Work in Progress: Path Attestation Scheme to Avert DDoS Flood Attacks

Raktim Bhattacharjee, S Sanand, and S V Raghavan

Dept. of Computer Science & Engg.
Indian Institute of Technology Madras,Chennai, 600036
{raktim,sanand,svr}@cs.iitm.ernet.in

Abstract. DDoS mitigation schemes are increasingly becoming relevant in the Internet. The main hurdle faced by such schemes is the "nearly indistinguishable" line between malicious traffic and genuine traffic. It is best tackled with a paradigm shift in connection handling by attesting the path. We therefore propose the scheme called "Path Attestation Scheme" coupled with a metric called "Confidence Index" to tackle the problem of distinguishing malicious and genuine traffic in a progressive manner, with varying levels of certainty. We support our work through an experimental study to establish the stability of Internet topology by using 134 different global Internet paths over a period of 16 days. Our Path Attestation Scheme was able to successfully distinguish between malicious and genuine traffic, 85% of the time. The scheme presupposes support from a fraction of routers in the path.

Key words: DDoS mitigation, Unspoofable Identity, Cascaded Filters

1 Introduction

"Distributed Denial of Service" (DDoS) can be defined as an attempt made by malicious users to deny resources and services to legitimate users. DDoS attacks are relatively simple, yet powerful enough to bring down a Critical Infrastructure (CI). DDoS identification schemes can broadly be classified into (1) Behavioral based and (2) Identity based. Though behavioral based schemes can detect unknown attacks, it suffers from several drawbacks as listed in [5]. The main drawback of identity based techniques is the possibility of source IP spoofing. The first part of our work is the creation of an unspoofable identity. The core idea is that the path taken by a packet from a source is dictated by the destination IP and routing behaviour of the Internet, and hence is unspoofable. Spoofing the source IP of a packet will not change its path to destination. The novelty of the proposed technique is that it can mark packets as malicious, with a varying degree of confidence. We propose a metric called *Confidence Index* to quantify this. The scheme is effective against flood attacks, which make use of IP Spoofing.

The second part of this work is a router-level flood attack mitigation scheme. Each router has a set of priority queues and each packet is assigned to one of the

queues based on its *Confidence Index*. Unlike other similar efforts, in Path Attestation Scheme (PAS), small changes in route will not lead to packet drop but only de-prioritization of packets. When cascaded, such routers can exponentially reduce the attack volume reaching the CI. This scheme is particularly effective against bandwidth-depletion attacks.

The last part of our work is a study on different types of errors and performance of proposed technique. The system is prone to false positives occurring due to path changes. Based on an experimental study of 134 routes over a period of 16 days, we have modeled the probability of path changes in the Internet. The probability of false negatives through collision of path identities is modeled analytically. Finally, we study the cascaded effect of a sequence of filtering routers by simulations.

The remainder of the paper is organized as follows: Section 2 deals with the background and related works. In Section 3 we present the Path Attestation Scheme (PAS) framework. In Section 4 we analyze the effect of change in network topology in PAS. Section 5 deals with security and performance analysis of the proposed system and finally Section 6 concludes the paper.

2 Background And Related Works

Limitations of the Current Internet Architecture: Most of the research effort in the past was directed towards improving the performance and scalability of the Internet. No attention was paid to make it safe and secure. Internet today is susceptible to DDoS because of the lack of Authenticity, Accountability and Uniformity. Without authentication any user can claim any identity and there is no means to trace these malicious users. Infact there is very little that present day routers can do to improve the situation. Routing and forwarding protocols are designed to be destination oriented. Routers are designed to forward the packet without bothering about where it has come from. Lastly the resources in the internet are not uniformly distributed, a lot of potential is concentrated in the core of the network which can be evenly distributed to the edge network.

DDoS Prevention Techniques: All DDoS prevention schemes can be classified into three classes based on the place of deployment as (1) source based, (2) host based and (3) network based. Deploying DDoS solution at the source itself is the ideal case because it saves the network resources from unwanted traffic. Ingress Filtering [1] is one such type of solution where ingress routers block packets that arrive with source addresses having prefixes that do not match the customer's network prefixes. DWARD [4] is another solution that performs a proactive identification and filtering of suspicious flows originating from a customer network. The impact of the source based prevention cannot be felt directly by the deploying network, because of which there is little motivation for Internet Service Providers to deploy source based schemes in their network.

