A proposal for new marking scheme with its performance evaluation for IP Traceback

S. MALLIGA and DR. A. TAMILARASI
Department of Computer Science and Engineering
Kongu Engineering College
Perundurai, Erode
Tamilnadu - 638 052
INDIA
mallisenthil@yahoo.com

Abstract: - Detecting and defeating Denial of Service (DoS) attacks is one of the hardest security problems on IP networks. Furthermore, spoofing of IP packets makes it difficult to combat against and fix such attacks. Packet marking is one of the methods to mitigate the DoS attack that helps traceback to the true origin of the packets. A hybrid packet marking algorithm, along with traceback mechanism to find the true origin of the attack traffic is presented in this study. The router marks the packets with inbound interface identifier of the router, but the novelty lies on the way it marks the packets. The stamping based on modulo technique and reverse modulo for the purpose reconstruction of attack path to traceback to the real source of the packets are proposed. The experimental measurements on the presented algorithm ensure that it requires less amount of time to mark and reconstruct the attack graph. It is also able to trace back to single packet, nevertheless it requires logging at very few routers and thus incurring insignificant storage overhead on the routers. The simulation study and the qualitative comparison with different traceback schemes are also presented to show the performance of the proposed system.

Key-Words: - DoS attacks, Logging, Modulo, Packet marking, Reverse modulo, Spoofing

1 Introduction

DoS and Distributed Denial of Service (DDoS) attacks have posed major security threats to the Internet. To define, a DoS is a malicious attempt to render a node or network incapable of servicing the legitimate requests. DDoS is the case in which one or more attackers coordinate the sending of enormous packets aiming at clogging the victim and the network. DDoS attacks come from various sources with different types of attacks. These attacks attempt to consume the finite resources like memory, computational power etc. in the network and also at the victim [1]. They would also result in heavy congestion on the network links thereby disrupting the communication among the users. Moreover, these attacks would exploit the inherent weaknesses in the IP protocol. One such is spoofing, which is, impersonating one’s IP address. Spoofing further complicates the detection of the DoS / DDoS attacks. More importantly, innocent host systems are involved for launching such attacks.

For instance, the DoS attacks have disrupted the Internet services and incurred heavy financial losses on the popular sites like Yahoo, CNN, Amazon etc. [2]. Such attacks are difficult to detect, prevent and traceback, but easy to implement. Thus the devastating effects of the problem have led to the development of many anti-DoS solutions.

The countermeasures that address the DoS attacks are deployed at different points on the Internet namely at the source-end, intermediate-network and victim-end. An ideal place to detect and filter the flooding attacks is at the network where they have been generated (i.e.) as close to the source of attacks as possible. The source-end defenses act as filters for the attacks and have advantages over the other two [3]. They include congestion avoidance over the network, small collateral damage, feasible deployment etc. The source-end defensive systems maintain the statistics of the outgoing traffic and use them to analyze and conclude about the ongoing attacks. But, because of the highly distributed nature of the attacks, detection near the source is cumbersome. Attackers may try to launch the attacks that resemble legitimate requests, thus there is no suspicious alert about the attack traffic can be generated.

The anti-DoS solutions that are deployed at the intermediate systems are useful for IP traceback which is the name given to the problem of reliably determining the origin of a packet. But, due to the stateless nature of the Internet, it is difficult to find the source of the spoofed packets. Existing approaches to handle this problem fall into logging and packet marking and they are tailored towards the detection of the flooding attacks by traceback.

Logging involves storing the packets on the routers which they come across and using data mining principles to find the path the packets traversed. This method can even trace back the attack long after it has been over. Since the process of logging incurs significant storage overhead on routers, a mechanism that stores packet digests rather than the packets themselves has been addressed by Snoren et. al. [4]. Although the hash based technique requires 0.5% of the total link capacity in digest table storage, the storage requirement at the routers, in general, would be too high.

One among the IP traceback approaches is packet marking, which lets the packet mark with path information during the forwarding of packets. The victim then, using the marked information, constructs the attack graph. The marking may be probabilistic or deterministic.

In Deterministic Packet Marking (DPM), a router marks every packet that passes through it and these packets are used to find the true source of the attack traffic. An example for DPM is presented in [5] and it uses the packet ID and flag fields for markings. When a packet enters into the network, it is marked by the interface closest to the source on the edge ingress router. Since single packet would not be sufficient to pass the IP address of the router, two packets are used to pass the IP address by splitting it into two parts of 16 bits each. The flag field tells which part of the IP address is carried into the packet. At the victim, a table that matches the source addresses into the ingress addresses is maintained. If there is no entry for the source address, then the victim would create an entry for the address and add the address of the ingress router into the table. This table would be used by the victim to reconstruct the attack graph.

Various approaches for Probabilistic Packet Marking (PPM) have been reported in the recent years [6] [7] [8] [9]. In PPM schemes, packets to be marked are selected using some fixed probability. The addresses of the routers are embedded into the selected packets. When the victim gets modest number of packets, it can reconstruct the attack graph. These schemes mainly aim at the detection of large scale DoS / DDoS attacks. In PPM schemes, the path reconstruction is based on multiple packet markings, but there is no guarantee that single packet would have enough marking for identifying the attackers. In addition, an attacker may also inject packets with erroneous information by spoofing the packets. Then such packets would cause confusion at the victim during path reconstruction. By nature, PPM solutions need large number of packets to converge on the attack path. Thus, a solution requiring more packets tends to be probabilistic in nature.

