggregations, the simple 24 bit aggregation (without hop count clustering) and the 24 bit aggregation (with hop count clustering) are compared. IPv4 addresses are 32 bit n signed integers and frequently represented as four "dotted-quad" octets (A. B. C. D). IP routing and address assignment use the notion of a prefix. The bit-wise and between a prefix '*p*' and a net mask '*m*' denotes the network portion of the address. The common notation *p/m* is the set containing b-bit IP addresses as in Equation 1 inclusive of:

$$p/m=[p, p+2^{b-m}-1] \tag{1}$$

Thus, *p/m* contains 2^{32-m} addresses. Define split '*s*' as inducing 2^s partitions, p_j as in Equation 2 on *p/m* [3]. Then, for $j=0, ..., 2^s-1$.

$$p_j=p+j2^{32-(m+s)}/(m+s) \tag{2}$$

Keeping the IP2HC mapping up-to-date is necessary for our filter to work in the Internet where hop counts may change. While adding new IP2HC entries or capturing legitimate hop count changes, one way to ensure that only legitimate packets are used during initialization and update is through TCP connection establishment. The HCF table should be updated only by those TCP connections in the established state.

3.3. SVM

SVM classifiers [16, 30] are very promising solution in the field of computational intelligence since, they outperform other classifiers with minimum generalized errors. SVM classifiers are widely employed in intrusion detection where the recent research trend revolves around detection of DDoS attacks. It has several advantages: Scalable in real time scenarios, speed and high accuracy. Consider a training set of instance-label pairs (x_i, y_i), $i=1, ..., l$, where $x_i \in R^n$ and $y \in \{1, -1\}^l$, where (x_i, y_i) denotes training data, n denotes the input vector and y denotes the class either 1 or -1 as shown in Figure 3.

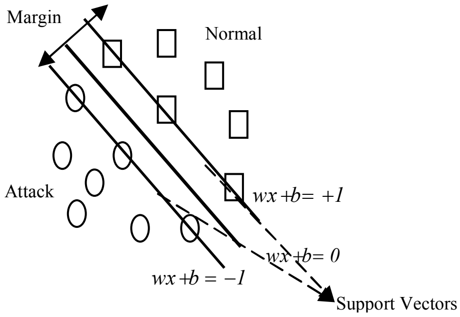

Figure 3. Binary SVM classifier.

The hyper plane formula is:

$$(w.x)+ b= 0 \tag{3}$$

Where, 'w' is the weight factor and 'b' is the bias manipulated during the training phase. Hyper plane is defined such that:

$$(w.x)+b \geq 1,\ if\ y_i= +1,\ Class\ Normal \tag{4}$$

$$(w.x)+b \leq 1,\ if\ y_i= -1,\ Class\ Attack \tag{5}$$

SVM maps real-valued input vectors into a higher dimensional feature space through nonlinear mapping using kernel function. A polynomial kernel function [16] of degree 'd' is used to fit the hyper plane for classification of attack and normal traffic. The polynomial kernel function is shown as below:

$$k(x_{i,}\ x_j)= (1 + x_i,\ x_j)^d \tag{6}$$

3.4. HCF-SVM

First, the dump data is processed to check whether all the sources have completed three-way handshake. The packets are directed to HCF system which satisfies the above mentioned criteria. Now, unique source IP and their associated TTL values are fed into HCF algorithm, which operates in offline mode. Existing HCF algorithm is primarily used as a base for the analysis of a DDoS defense algorithm. HCF aims at detecting spoofed traffic from normal traffic, since spoofed traffic will share the same resources [31] as the legitimate users. Most of the DDoS attacks are launched using spoofed IP addresses where the attackers hide their identity making traceback very complex and difficult. A strong filtering mechanism need to be deployed at the victim to weed out spoofed traffic before the attack traffic overwhelms the target resources. So, the existing HCF algorithm is used for the analysis of its effectiveness against flooding style attacks. HCF algorithm is used since, it operates at the potential victim side and does not involve the cooperation of intermediate routers. Though an attacker modifies any field in the IP header, he cannot falsify the number of hops a packet takes to reach its destination, which depends on the internet routing infrastructure. The hop count information is indirectly reflected in the TTL field of IP header, since each router decrements it to one before forwarding to next hop router [17]. The information about a source IP address and its responding hop count to the victim (server) is stored in an IP2HC table when the network is attack free. Now, the major concern is that IP2HC table should be updated and accurate. For this purpose, the training period lasts from a few weeks to months, so that the majority of legitimate clients in the network under observation are learned and a normal profile being built using SVM model.