The host based DDoS mitigation schemes are preferred because the benefit of DDoS prevention is felt directly by the deploying system or network. Hop-

Count Filtering [8] is a type of host based scheme. Other spoofing prevention method like history-based filtering [7] and Packet-score [3] have also adopted this approach. Though these solutions are good for preserving server resources, they do not prevent the abuse of network resources. Moreover, there is a possibility of launching an attack against the prevention system in which case the prevention system itself becomes a single point of failure.

The network based solutions require support from routers as well as wide scale deployment to be effective. The Pushback scheme [2] view flooding by DDoS as a congestion problem. This scheme requires router modification to detect and drop packet belonging to an attack flow. Further, network based solutions like Pi [10] and SIFF [9] use path based identification to filter out attack packets. These methods are prone to false positives due to frequent route changes and load balancing, in which case legitimate traffic may get filtered even in the absence of an attack.

Considering the different mitigation schemes we can say that Network based solution is a must if DDoS needs to be nipped in the bud. Thus we have proposed PAS that not only allows router level differential filtering but also does not suffer the drawbacks of other network based mitigation schemes.

3 Path Attestation Scheme (PAS)

The objective of the PAS is to mitigate flood attacks and give service to the legitimate users. The flooding problem is relatively difficult to handle because there is no way to differentiate between spoofed and genuine traffic. This differentiation is possible if we can attach an element to the packet that the attacker cannot modify. One such element that an attacker cannot modify is the path of the packet from its source to destination.

Any packet moving from a particular source to a destination follows a path. It is highly probable that the same path will be followed by subsequent packets between those two systems. Based on the amount of deviation from the older path each packet is given a *Confidence Index*. Packets with less deviation will have high *Confidence Index* compared to packets with more deviation. Packets with very low or zero *Confidence Index* are considered to be suspicious or malicious. Based on the *Confidence Index* packets are segregated in different priority queues. The packets in the higher priority queues will be processed before any packet in lower priority queues. When there is a DDoS flooding, lots of spoofed packets will be generated and these packets will have low *Confidence Index*. These packets will be assigned to the default/lowest priority queue. Thus, when there is a congestion (caused due to flooding) the lowest priority queue will get filled with spoofed packets. This will inturn result in packet drop, which reduces the number of attack packets leaving the router. The whole process in described in detail in the following subsection.

3.1 ID List Creation

Each router in the path puts its signature (ID) in the packet. The sequence of router IDs of a path forms an *ID List*. The ID, which router puts in the packet is the function of (1) source IP address, IP_{src}, and (2) Interface Identity Layer 2 Address, ID_{inf}, of the router outgoing interface.

$$ID = hash(IP_{src}|ID_{inf}) \tag{1}$$

The hash function proposed to use is MD5. MD5 generates 128 bit hash value but only the most significant 'n' bits will be taken as ID.

The purpose of selecting the router outgoing interface for the hash is to have invariability and uniqueness properties of path into the ID. Packets from different sources will have different path to a destination and thus different IDs. The reason for the ID to be a function of Source IP is that packets with different source IPs should have different IDs. In the event of an IP spoofing attack, a series of packets will be generated with random source IP address to a destination. Since ID is a function of Source IP also, each of these packets will have different ID value because of the change in source IP.

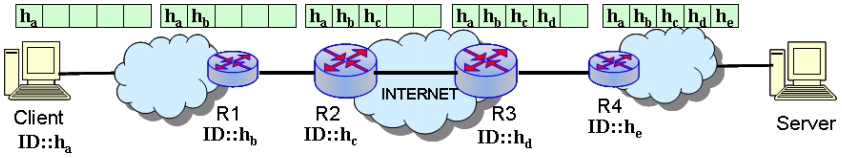
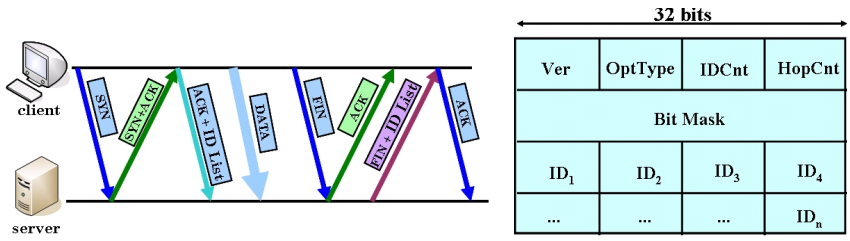


Fig. 1. ID List creation process.