Besides the packet marking and logging, mechanisms such as link testing [10] and ICMP traceback [11] have also been proposed for IP traceback. The idea behind link testing is to start from the victim to determine the attack from upstream links and subsequently find the link that carries the attack traffic. This scheme requires the attack remain active while tracing back and is not suitable for post-mortem analysis. In ICMP traceback, every router, with low probability, sends special ICMP traceback messages that contain the information about the neighbor routers along the path towards the destination. During the attack, these messages are used by the victim for attack path construction.

Victim-end deployment of any defensive system protects the victim from the attacks and reduces the impact of such ongoing attacks by responding to them immediately. Many Network Intrusion Detection (NID) systems have been advocated for such deployment. These systems work on-line and detect real time intrusion attacks. Such intrusion detection systems use a set of features or models to analyze the incessant stream and discover the attack scenarios. A classification scheme for NIDs is presented in [12]. An example for such system is addressed in [13]. Most of the defensive systems are installed at the victim since it has been suffering from the attacks. But, such attacks can be detected only at bursting stage. By that time, the resources would have been consumed and hence the victim may not be able to protect itself from overwhelming packets.

In this study, a packet marking algorithm, which follows hybrid marking scheme to solve IP traceback problem is presented. As the packets travel through the network, they are marked with router information using modulo technique. Upon traceback request, reverse modulo is used to reconstruct the path traversed by the packets. In particular, this approach reconstructs the attack path with one packet and incurs very less overhead on the network and router. Hardly this method requires logging at routers, so the storage overhead on the routers is also significantly reduced. We extend the proposal, which we made in [14], to show the effectiveness of the same and also consider a new method to begin marking in order to achieve effectual results.

The paper is organized as follows: section 2 discusses about the need and the motivation behind IP traceback approaches. The details of the marking and attack path reconstruction procedures of the proposed system are given in section 3. The interpretation of experimental findings and performance of the proposed system along with qualitative comparison on different criteria with different promising packet marking systems are presented in section 4. The practical issues on the implementation of the system are discussed in section 5. A survey on the related works on packet marking is presented in section 6. The concluding remarks and directions for future work are drawn in section 7.

2 Need, assumptions and motivations behind IP traceback problem

Falsification of source address, called spoofing, makes it hard to find the true origin of a packet thus leading the DoS or one-way attacks[1] to go undetected. IP traceback is an ability to identify the sources of such attacks and institute preventive and protective measures.

The DoS attacks flood a network with the objective of degrading or refusing the legitimate users from accessing the resources or services on specific system. DDoS can cause more damages on the victim by employing a group of hosts, called zombies. To disguise the true locations, the attackers tend to spoof the packets, thus complicating the detection process.

The first proposal for IP traceback using packet marking was put forth by Savage et. al. and later augmented by others. The family of traceback schemes was motivated by one or more of the following assumptions [6] [9] [15]. These basic assumptions are made prior to the design of IP traceback system to establish practical guidelines.

2.1 Assumptions

The assumptions for the traceback schemes are as follows.

2.1.1 About attackers

- Attackers may generate any packets
- Co-ordination among multiple attackers to generate DoS / DDoS attacks
- Attackers may overwhelm the network traffic, thus the victim
- Attackers may send numerous packets
- Attackers may also launch the attacks that consist of single packet

2.1.2 About packets

- Path of the traffic is occasionally changed
- Packets may be re-ordered or lost
- Packets should not be made grow for the purpose of traceback

2.1.3 About routers

- Routers are resource constrained
- Routers are not widely compromised
- Routers would not generally generate forged packets
- Knowledge of map of upstream routers for reconstruction of attack path

2.2 Design goals

The security threats posed by the flooding attacks necessitate traceback schemes with the design goals presented in [8]. We present below the desirable characteristics of an anti-DoS system.

- Making available the information about the routers that are nearer to the attack source
- Identifying false information injected by the attackers
- Locating the source of attacking traffic rather than the first hop router
- Less number of packets to reconstruct the attack graph
- Low computational and storage overhead on the routers (i.e.) the routers should not demand extensive resources
- Less time to mark and reconstruct the graph of attack path
- Less cost and deployment time

[1] One-way attacks – it is a kind of attacks where it is not necessary to receive responses from the victim to continue the attacks.

- The network complexity should be kept as minimum as possible (i.e.) the packet size should not increase
- Must be a deployable solution

In the recent years, many counter measures for DoS attacks have been advocated that possess some of the above design goals and lack one or other. For instance, previous PPM systems require a considerable number of packets to find the attack path, at least in the order of 10's. There are schemes that require more number of bits to be marked in the IP packet. In contrast to this, there are logging schemes that require huge volume of memory for buffering the packets. Hence we need a scheme that provides almost all of the above design goals and we attempt to do so.

3 Design philosophy of MRT (Modulo / Reverse modulo Technique)

The objective of the proposed model is to keep track of the routers that contribute for marking the packets using a new marking scheme. Here, not every router is involved in marking. The marking is a hybrid scheme which implies that it may be probabilistic and/or deterministic (i.e.) the marking of the remaining routers through which the packets pass depends on the edge router of the source network. If the edge router decides to mark, then each of the upstream routers would contribute to the marking or otherwise not. The procedures for marking and reconstruction are presented below.

3.1 Marking procedure

This section describes about the marking procedures of edge and core routers.