The hop count variations during network changes are recorded and statistics maintained to track the stable hop count for the respective source IP. SVM model is learned and updated with source IP and the respective stable hop count. During testing phase, the victim under observation is flooded with attack traffic where the packets come from spoofed sources. The attack traces are checked for three way handshake. The sources completing the handshake are allowed, whereas the rest are dropped after a fixed time out.

The remaining packets pass through HCF system where the features extracted from the attack packet are source IP address and the final TTL. The algorithm infers the initial TTL value and subtracts it from the final TTL value and thus, hop count is obtained. If the calculated hop count matches with the stored hop count value then the source address is 'genuine' else it is considered as 'spoofed' as shown in Figure 4.

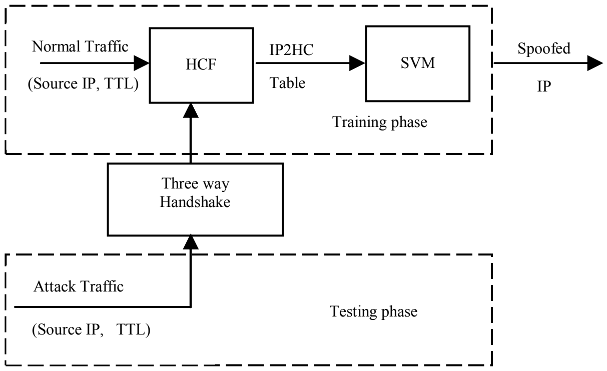

Figure 4. HCF-SVM model.

4. Scheme Implementation

The traffic traces are collected using an experimental testbed. Packets come from several nodes located at different locations. These collaborative working nodes are interconnected through MPLS-VPN cloud. The incoming packets from these nodes are collected using wireshark. The useful information from the packet, the source IP address and TTL value are collected. An IP address to hop count table is constructed using the traffic traces and is stored at the victim end. The random IP spoofing is done to check the effectiveness of the learning process. The new packet on arriving is checked for its authenticity. The source IP and TTL values are extracted and the hop count is the difference between initial and final TTL value. Then, this source IP is searched against IP2HC table and once found it is treated as legitimate, otherwise it is considered as spoofed.

Firstly, SVM distinguishes spoofed attack traffic form the normal traffic. Secondly, rate limiter further mitigates the subsets of attack traffic namely; TCP flood attack, UDP flood attack and ICMP flood attack. SVM model is built with the normal profile and the attack profile is collected for the period of two weeks. The training and testing samples are collected in the campus network during normal activities to track the user behaviour like browsing sites, downloading files etc., the traffic samples that depict the normal user activities are logged. Then, various types of flooding attacks such as TCP flooding, UDP flooding and ICMP flooding are targeted to victim system that sabotage the crucial resources of the system namely memory, processing speed, bandwidth usage and so on. The traffic sample that illustrates the flooding kind of attacks is logged.

The training dataset contains 3000 instances out of which 2000 instances are labelled as attack and remaining 1000 instances are labelled as normal. On keen observation of the collected statistics one can infer that the traffic statistics remain in constant interval during the normal state where the victim system is attack free. Yet, during the attack there may be a drastic change in the statistics that may be very unusual in the campus network.

To validate the proposed methodology, a testing dataset with 617 attack instances and 382 normal instances are fed into the SVM model. The detection accuracy rate of the SVM is 98.99% when C=1.0 in the polynomial kernel function. The time taken to build a model was observed as 9.82 seconds. The proposed HCF-SVM model performance is evaluated with other existing classifiers such as decision tree (J48) and random forest as shown in Table 1.

Table 1. Performance evaluation of the proposed model.