At the beginning of a TCP connection, i.e. when the first ACK is sent from client to the server, the *ID List* is generated. As the packet moves from source to destination each of the routers in the path calculates the 'n' bit ID and puts it into the packet as shown in Fig.1. The forty byte option field of IP packets is used to carry this *ID List* . When the server receives this, it keeps the *ID List* for subsequent use.

3.2 ID List Delivery

Once the *ID List* of a client is received at the server, it is kept in its memory. This *ID List* needs to be transferred to the client after verifying its authenticity. This is done at the end of a genuine TCP connection, piggybacked in the FIN packet. A connection is considered to be genuine when there is transfer of a certain amount of data between the client and the server. The rationale behind sending the *ID List* to client at the end of a genuine TCP connection is that; one cannot establish a TCP connection using a spoofed IP and thus cannot receive the *ID List* back. This *ID List* delivery process is shown in Fig.2(a).



(a) (b)

Fig. 2. (a) ID List delivery process. (b) Proposed IP Option Field.

3.3 ID List Verification

Once the client receives the *ID List* from the server it keeps it in its file system. From the next connection onwards client inserts this *ID List* to every packet destined to that server. All the routers in the path verifies this *ID List*. Each router in the path calculates the 'n' bit ID using the IP_{src} and ID_{inf}. The proposed IP option (Fig.2(b)) field carries a Bit Mask field. Each bit of the Bit Mask field is used to specify whether the corresponding ID is valid or not. By default all the bits in this Bit Mask is zero, indicating invalid. Now, if the calculated ID matches to the ID carried by packet then it marks the corresponding bit of the Bit Mask as 1 otherwise 0, as explained in Fig.3.

3.4 Packet Classification

Packets are segregated by the router based on a metric called *Confidence Index* which quantifies the degree of maliciousness. *Confidence Index* of a packet is a function of its Bit Mask field. Higher the number of IDs matched, the higher is the number of ones in the Bit Mask and thus higher is the *Confidence Index*. While calculating the *Confidence Index*, the position of the routers in the path is also taken into consideration. The routers near the destination server are considered to be more trusted than the routers near the source. Thus we provide a weightage to each bit position. The weight is a linear function of the router position in the path. The *Confidence Index* of a packet in router 'r' is determined by this expression.

$$C_r = \frac{\sum_{i=0}^{r}(W_i \times V_i)}{\sum_{i=0}^{r} W_i} \tag{2}$$

Where,

C_r is *Confidence Index* of the packet in router 'r', W_i is Weight of i^{th} Bit Mask position ($W_i = i$), V_i is Bit Mask in i^{th} position (1 implies Valid, 0 implies Invalid) and r is the Number of routers passed by the packet.