3.1.1 At edge routers

An edge router, also called as access router, is a device that routes the data between one or more local area networks and a back bone network whether a campus or wide area network like Internet. As per proposal, the edge router of a network maintains a lookup table, called MACtoID table, which has physical addresses of the hosts attached to the network and equivalent numeric code for each of the physical addresses. When an edge router decides to mark an incoming packet, it fetches the code to be marked that corresponds to physical address of the host from the lookup table and encodes it into the packet.

For marking, the router requires 1 bit for indicating whether it marks or not, 1 bit in case of logging and few bits for marking code. The markings can be stamped on IP options field but it is computational intensive to append any data to a packet while forwarding which increases the size of the packet and thus ignoring the design goals. Hence, as in most of the marking schemes, we overload the 16 bits ID and 16 bits flags and fragment offset fields to carry the marking information. This may cause inconvenience for fragmented packets. But the researches on the Internet study show that only less than 0.25% of the packets are fragmented [16]. Fig. 1 shows the encoding into the packets by the edge router.

Marked field (1 bit)	Log field (1 bit)	Code from MACtoID table (30 bits)

Fig. 1. Marking by edge router

The algorithm for marking at edge router

```
For every packet,
 (i)    Let x be a random number between 0 and 1.
 (ii)   If x ≥ p, where p is marking probability
            (a) Use physical address of the sender to find the code to be marked from MACtoID table.
            (b) Set marked field.
            (c) Stamp the code into marking field.
 (iii)  Forward the packet to the next router.
```

3.1.2 At core routers

A core router is the router that transmits the data between other routers. A core router marks if only the packet has been already marked by the edge router. Otherwise it would simply forward the packets. The marking by the core routers is different from that of the edge router. These routers use modulo technique for marking. Every core router maintains a small table called MACtoInterface that contains the physical addresses of all of its hardware input or inbound interfaces and link numbers assigned to each of these interfaces. Here we would like to note that the number of hardware interfaces to a router also refers to the degree of that router. Hence we use interface and degree interchangeably. When a core router decides to mark, it consults the table to find the link number assigned to the inbound interface. To do so, it fetches the destination physical address from the packet and uses the address to find the link number assigned to the interface corresponding to it. The router then uses modulo method for recording the mark as,

New marking information = current marking information * number of interfaces on the router + the link number → (1)

The new marking information is recorded into the packet by overwriting the existing information. Fig. 2 shows encoding of marking information by the core routers.

1 (1 bit)	0/1 (1 bit)	current marking information * number of interfaces on the router + the link number (30 bits)

Fig. 2. Marking by core router

Due to the multiplication factor in the marking procedure, occasionally the marking field in the packet may be insufficient for the marking by a router. In such case, the marking field would have to be reinitialized to 0. But, before reinitialization, the router takes the digest of the packet and stores the digest along with old marking and log field into a log table. This is done to avoid the loss of markings made by upstream routers. A router that performs logging would set log field to help traceback process.

The algorithm for marking at every core router

```
For every packet,
If marked field is set
    (i)  Use MACtoInterface table and find the link
         number for the inbound interface on
         which the packet arrived.
    (ii) Calculate the new marking information as
         in (1).
    (iii) If the marking field is sufficient to hold the
         calculated value, then stamp it in the
         marking field.
       Otherwise,
         (a) Take the digest of the packet.
         (b) Log the triple (digest, old marking
         information, log field) into the log table.
         (c) Clear the marking field. Calculate
         and set the marking field as in (1) and
         set the log field also.
    (iv) Forward the packet to the next router.
```

3.2 Attack graph construction

Here we assume that there exist NID mechanisms like [17] at the victim to detect the DoS / DDoS attacks. Once the victim understands that it is under attack, it issues traceback request containing the marking information of the packet to be traced to the nearest upstream router that delivered the packet. The upstream router uses the reverse modulo to find the inbound interface of the traceback requested packet using the marking information found in the traceback request and then using the hardware address table at the inbound interface, the router finds the previous upstream router connected to that interface. Then the upstream router becomes the current router and traceback procedure is repeatedly performed till the edge router of the sending host is reached. When this is done, the victim would have found the routers crossed by the attack packet and would send a request to the edge router to find the physical address of the node that originated the attack packet. The procedures for the edge and core routers are presented below.

3.2.1 At core routers

```
Starting at the nearest router to which the victim is
attached to,
If the marking field is set, then the router has
involved in marking
While(true) {
(i) If the marking information in the traceback
request lesser than the number of degree of the
router and log field is 1, then
        (a) Set the inbound interface as the value of
        the marking information.
        (b) Take the digest of the packet and
        compare with log table to find marking
        by the previous router, set it in the
        marking field and also copy the log field.
 (ii) Else
        (a) Calculate the inbound interface that
        delivered the packet using the marking
        information in the traceback request as,
        inbound interface = mod (marking
        information, number of interfaces of the
        router)
        (b) Calculate the marking information sent
        by the upstream router as,
        old marking information = current marking /
                  number of degree of the router
(iii) Set the upstream router connected to the
    inbound interface as the current router and
    continue the process.
}
```

3.2.2 At edge router

(i) Use the marking information in the traceback request as index into MACtoID table.
(ii) Find the physical address that corresponds to the index.
(iii) Conclude the host associated with physical address is the compromised host and might have involved in attack.

.

4 Test results and performance evaluation

4.1 Simulation environment

We used network simulation package 'Network Simulator (NS2)' to simulate hundreds of nodes, conducted experiments by simulating DoS attacks along with the legitimate traffic and evaluated the performance of the proposed system. For the generation of the traffic, we used the simulation environment similar to the one described in [26]. The results of the simulation are compared against the results of different traceback approaches and presented below.