S. No	Classifiers	Correctly Classified Instances (%)	Incorrectly Classified Instances (%)
1	Proposed Model	98.99%	1.01%
2	Random Forest	93%	7%
3	Decision Tree	61.76%	38.24%

Thus, HCF-SVM model at the victim end offers high detection accuracy with low false positive rate thereby allowing the detection methodology for real time deployment. To evaluate the detection accuracy of the chosen classifiers, True Positive (TP), False Positive (FP), precision, recall, F-Measure and accuracy metrics are observed which is shown in Table 2. *TP* is the amount of attack detected when the system is actually under attack. *FP* is the amount of attack detected when the system is actually normal. Precision is the percentage of instances a classifier labels as "relevant". Recall is the percentage of relevant labels that are predicted. F-Measure is a combined measure for precision and recall. Accuracy refers to the proportion of data classified as an accurate type in the total data, namely, the situation *TP* and *TN*.

Table 2. Performance metrics.

Class	TP	FP	Precision	Recall	F-measure	Accuracy
Normal	0.974	0	1	0.974	0.987	98.99%
Attack	1	0.026	0.984	1	0.992	

$$TP=TP/(TP+FN) \tag{7}$$

$$FP=FP/(TN+FN) \tag{8}$$

$$Precision=TP/(TP+FP) \tag{9}$$

$$Recall=TP/(TP+FN) \tag{10}$$

$$F\text{-}Measure=2\ {*}Precision{*}Recall\ /(Precision+Recall) \tag{11}$$

$$Accuracy=(Correctly\ Classified\ Instances/Total\ No.\ Of\ Instances){*}100 \tag{12}$$

The incoming traffic needs to be investigated again to check if legitimate users misbehave after passing through the filtering mechanism. Though HCF algorithm is very effective against spoofed traffic, it doesn't attempt to protect victim from flooding DDoS attacks coming from genuine Source IP's. Therefore, the incoming traffic even after passing HCF filtering needs further treatment. Attackers are more aggressive than legitimate users; their intention is to overwhelm the victim and not to affect any other components. Ultimately, they deny the request from the legitimate users.

The filtered output from HCF-SVM algorithm is passed through rate limiter. The punished flows when behaving aggressive even after the bandwidth cut-off; bandwidth is reduced accordingly and at one point the flows are allotted zero bandwidth. Thus, rate limiting can achieve more legitimate users passing through the bottleneck link with sophisticated bandwidth, thereby increasing the throughput. This is simulated in ns-2 network simulator [32].

The ns-2 is a packet level, discrete event simulator, widely adopted in the network research community. A simple topology is created with 6 attack sources, 4 legitimate sources and 1 target. The bottleneck link capacity is 1 Mbps as shown in Figure 5. Packets coming from four nodes, including node 2 to node 5, are from normal users and packets coming from six nodes, including node 6 to node 11, are from attackers. All packets are routed through the main node, node 0, and are sent to the target node, node 1. Normal packets coming from node 2 to node 5 have flows of 0.1 Mbps from 0 to 10 seconds. Node 6 is used by a DoS attacker with a flow of 0.8 Mbps from 10 to 20 second of simulation. Nodes 7 to node 11 are used by DDoS attackers, each with a flow of 1 Mbps sent after 20 second of simulation.

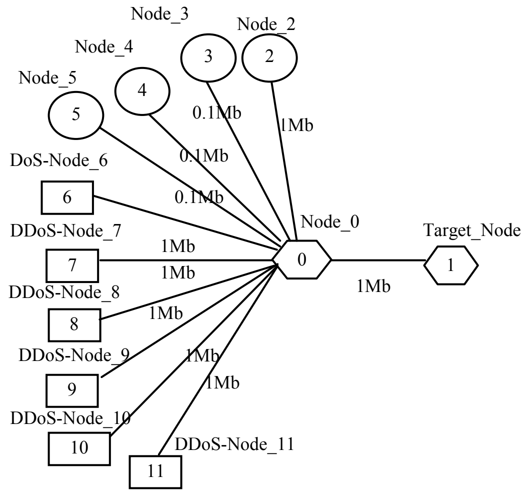

Figure 5. Network topology under DDoS attack.

The rate limiter takes into account, the aggregate traffic at the router. If the aggregate traffic is relatively high to hold the packets at the bottleneck link then the bandwidth for those aggressive flows is reduced by 1/10th. By doing this, the denial of services to genuine flows can be avoided to a greater extent. This collaborated defense mechanism can yield maximum efficiency. Moreover, aggregating the IP addresses leads to less memory, less storage space. The proposed model is highly robust against varying traffic conditions with varying attack strength. The parameters chosen to evaluate the performance of the proposed model are packet loss rate, Throughput and delay. Low packet loss rate leads to increase in throughput and decrease in delay thereby increasing the quality of service.