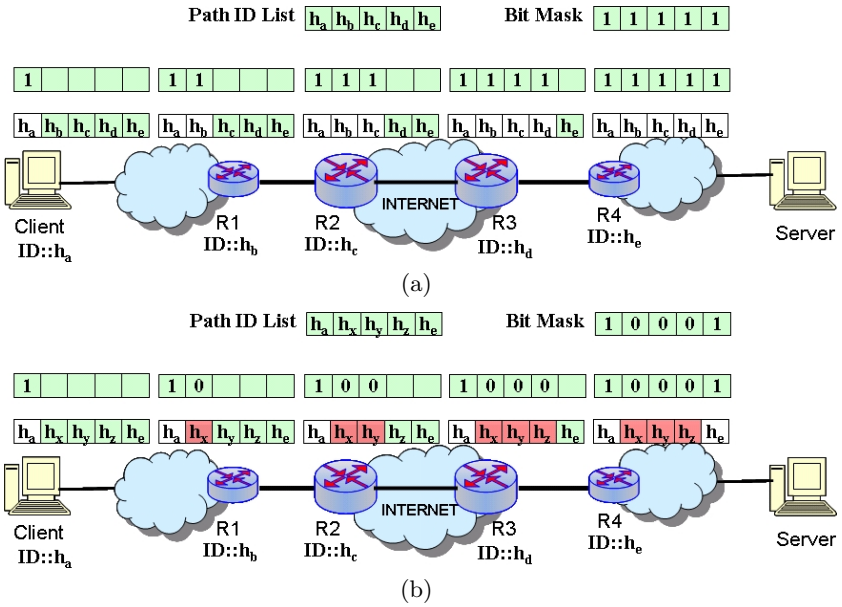


Fig. 3. *ID List verification process.* In Case (a) all the IDs in the List are correct. Router verifies and marks all the Bit Mask as 1. In Case (b) IDs h_x, h_y and h_z are invalid, router marks corresponding Bit Mask as 0.

In the example, in Fig.3, based on the Bit Mask value the *Confidence Index* is calculated. In the first case where all the Bit Mask are 1 the *Confidence Index* is 1 but in the second case the *Confidence Index* is 0.4. When there is a genuine path change, at least a part of the *ID List* will match and the packet will have higher *Confidence Index* than a spoofed packet.

3.5 Router Queuing Process

Each router in the system maintains N_r priority queues. The queue to which a packet belongs is determined based on the value of its *Confidence Index*. The queue selection for a packet is done by this expression.

$$Q = \lceil C_r \times N_r \rceil \tag{3}$$

Based on this 'Q' value packet will be put in one of the priority queues. If a packet does not come with any *ID List* then it will have zero *Confidence Index* and will be put into the default queue, i.e. lowest priority queue. The packets having a valid *ID List* will be put into high priority queues. Spoofed packets will have very low *Confidence Index* and will be put into the lowest priority queue. When there is a congestion, due to DDoS attack, the lowest priority queue will get filled with spoofed packets. This will result in packet drop, which inturn reduces the number of attack packets leaving the router.

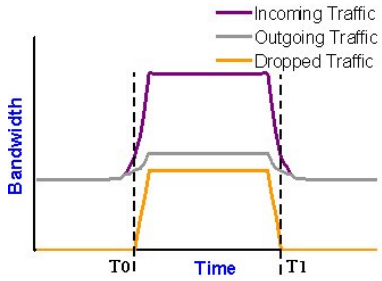


Fig. 4. Bandwidth profile of a router in the event of an attack.

Fig.4 illustrates the temporal variation in incoming, outgoing and dropped traffic in a router in the event of an attack. During the normal phase, when there is no packet drop, the incoming and outgoing bandwidths are same. But when the attack packets start flowing in the lowest priority queue starts getting filled. At one point, T_0, the lowest priority queue becomes completely filled and attack packets starts getting dropped. Beyond this point only a small fraction of attack packets will go out of the router.

4 Experimental Study

The effectiveness of PAS depends on the stability of paths in the Internet topology. According to a study [6] done during 1994-1995, about 2/3 of the Internet paths were having routing persistence of either days or week and most variation was either in one or two routers. Since this result is considerably old, we undertook an experiment to characterize the path change between client and server in the present day Internet.

We selected a set of 134 IPs from traceroute.org, which allows traceroute to their servers. These servers were geographically distributed over 40 countries which gives a more or less true representation of the Internet. A system in Network System Lab, IIT Madras, which has a public IP, was selected as the client. A Perl script was written which will do repeated traceroute operations to these servers at a regular interval. In all 257280 traceroutes were made and the data thus collected were stored in MySQL database to make analysis easier.

4.1 Temporal Variations of Path Stability

The aim of this analysis is to characterize the variation of path stability against different observation intervals. For each observation interval we plot the percentage of paths having hop variations ranging from 0 to maximum hop count. We derive the expression for path stability as follows. Let the set of all paths be 'P'. Let 'p' denote a path and 'h_p' be the number of hop variations of a path 'p'. Then the number of paths having 'h' hop variations is given by