We have chosen the following parameters for evaluation of various traceback approaches from [20]. These include

i.. Convergence time (i.e.) the number of packets needed to reconstruct the full attack path and estimated time taken by the routers to do so.
ii. The estimated time for marking by every router (i.e.) the computational overhead in a router due to marking
iii. Storage overhead in a router due to logging
iv. Robustness of the traceback mechanism
v. Average size of the marking information

The first two parameters determine the fastness in response to the attack and reducing its strength. The third parameter represents the amount memory required at the routers as result of logging while marking. The last two parameters depict the false positive rate and sufficiency of the marking field on average case of the proposed model.

For the qualitative analysis, we use the following notations.

$(CT)_{sys}$	-	convergence time for the given system 'sys'
p	-	probability of marking
d	-	length of the attack path
EMS	-	Edge Marking Scheme
AMS	-	Advanced Marking Scheme
DLLT	-	Distributed Link List Traceback
PPPM	-	Pipelined Probabilistic Packet Marking
MDADF	-	Marking-based Detection And Filtering
Huff	-	Huffman Code for marking

4.2 Convergence time analysis

Convergence time is the time taken to reconstruct the attack graph. Given the packets marked by PPM, it is important to know the number of packets needed and time taken to find the IP addresses of all the routers along the attack path. The purpose of reconstruction is to determine the address of the host involved in attack or at least the address of the edge router nearest to the attacking host. It is necessary to verify that the path reconstructed is correct and complete. But it has been found in [20], that none of the PPM approaches provide a mechanism to verify the completeness of the reconstructed path. To verify, a large number of packets need to be collected. We compared the convergence time required by various traceback approaches and presented below.

The Edge Marking scheme by Savage et.al. [9] uses the edges sampled in the marked packets to construct the path of attack. Here the probability of receiving marked packets from the furthest routers is smaller than the routers nearer to the victim and hence the time to receive the samples from the furthest router is given by $1/p(1-p)^{d-1}$ for a router which is 'd' hops away. The factor ln(d) accounts for the small probability for the marked packets from furthest router than the nearer ones. Thus the number of packets needed to reconstruct the attack path of length 'd' is given by,

$$(CT)_{EMS} < \frac{\ln(d)}{p(1-p)^{d-1}}$$

For example, if p=2/10 and d=15, then the victim would require 85 packets approximately for attack path reconstruction.

For the DLLT and PPPM in [8], u is a parameter that helps in choosing the success probability of routers (i.e.) $p(x\text{-}d) \geq u$, where x is a random variable that represents the number of routers out of 'd' that succeeded in marking.

$$(CT)_{DLLT} \geq \frac{\log_{10}(1-u^{1/d})}{\log_{10}(1-p)}$$

$$(CT)_{PPPM} \geq \frac{\log_{10}(1-u^{1/d})}{\log_{10}(1-p(1-p)^{d-1})}$$

For the schemes in [7] [17], the convergence time is

$$(CT)_{MDADF} = (CT)_{Huff} = 1$$

Since the above two schemes are of DPM in nature, they require one packet to traceback to the true source of the attack.

The convergence time need for the proposed system is

$$(CT)_{MRT} = 1$$

This shows that, when compared with PPM approaches, the proposed system has less time to converge. But the systems presented in [7] [17] too have less convergence time of 1. But the estimated time needed for marking and reconstruction procedure is small in our system. We ran different approaches and measured the time needed for reconstruction procedure in all cases. The simulation results are tabulated below in table 1.

Table 1. Average estimated time for reconstruction.

Traceback schemes	**Convergence Time**	**Average estimated time for the reconstruction procedure (ms)**
AMS[2]	$<=4*10^3$	84.2
DLLT	in the order of 10s	0.379
PPPM[2]	in the order of 10s	519.2
MDADF[3]	1	5.956
Huffman Coding	1	0.30
MRT (proposed system)	**1**	**0.31**

It is clearly seen from the above table that the proposed system has small convergence time and needs less time to reconstruct the attack graph. Even though the system in [7] requires less amount of time to reconstruct the attack path than the proposed system, the difference is very marginal. We have also shown in the later sections that the proposed system performs better in all the other aspects.

4.3 Computational overhead on marking

This parameter defines the estimated time needed for marking by the routers along the path towards the destination. The complexity involved in marking procedure was evaluated by extending the network simulation for different traceback approaches. We ran the simulation by considering each of the different traceback approaches in turn and calculated the marking time taken by each of them. The result of this simulation is presented in table 2. The table shows the time needed for marking of every 1000 packets by the edge router and core routers and the time taken to cross the routers by the marked packets. The table projects the time taken by the three routers alone. In case of edge router, the time taken to mark a packet in PPM system is even though lesser that any other projected systems, it does not mark all the packets. But, MRT marks all the packets with very slight increase in time. But for the core routers, the MRT consumes less amount of time while competing with other methods. From, the table 2, it is apparent that the average time taken by the routers to mark packets is less in the case of our system when compared with other traceback approaches.

The graphical representation of the average time contributed for marking by the edge and core routers is shown in Fig. 3.