4.1. Packet Loss Rate

Packet loss rate describes how many packets are lost in transit between the source and the destination [29]. There is considerable decrease in packet loss rate using HCF-SVM with rate limiter for varying traffic intensities. Its performance against legitimate, DoS and DDoS node is clearly shown in Table 3. Packet loss rate (*Pkt_loss_rate*) can be calculated using the Equation 13.

Table 3. Packet loss rate before and after rate limiting.

S.No	Nodes	Before Rate Limiting (%)	After Rate Limiting (%)
1	NODE_2	50.98	30.17
2	DoS-NODE_6	60.89	48.9
3	DDoS-NODE_7	70.37	60.28

$$Pkt\_loss\_rate = (Tot\_pkt\_lost / Tot\_pkt\_sent) * 100 \quad (13)$$

4.2. Throughput

Throughput is the rate at which a network sends or receives data and usually rated in terms bits per second (bit/s) [29]. It describes the rate of the packet measured as an average or peak which is an important factor that directly accounts for the network performance. Better throughput is achieved by HCF-SVM with rate limiter for legitimate node and reduced throughput for DoS and DDoS nodes as shown in Figure 6.

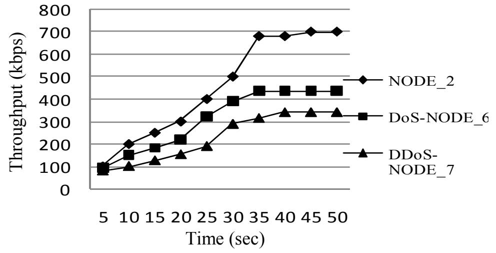

Figure 6. Throughput vs time.

4.3. Delay

Delay describes the time taken for a packet to travel from the source to target. The delay is not only due to the propagation time, but also the time spent in the queues and the processing time. Delay is the time interval between the generation of a packet from a source node and the successful delivery of the packet at the destination node. It counts all possible delays that can occur in the source and all intermediate nodes, including queuing time, packet transmission and propagation and retransmission [19]. The queuing time can be caused by network congestion or unavailability of valid routes. Delay is comparatively low for legitimate node after rate limiting the DoS and DDoS nodes, which are the major sources for denial of services and its robustness of the HCF-SVM with rate limiter is shown in Figure 7. Congestion occurs when the traffic intensity is high which implies that the packet arrival rate is high at the router.

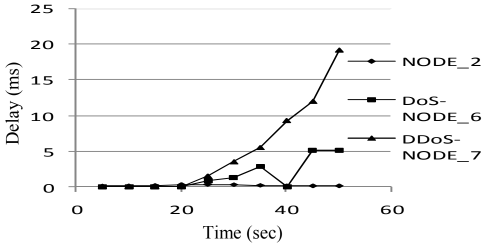

Figure 7. Delay vs time.

The proposed HCF-SVM model coupled with rate limiter can provide sufficient bandwidth to legitimate users since, majority of the attack traffic is rate limited at the router nearest to the victim. Thus, the proposed model maintains an adequate level of packets in the queue under varying traffic and network conditions, so that the packet loss remains at a lower rate. It behaves well with low packet loss which in turn leads to high throughput, low queuing delay with good network performance. From the results, it is seen that the network performance is improved for legitimate node and the legitimate packets make their way into the network even under the DoS and DDoS attack and finally survive a critical attack.

5. Conclusions

HCF-SVM works on the potential victim side at the network layer, which has a strong incentive to implement the filtering function. No cooperation among routers is required. It deploys very limited information such as source IP addresses and its corresponding TTL values to filter the attacking packets, which simplifies the requirements for implementation. The design of IP2HC table reduces the amount of storage space. Further rate limiter at the application layer punishes aggressive flows and provides sufficient bandwidth for legitimate users without denial of services. The implementation of the proposed work is carried out on an experimental testbed.

Acknowledgements

We thank the Smart and Secure Environment (SSE) project led by IIT Madras, chennai for providing excellent Infrastructure. This work was supported in part by a grant from the National Technical Research Organization, New Delhi.

References

[1] Arora K., Kumar K., and Sachdeva M., "Impact Analysis of Recent DDoS Attacks," *the International Journal of Computer Science and Engineering*, vol. 3, no. 2, pp. 877-884, 2011.