$$n_h = |\{p|p \in P, h_p = h\}| \tag{4}$$

The total number of paths is given by

$$n = |P| \tag{5}$$

The percentage of path having 'h' hops variations is given by

$$\acute{n_h} = \frac{n_h}{n} \times 100 \tag{6}$$

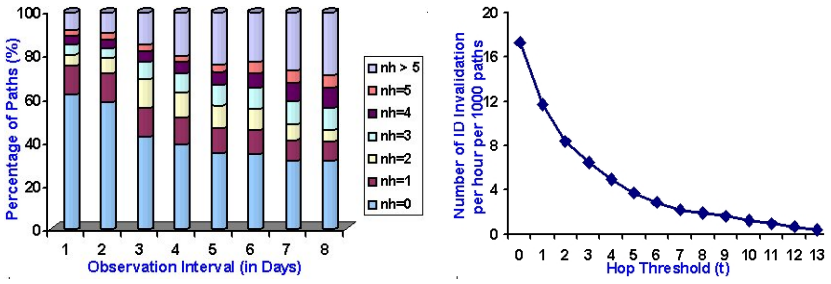
In the Fig.5(a) we have plotted Percentage of Path having 'h' hop variations '$\acute{n_h}$' for different observation intervals. The observation interval is varied from 1 to 8. The value plotted against an interval 'w' is the average of '$\acute{n_h}$' for all possible intervals of size 'w'. From the graph we observe that number of paths having 0 hop variation ($n_h = 0$) is asymptotically approaching a minimum value. We also observed that number of paths having hop variation less than 5 remains above 65%, even for large observation interval. Next, we define a term which is the number of paths having atmost 'h' path variation.

$$\hat{n_h} = \sum_{x=0}^{h} n_x \tag{7}$$

We define path stability as follows,

$$P_t = \frac{\hat{n_h}}{n}, where, h \leqslant t \tag{8}$$

Here,'t' is the maximum number of hop variations which can be sustained by the system. Now suppose that atmost 3 hop variation is acceptable, in that case, for w=1 we have $\acute{n_h} = 0.85$ for w=2 $\acute{n_h} = 0.84$ and for w=3 $\acute{n_h} = 0.76$. Thus it can be observed that if the observation interval is one or two days and at most three hops variation is allowed then we can achieve path stability above 0.8.



(a) (b)

Fig. 5. Graph (a) between the observation Intervals in Days Vs. Percentage of Paths with different hop variations. Graph (b) between the number of ID Invalidations per hour per 1000 paths Vs. Hop Threshold

4.2 Frequency of ID Invalidation

The validity of ID depends on the stability of the path. As long as the path is stable, the C_r value will be higher than user defined threshold T_h and *ID List* will be valid. But due to path changes C_r value will decrease and once it falls below T_h the *ID List* is renewed. Now the frequency of *ID List* change depends on the stability of the path P_t and specifically on the value of t. Fig.5(b) shows a variation of the number of ID invalidation per hour per thousand paths plotted against the acceptable hop variations. From the graph we can infer that if the invalidation algorithm does not accept any hop variation at all, it results in around 17 IDs invalidation per hour per thousand paths. But if we increase the threshold to 3 the number of ID invalidation drops to seven.

5 Security and Performance Analysis

5.1 Analysis of Brute Force Attack

There is a possibility that an attacker may try to spoof the *ID List* so as to give higher priority to his packets. By this analysis we are able to show that such a brute force attack is practically impossible and its success rate is of the order of $(2)^{-m}$ where 'm' is of the order of hundreds. Router prepares an ID which is an 'h' bit hash as explained in equation (1). The probability of an attacker selecting a correct match by brute force is $(\frac{1}{2})^h$. Now, suppose there are 'r' routers between the source and destination, then probability of getting a match in all the router is $\{(\frac{1}{2})^h\}^r$. The higher the size of the ID the higher is the probability of being safe, but at the same time IDs of the entire routers in the path should be accommodated in the 40 bytes IP option field. It is known that the maximum number of hops count in Internet is 30 and average is 14-19 [8]. Thus maximum of 30 router IDs should be accommodated in the option field. If we consider the size of ID as 8 bit, for 30 IDs, space required is 30 bytes which can be accommodated in the option field of 40 bytes. Now considering ID to be 8 bits the probability of getting the entire match in maximum case of 30 routers is $\{(\frac{1}{2})^8\}^{30} = (\frac{1}{2})^{240}$, and the probability of getting the entire match in average case of 17 routers $\{(\frac{1}{2})^8\}^{17} = (\frac{1}{2})^{136}$. These are very small values which implies that the probability of getting the entire IDs match is practically impossible.