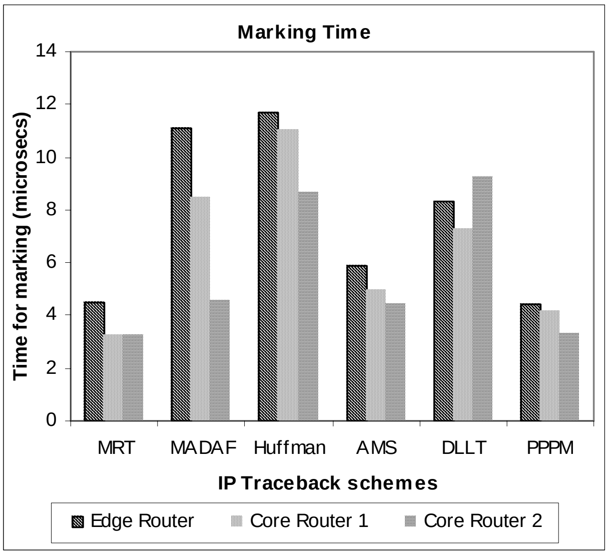

Fig. 3. Average time for marking at three routers for various traceback schemes

[2] These systems require the victim to keep on gathering the packets from different hosts to construct the attack graph

[3] This system requires echo reply packets from the suspected hosts in addition to the average estimated time for path reconstruction.

Table 2. Computational overhead of different traceback approaches

Traceback approaches	**For marking 1000 packets**			**Time taken for marking of 1000 packets by the routers (µs)**
	Edge router (µs)	**Core router 1 (µs)**	**Core router 2 (µs)**	
AMS	5.91	4.96	4.41	15.28
DLLT	8.35	7.28	9.26	28.93
PPPM	4.45	4.16	3.33	11.94
MDADF	11.08	8.44	4.57	24.09
Huffman Coding	11.71	11.00	8.66	31.37
MRT (proposed system)	**4.52**	**3.26**	**3.23**	**11.01**

4.4 Storage overhead

The amount of memory to be dedicated at every router is also an important factor that determines efficiency of a traceback scheme. We have analyzed the memory requirement of the proposed system and compared with that of different traceback systems. Average case analysis has been conducted to determine the requirement of memory at each router participating in marking. As it has been shown in [7] [18] [19] that, on average a packet needs to make more or less 32 hops and average path length is around 16. And also the average degree of router is slightly higher than three. When we implemented our system, we have found that only very few routers, hardly one to two, need to log the packets, that too for higher degree of routers. This is needed when the IP field is insufficient to hold the marking information.

If there are ‘n’ packets generated by a host, and ‘d’ routers along the path, then only ‘k’ out of ‘d’ routers are involved in logging. Thus the storage requirement for ‘n’ packets is given by ‘nqbk’ with the probability of marking ‘q’, which is lesser when compared to other systems as shown in table 3. In addition, every router maintains MACtoInterface table which has only very few entries (i.e.) an entry for every neighbor. So, the storage requirement for this table is negligible and immaterial. The storage requirements for various traceback approaches are listed below in table 3.

The variables in the table 3 represent as follows

b – marking field size

f – digest array size

p – percentage of different destination addresses at a router in a sample of ‘n’ packets

a – size of the IP destination address suffix used to index the buffer of size 2^a

Table 3. Memory requirement for various IP traceback approaches

Traceback schemes	**Memory requirement**
DLLT	nq(b+f)d
PPPM	nqpf+57*2^a
Huffman Coding	nqbd (where q is always 1)
MRT (proposed system)	**nqbk**

4.5 Robustness

A traceback system is said to be robust if it yields low false positives. False positive rate can be reflected by the number of false nodes in the attack graph generated that is, identifying the legitimate nodes as attack nodes. To show the accuracy of the construction of attack graph, we measured the number of false positives along the reconstructed attack graph and compared with two logging based schemes advocated in [4] [21]. The upper bound of the average number of false nodes (F_p) in the reconstructed attack path in hash based approach [4] is given as,

F_p=ndP/(1-dP) where n is number of hops in the attack path, d is the average degree of routers and P is the false positive rate.

The F_p for the hybrid approach in [21] is as,

F_p = (n/2)(2dP/(1-2dP)) where n/2 indicates that the hybrid scheme requires logging by alternate routers.

For the proposed system MRT, the F_p is given as,

F_p = kdP/(1-dP) where k is the number of routers involved in logging and k << n/2.

To illustrate that F_p is small when compared to other two systems, we provide an example. If P is 1%, n is 16 and d is 4, then the system presented in [21] results in false positive rate of 0.69 and the hash based approach yields a false positive rate of 0.67. But the proposed system required logging by only one router along the path and hence yields the false positive rate of 0.042 only.

4.6 Average length of marking field

Depending on the degree of routers, the length of the marking information varies (i.e.) as there is increase in degree, so is the size of the marking information. In the table 4, we present the average length of the marking code for few degrees of routers after marking by themselves.

Table 4. Average length of marking codes for few average degrees of routers

Average degree	Average length of marking field (in bits)
2	1.00
3	1.33
4	1.5
5	1.8
6	2

We compared the average length of the code required by the proposed model with the length required by the system in [7], since this scheme also requires marking by routers. The result of comparison is presented in the Fig. 4.

As has been mentioned in [7], the average length of the marking code with average degree 3 and hop distance of 16 is 24.95 with equal distribution of packets on all interfaces of routers. But in our model, we found that it is 23.3, as shown in Fig. 5, which shows that MRT requires lesser bits than the Huffman way of coding. In the Fig. 5, we also show the increase in average length of the marking field for two different degrees of routers with hop distance of 16.