[2] Barlow J. and Thrower W., "TFN2K an Analysis," available at: http://security.royans.net/info/posts/bugtraq_ddos2.shtml, last visited 2000.

[3] Beverly R. and Sollins K., "An Internet Protocol Address Clustering Algorithm," *in Proceedings of USENIX Tackling Computer Systems Problems with Machine Learning Techniques*, CA, USA, pp.1-6, 2008.

[4] Bhuyan M., Kashyap H., Bhattacharya D., and Kalita J., "Detecting Distributed Denial of Service Attacks: Methods, Tools and Future Directions," available at: http://www.researchgate.net/publication/258875120_Detecting_Distributed_Denial_of_Service_Attacks_Methods_Tools_and_Future_Directions, last visited 2013.

[5] Bysin C., "knight.c Sourcecode," available at: http://packetstormsecurity.nl/distributed/knight.c, last visited 2001.

[6] Devi B., Preetha G., Nidhya S., and Shalinie S., "A Novel Fuzzy Congestion Control Algorithm for Router Buffers," *in Proceedings of IEEE International Conference on Recent Trends in Information Technology*, Tamil Nadu, India, pp. 423-427, 2011.

[7] Dietrich S., Long N., and Dittrich D., "Analyzing Distributed Denial of Service Tools: The Shaft Case," *in Proceedings of the 14th Conference on Systems Administration*, LA, USA, pp. 329-339, 2000.

[8] Dittrich D., "The DoS Project's Trinoo Distributed Denial of Service Attack Tool," available at: http://staff.washington.edu/dittrich/misc/trinoo.analysis.txt, last visited 1999.

[9] Dittrich D., "The Stacheldraht Distributed Denial of Service Attack Tool," available at: http://staff.washington.edu/dittrich/misc/stacheldaht.analysis.txt, last visited 1999.

[10] Dittrich D., "The Tribe Flood Network Distributed Denial of Service Attack Tool," available at: http://staff.washington.edu/dittrich/misc/tfn.analysis.txt, last visited 1999.

[11] Dittrich D., Weaver G., Dietrich S., and Long N., "The Mstream Distributed Denial of Service Attack Tool," available at: http://staff.washington.edu/dittrich/misc/mstrea.analysis.txt, last visited 2000.

[12] Duan Z., Yuan X., and Chandrashekar J., "Constructing Inter-Domain Packet Filters to Control IP Spoofing based on BGP Updates," *in Proceedings of the 25th IEEE International Conference on Computer Communications*, Barcelona, Spain, pp. 1-12, 2006.

[13] Ferguson P. and Senie D., *Network Ingress Filtering: Defeating Denial of Service Attacks which Employ IP Source Address Spoofing*, RFC Editor, USA, 2000.

[14] Ghazali K. and Hassan R., "Flooding Distributed Denial of Service Attacks-A Review," *the Journal of Computer Science*, vol. 7, no. 8, pp. 1218-1223, 2011.

[15] Hancock B., "Trinity v3, a DDoS Tool, Hits the Streets," *the Computers Security Journal*, vol. 19, no. 7, pp. 574-574, 2000.

[16] Jakkula V., "Tutorial on Support Vector Machine (SVM)," available at: http://eecs.wsu.edu/~vjakkula/SVMTutorial.doc, last visited 2014.

[17] Jin C., Wang H., and Kang S., "Hop-Count Filtering: An Effective Defense against Spoofed Traffic," *in Proceedings of the 10th ACM Conference on Computer and Communication Security*, DC, USA, pp. 30-41, 2003.

[18] Juyal S. and Prabhakar R., "A Comprehensive Study of DDoS Attacks and Defense Mechanisms," *the Journal of Information and Operations Management*, vol. 3, no. 1, pp. 29-33, 2012.

[19] Karthikeyan N., Palanisamy V., and Duraiswamy K., "A Performance Evaluation of Proactive and Reactive Protocols using NS2 Simulation," *the International Journal of Engineering Research and Industrial Applications*, vol. 2, no. 2, pp. 309-326, 2009.

[20] Kaur D. and Sachdeva M., "Study of Recent DDoS Attacks and Defense Evaluation

Approaches," *the International Journal of Emerging Technology and Advanced Engineering*, vol. 3, no. 1, pp. 332-336, 2013.