5.2 Performance Analysis

Experimental Setup In order evaluate the performance of our system we conducted some experiments considering the multilink topology of the network. The unique feature of PAS is that it can coexist with legacy routers that do not support it.The network setup (Fig.6) consists of two good sources (G) and ten attacking sources (A). There are a number of routers between the source and CI. Some of these routers support PAS (R_p) and some does not (R_L). The two genuine sources are connected to one router where as the attacking sources are distributed over the whole network. Each of the genuine sources and attacking

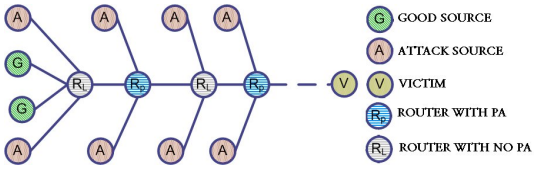
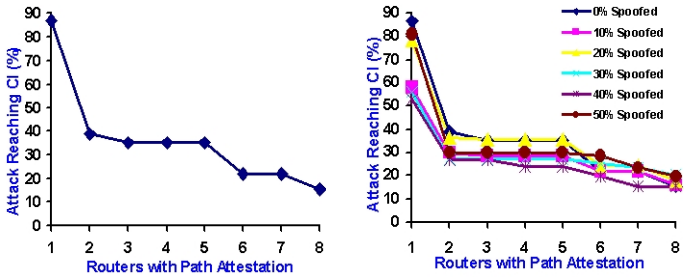


Fig. 6. Network Topology for the Experiment

source are generating traffic at the rate of 1 Mbps each. The routers are designed to handle traffic at the rate of 2 Mbps. The number of attacking sources and numbers of routers between source and destination are varied based on our experimental requirement.

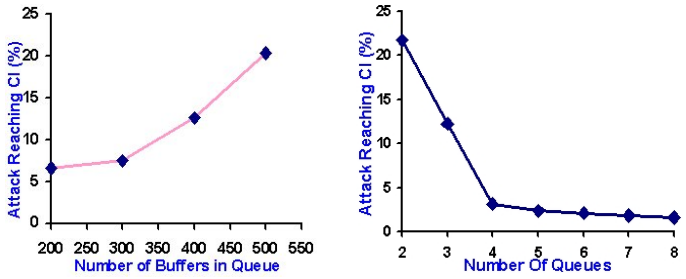


(a) (b)

Fig. 7. Graph between the number of Routers with PAS Vs. percentage of Attack Packets reaching the CI. In case (b) the probability of ID spoofing varied from 0 to 0.5

Performance vs. Router Support The first experiment was conducted to quantify how effective our PAS is in mitigating DDoS flood attack. As this scheme is a multi-level filter, its effectiveness depends on the number of routers that supports it. For this purpose, in the path between source and critical Infrastructure, varied the number of routers with PAS from 1 to 8. The result obtained after this experiment was plotted in the graph in Fig.7(a) . It can be observed from the graph that as number of routers with PAS support increases, the percentage of attack packets reaching the CI decreases exponentially. With the support of 8 routers in the path we are able to mitigate DDoS flood attack reaching CI upto 85%. The same experiment was conducted assuming that attacker is trying to spoof the *ID List*. The *ID List* is generated with probability of match varying from 0 to 0.5. The same graph (Fig.7(b)) was plotted and was observed that increase in the probability of ID match does not change the nature of curve significantly.

Performance vs. Buffer size The second experiment was conducted to see how the number of buffers in the router queues affects the DDoS mitigation scheme. In this case the number of routers supporting PAS is kept constant, i.e. 8, but the numbers of buffers in the router queues were varied. The queue size was varied from 200 packets to 500 packets and the graph (Fig.8(a)) is plotted. It can be observed that with the increase in queue size the percentage of attacks reaching the CI increases exponentially. Thus smaller the queue size the better it is to overcome flood attack. But at the same time if the queue size is reduced too much there is a chance of legitimate packets being dropped. Thus the challenge is to fix the queue in such a way that it drops only the spoofed packet in case of DDoS.