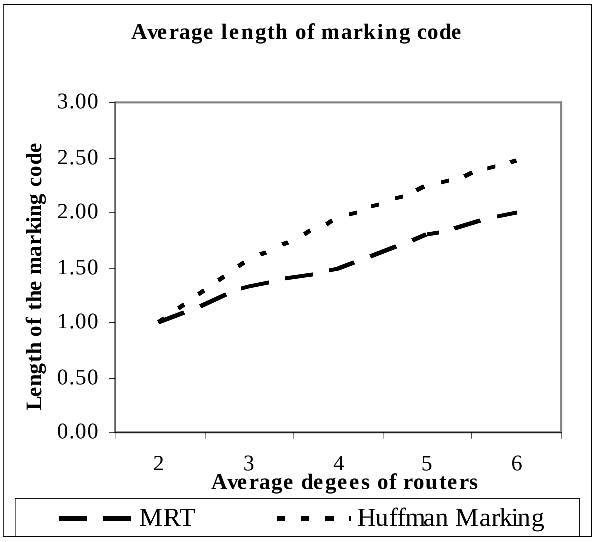

Fig. 4. Average length of marking field for Huffman marking Vs MRT

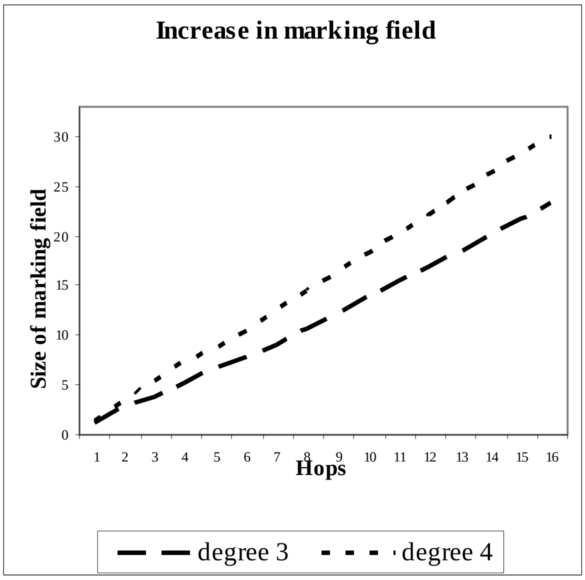

Fig. 5. Increase in length of marking field

In the Fig. 5 with average degree 3 and average path length of 16, the average size of the marking information is 23.3, which does not require logging at all. The Fig. 5 also shows that even for the average degree 4, the system does not need logging.

The table 5 summarizes the comparison of the proposed system with different DPM and PPM schemes that are especially designed for IP traceback. The evaluation results are based on the simulation on network simulator. The merits and demerits of different schemes are also presented in the table 6.

Table 5. Qualitative comparison of the proposed system with other traceback systems

Traceback Approach	**No. of packets needed for tracing**	**Complexity**	**Robustness**	**Marking Prob.**	**Logging**		**Knowledge of map of routers**	**Granularity of traceback**
					At network	**At victim**		
MRT (proposed system)	**1**	**O(1)**	**High**	**1**	**Low**	**Low**	**Not needed**	**Attacking Host**
CEMS	in the order of 1000s $(4*10^3)$	$O(l*n^8)$	Low	0 to 1	None	Very High	Not needed	Nearest Router
AMS	$< 4*10^3$	O(l*n)	Medium	0 to 1	None	Very High	Needed	Nearest Router
DLLT	in the order 10s	O(l*n)	High	0 to 1	Moderate	Moderate	Not needed	Nearest Router
PPPM	in the order 10s	O(l*n)	High	0 to 1	Moderate	Moderate	Not needed	Nearest Router
Huffman	1	O(1)	Moderate	1	Low, but higher than MRT	Low	Not needed	Nearest Router
MDADF	1	O(1)	High	1	None	Very High	Not needed	Nearest Router

Table 6. Merits and demerits of different approaches

Traceback Approaches	**Pros**	**Cons**
MRT (proposed system)	➢ **Single packet traceback** ➢ **Less storage and computational overhead of routers** ➢ **Support for DDoS** ➢ **Robustness** ➢ **Spoofed traffic can be dropped at source end after traceback request**	➢ **Irresistant to MAC spoofing**
CEMS	➢ High scalability ➢ Low network processing overhead	➢ Difficult for DDoS support
AMS	➢ Robust ➢ Low network and router overhead	➢ Design is limited to 32 hops ➢ Knowledge of map of upstream routers
DLLT	➢ Keeps the size of the packet from growing while preserving marking information ➢ Reduced number of packets for traceback	➢ Long term storage of packet digest at intermediate routers ➢ As path length increases, number of packets needed for traceback also increases
PPPM	➢ Reduced number of packets for traceback	➢ High processing overhead ➢ Requires 57 bits for marking ➢ Logging at every router ➢ As path length increases, number of packets needed for traceback also increases
Huffman	➢ Less computation overhead for traceback ➢ Support for DDoS attacks	➢ Overhead on routers due to marking
MDADF	➢ Single packet traceback ➢ Support for DDoS attacks ➢ Less overhead on routers	➢ Computational and storage overhead at victim ➢ Echo messages generate traffic on network

5 Observation and discussion on the issues of implementation

In this section, we discuss the practical issues that determine the success of the proposed system.

5.1 The need for logging

One of the design goals of any IP traceback system is to reduce the amount of storage needed at the intermediate routers. The system presented here is so carefully designed that it requires logging at only very few routers. Logging is needed only if the marking field is not sufficient to hold the marking. As it has been shown earlier, the proposed system faces this challenge occasionally and requires logging only at one or two routers. Thus storage overhead incurred by the proposed system is comparably lesser than other systems.