[21] Kumar P. and Selvakumar S., "Distributed Denial-of-Service (DDoS) Threat in Collaborative Environment-a Survey on DDoS Attack Tools and Traceback Mechanisms," *in Proceedings of IEEE International Conference on Advance Computing*, Patiala, India, pp. 1275-1280, 2009.

[22] Lee F. and Shieh S., "Defending Against Spoofed DDOS Attacks with Path Fingerprint," *the Computers and Security Journal*, vol. 24, no. 7, pp. 571-586, 2005.

[23] Mirkovic J. and Reiher P., "A Taxonomy of DDoS Attacks and Defense Mechanisms," *ACM SIGCOMM Computer Communication Review*, vol. 34, no. 2, pp. 39-53, 2004.

[24] Park K. and Lee H., "On the Effectiveness of Route-Based Packet Filtering for Distributed DoS Attack Prevention in Power-Law Internets," *in Proceedings of Conference on Applications, Technologies, Architectures and Protocols for Computer Communications*, CA, USA, pp. 15-26, 2001.

[25] Patel C. and Borisagar V., "Survey on Taxonomy of DDoS Attacks with Impact and Mitigation Techniques," *the International Journal of Engineering Research and Technology*, vol. 1, no. 9, pp. 1-8, 2012.

[26] Peng T., Leckie C., and Ramamohanarao K., "Protection from Distributed Denial of Service Attacks using History-Based IP Filtering," *in Proceedings of IEEE International Conference on Communications*, Alaska, USA, vol. 1, pp. 482- 486, 2003.

[27] Raghavan S. and Dawson E., *An Investigation into the Detection and Mitigation of Denial of Service (DoS) Attacks: Critical Information Infrastructure Protection*, Springer, India, 2011.

[28] Shalinie S., Preetha G., Nidhya S., and Devi B., "Fuzzy Adaptive Tuning of Router Buffers for Congestion Control," *the International Journal of Advancements in Technology*, vol. 1, no. 1, pp. 85-94, 2010.

[29] Singh D., Ke C., Jain G., and Sanadhya H., "Measurement of Wireless Network Performance," *in Proceedings of IEEE National Conference on Advanced Technologies and Applications*, Udaipur, India, 2009.

[30] SVM Tutorial-Data Mining Tools., available at: http://www.dataminingtools.net/wiki/svm.php, last visited 2012.

[31] Swain B. and Sahoo B., "Mitigating DDoS Attack and Saving Computational Time using a Probabilistic Approach and HCF Method," *in Proceedings of IEEE International Conference on Advanced Computing,* Patiala, India, pp. 1170-1172, 2009.

[32] The Network Simulator-NS-2., available at: http://www.isi.edu/nsnam/ns, last visited 2012.

[33] The Swiss Education and Research Network., "Default TTL Values in TCP/IP," available at: http://secfr.nerim.net/docs/fingerprint/en/ttldefault.html, last visited 2002.

[34] Wu Z. and Chen Z., "A Three-Layer Defense Mechanism based on Web Servers Against Distributed Denial of Service Attacks," *in Proceedings of the 1st International Conference on Communications and Networking*, Beijing, China, pp. 1-5, 2006.

[35] Yaar A., Perrig A., and Song D., "StackPi: New Packet Marking and Filtering Mechanisms for DDoS and IP spoofing Defense," *the IEEE Journal on Selected Areas in Communications*, vol. 24, no. 10, pp. 1853-1863, 2006.

Kiruthika Devi Bodinayakanur Subramanian is currently pursuing MS (by Research) at Anna University. She received her BE degree in electronics and communication engineering from Coimbatore Institute of Engineering and Information Technology in 2006. Her current research interests include network security and machine learning.

Preetha Gunasekaran is currently pursing PhD degree at Anna University. She received her MSIT in information technology in 2002 and MPhil in Computer Science from Madurai Kamaraj University in 2005. She worked as a Lecturer from 2002 to 2008. Her current research interests include network security and wireless adhoc networks.

Mercy Shalinie Selvaraj is currently the Head of the Department of Computer Science and Engineering at Thiagarajar College of Engineering. She has published several papers in International Journals/ Conferences. Her current areas of interest include machine learning, neural networks and information security.