(a) (b)

Fig. 8. Graph (a) between number of buffers in Queue Vs. percentage of Attack Packets reaching CI. Graph (b) between number of Queues in router Vs. percentage of Attack Packets reaching the CI

Performance vs. Number of Queues This experiment was conducted to see how the number of queues in the router affects our DDoS mitigation scheme. This experiment was conducted by varying the number of queues from two to eight. The number of buffers in the router and the number of routers supporting PAS is kept constant. The graph is plotted between the number of queues and percentage of attack reaching the CI. From the graph in Fig.8(b) it can be observed that with the increase in numbers of queues, the percentage of attack reaching CI decreases. Further there is no significant drop in percentage of attack packet reaching CI when the number of queues is four and when it is increased to eight. Thus four priority queues are considered to be optimal in this scenario.

6 Conclusion And Future Work

The PAS addresses one of the key limitations of the existing DDoS mitigation schemes viz. the lack of router based differential filtering. In order to bring properties of Invariability, Uniqueness and Unspoofability we have used a sequence

of router IDs as the basis of Identity. This sequence of router IDs helps in identifying a malicious packet with varying levels of certainty. The proposed metric called *Confidence Index* makes router-level differential filtering mechanisms possible. We have used multiple queues and a *Confidence Index* based scheduling algorithm to filter out attack packets from normal traffic at the routers. The performance analysis shows that we have achieved a success rate of 85% with the support of a very few routers in the path. Based on the study of Internet path stability we were able to derive the functional relationship between *Confidence Index* threshold and filtering accuracy. We were also able to predict the average number of ID renewals that can be expected in a system after it is deployed.

The limitation of PAS is that it is effective only against a particular type of DDoS attack viz. flooding with spoofed source IPs. Being a novel concept PAS paves way for multiple research directions. Firstly, one can analyze the impact of various weightage functions in the definition of *Confidence Index* and its subsequent effect on filtering accuracy. Further, a study of the impact of various queuing techniques on the mitigation process is to be conducted. A lot more insights can be derived by further analysis on the data collected using traceroute. To conclude, a lot more engineering and technological studies need to be carried out before PAS can be deployed widely over the Internet.

References

1. P. Ferguson and D. Senie, Network ingress filtering: Defeating denial-of-service attacks which employ IP source address spoofing, RFC 2827, May 2000.
2. J. Ioannidis and S. M. Bellovin, Implementing Pushback: Router-Based Defense Against DDoS Attacks, In Proc. Network and Distributed System Security Symposium, San Diego, CA, February 2002.
3. Y. Kim, W. Lau, M. Chuah, J. Chao, PacketScore: A statistical-based overload control against DDoS attacks, In Proc. IEEE INFOCOM 2004, China, March 2004.
4. J. Mirkovic, D-WARD: Source-End Defense against Distributed Denial-of-Service Attacks, PhD. Thesis, UCLA, August 2003.
5. Jelena Mirkovic , Peter Reiher, A taxonomy of DDoS attack and DDoS defense mechanisms, ACM SIGCOMM Computer Communication Review, vol. 34, no. 2, pp. 39 53, April 2004.
6. Vern Paxson, End-to-end routing behavior in the Internet, Conference proceedings on Applications, technologies, architectures, and protocols for computer communications, Palo Alto, California, United States, August 28-30, 1996, pp. 25-38.
7. T. Peng, C. Leckie and K. Ramamohanarao, Protection from Distributed Denial of Service Attack Using History-based IP Filtering, In Proc. of IEEE ICC 2003, Anchorage, AK, May 2003.
8. H. Wang, C. Jin, and K.G. Shin, Defense against Spoofed IP Traffic Using Hop-Count Filtering, IEEE/ACM Trans. Networking, vol. 15, no. 1, pp. 40-53, 2007.
9. Yaar, A. Perrig, and D. Song. SIFF: A Stateless Internet Flow Filter to Mitigate DDoS Flooding Attacks. In IEEE Symposium on Security and Privacy, 2004.
10. Abraham Yaar , Adrian Perrig , Dawn Song, Pi: A Path Identification Mechanism to Defend against DDoS Attacks, In Proc. of the 2003 IEEE Symposium on Security and Privacy, pp.93-107, May 11-14, 2003.