5.2 Packet marking

The other issue is the way the marking is begun with the physical address of the host that sent the packet. Since the physical address of the host is used by the edge router to find the host involved in spoofing during the path reconstruction, one may argue that it is possible to determine the spoofed packets at the time they leave the network via the edge router by consulting the ARP protocol. But this requires checking of every packet leaving the source network for having the correct IP and physical addresses and consumes quiet large amount of time. But, the traceback is initiated only for the requested packets. Hence the edge router does the process of lookup only on request to find the IP where the packet is originated. As it has been discussed already in section 4.3, the packet marking procedure along the path requires less amount of time while compared with other marking schemes.

5.3 Memory requirements at core routers

Every router needs to maintain a table called MACtoInterface, that contains physical addresses of every interface of the routers and a link number assigned to each of them. As discussed in the study [19], since the average degree of every router is just more than three, the size of the table is negligible. In the case of the link failure or addition of new links, that is, when the interfaces of the router change, the table also needs to be updated to reflect the changes appropriately. By assigning minimal link numbers to the links that shows more traffic, the packet logging can even be reduced to nil.

5.4 Other issues

As it has been mentioned already, NS2 has been used for simulation of both attack and legitimate traffic. The performance of the proposed system can be better tested by using test data synthesizer suggested in [25]. The data synthesizer provides simulated DDoS flooding attacks to evaluate the performance of any anti DoS system. We have not tested our system using data synthesizer and planned to take up for further research.

6 A case study on related work

To mitigate the flooding attacks, it is important to identify the common features of these attacks to distinguish them from normaltraffic. But such identification would be difficult since these attacks vary significantly. In [23], guidance for such classification is presented. Many attacking tools are being employed for performing DoS attacks and these include TFN, TFK2k, Trinoo [24]. These tools employ a master to control daemons on the compromised systems to launch the attacks.

One of the inherent weaknesses of the IP protocol is that the source address of the packet can be spoofed in malicious attempts. There is no provision to detect the origin of such spoofed packets. Various solutions have been addressed to thwart the spoofed flooding attacks. One possible way to deal with DoS attacks is to traceback the attackers and prevent them from doing so. To find the path followed by the attack traffic, the traceback approaches rely on the routers and find the attack source. One of the traceback techniques is packet marking. In packet marking, a router either deterministically or probabilistically marks its identification information during the packet forwarding.

The main idea of PPM is to mark the packets probabilistically as they traverse through the routers. This is done with the belief that after having received ample number of packets, the path can be reconstructed using the marking information present in the packets. Many PPM schemes have been tailored to address traceback problem. The scheme proposed in [9], namely Compressed Edge Fragment Sampling, marks the XOR of the two IP addresses that make up an edge. The resulting XOR is fragmented in eight parts. Each fragment is marked in a packet. The packet also carries offset of the fragment. The victim uses these fragments to reconstruct the attack path. This scheme requires orders of thousands of packets to reconstruct the attack path. The scheme proposed by Song et. al. [6]

marks the XOR of hash value of the edges. The distance between the marking router and the victim is also carried in the packets. Using these details, the attack path is found. This method relies on the knowledge of map of upstream routers.

G.Manimaran et. al. proposed two novel schemes that employ marking and logging namely Distributed Linked List Traceback (DLLT) and Probabilistic Pipelined Packet Marking (PPPM) [8]. DLLT is based on a store, mark and forward approach. It employs both packet marking and storage schemes. When a router gets a packet, it would mark the packet with some fixed probability. If the packet has been already marked, the router would store the marking information in the packet before remarking. DLLT maintains a digest array and marking information table at every router. The fraction of the traffic is logged at each router and needs significant amount of memory. DLLT also requires long term storage at each router. Few hybrid mechanisms involving both packet marking and logging have been addressed. One such is presented in [21].

In PPPM, the marking information that belongs to certain packet is transferred by propagating it from one router to another using subsequent packets traversing to the same destination. A PPPM enabled router requires 57 bits for transferring marking information, which may practically be unfeasible.

In the proposal by Y.Chen et. al. [17], a DPM has been addressed which is based on a firewall running at the victim and filters the attack traffic. This scheme requires no logging at core routers, but consumes high volume of storage at the victim. Otherwise it creates heavy traffic on the network. The success of scheme depends on the filter table at the victim, which contains the source IP addresses and consistent markings for these addresses. If markings in the packets do not match with the markings in the table, then the packets are assumed to be spoofed and hence dropped. To drop the attack traffic before reaching the victim, pushback technique is implemented. This necessitates logging at the core routers.

In summary, existing solutions are found to be good, but they too have drawbacks as discussed earlier. Hence we attempt to propose a more comprehensive solution that overcomes the drawbacks, even if not all, at least most of them. In this paper, a new marking algorithm, which is deterministic and/or probabilistic, is proposed. We borrowed the idea of assigning unique number to each of the interfaces of a router from [22] and used this information for marking rather than the IP addresses. The proposed scheme requires less time to mark the packets and reconstruct the attack graph. The system possesses the convergence time of 1. When compared with other packet logging methods, the system needs hardly logging to be performed at very few routers. While considering the practical difficulty in implementing, it may cause additional burden on router in terms of marking. But this is unavoidable for any marking scheme. However, it has been shown that the proposed system requires less time to mark than most of the other packet marking approaches.

7 Concluding remarks and scope for future work

DoS attacks create imminent threat to the legitimate users on IP networks. IP Traceback is a mechanism that has evolved as an effective solution to combat against these flooding attacks. Although many IP traceback approaches exist, they suffer from one or more of the problems as discussed earlier. In this study, we proposed a marking scheme, based on modulo technique, which lets the router mark the interface codes rather than IP addresses itself. The objective of all PPM schemes is to track back to the attacking origin, but these schemes can reveal the edge router attached to the attacking source. But our scheme has extended to the limit of tracking the attacking host also. Although egress filters deny the spoofing of IP addresses beyond the network, they would fail if the attacker resides in the source network. A disgruntled user may reside and launch the spoofing attacks from the source network. But our system is capable of handling the malicious attempt made from the network and fixing it too. The presented approach stores the entire path traversed in a single packet and thus leads to less convergence time to find the attack path at the victim. The method also incurs less computation and storage overhead on the core routers which involved in marking. Thus the scheme possesses many of the designed goals presented in section 2. But the proposed scheme is by no means immaculate. Still there are new directions to proceed further. One of the issues to be considered is the need for efficient data structures to keep track of packets at the logging routers. Since the marking begins with the physical address at the edge routers and if the attacker is aware of this, they may tend to spoof the MAC also. Under such situation, the scheme might fail. Hence it is necessary to keep the system running on MAC spoofing also. The proposed system can also be integrated with source end defensive systems, wherein the spoofed traffic from the network is dropped at the edge router itself. Any spoofed traffic that escaped from these defenses would be marked and traced back. If such trace traffic is found to be spoofed, further such traffic can be dropped at the source-end. The other issue for further study is the behavior of the proposed system in case of Distributed Reflection DoS. However, we believe that our system would be one more step towards the designing a network which is resistant to DoS / DDoS attacks.

References:

[1] CERT Coordination Center, Denial of Service attacks, http://www.cert.org/tech-tips/denial-of service.html.

[2] S. Tanachaiwiwiat and K. Hwang, Differential packet filtering against DDoS flood attacks, *in the proc. of ACM conference on Computer and Communications Security (CCS),* Washington DC, 2003.

[3] Prateek Mittal, Defense against DDoS attacks, Seminar Reoprt, IIT, Guwahati, 2005.

[4] A.C. Snoren, C. Partridge, L.A. Sanchez, C.E. Jones, F. Tchakountio, B. Schwartz, S.T. Kent and W.T. Strayer, Single-packet IP Traceback, *IEEE/ACM Transactions on Networking,* Vol. 10, No 6, 2002, pp. 721-734

[5] A. Belenky and N. Ansari, IP traceback with Deterministic Packet Marking, *IEEE Communications Letter,* Vol. 7, 2003, pp. 162–164.

[6] D.X. Song and A. Perrig, Advanced and Authenticated marking scheme for IP traceback, *in the proc. of 20th Annual Conference of IEEE Communications and computer Societies (INFOCOM 2001),* 2001, pp.878- 886.

[7] K.H.Choi and H.K.Dai, A Marking Scheme using Huffman Codes for IP Traceback, *in proc. of 7th International Symposium on Parallel Architectures, Algorithms and Networks (SPAN'04).*

[8] B.Al-Duwari and M.Govindarasu, Novel hybrid schemes employing packet marking and logging for IP traceback, *IEEE Transactions on Parallel and Distributed Systems,* Vol. 17, No. 5, 2006, pp. 403-418.

[9] S. Savage, D. Wetherall, A. Karlin, and T. Anderson, Practical network for IP Traceback, *IEEE / ACM Transactions on Networking*, Vol. 9, No. 3, 2001, pp. 226-237.

[10] H.Burch and B.Cheswick, Tracing anonymous packets to their approximate source , *in the proc. of 14th Systems Administration Conference,*New Orleand, USA, 2000, pp. 313- 322.

[11] S. M. Bellovin, Internet Draft, 2001.

[12] Iosif-Viorel Onut and Ali A. Ghorbani, A Feature Classification Scheme For Network Intrusion Detection, *International Journal of Network Security*, Vol. 5, No. 1, 2007, pp. 1-15.

[13]Gianni Tedesco and Uwe Aickelin, Data Reduction in Intrusion Alert Correlation, *WSEAS Transactions on Computers*, Issue 1, Vol. 5, 2006, pp. 186-193 [12] S.Malliga and A. Tamilarasi, A defensive mechanism to defend against DoS/DDoS attacks by IP Traceback with DPM, *in the proc. of ICCMA*, 2007, pp. 115-119.

[15] D.Dean, M.Franklin, and A. Stubblefield, An algebraic approach to IP traceback, *in the proc. of Network and Distributed System Security Symposium (NDSS' 01)*, 2001.

[16] I. Stocia and H. Zhang, Providing guaranteed services without peer flow management, *in the proc. of SIGCOMM'99*, 1999, pp. 81-94.

[17] Y. Chen, S. Das, P.Dhar, A.E. Saddik and A. Nayak, Detecting and preventing IP-Spoofed distributed DoS attacks, *International Journal of Network Security,* Vol. 7, No.1, 2008, pp. 70-81.

[18] W.Theikmann and K.Rothermet, Dynamic distance maps of the Internet, *in the proc. of 19th Annual Conference of IEEE Communications and Computer Societies (INFOCOM2000),* 2000, pp. 275-285.

[19] C.R. Palmer, G. Siganos, M.Faloutsos, C.Faloutsos, and P.B.Gibbons, The connectivity and fault-tolerance of the Internet topology, *in the 2001 workshop on Network-related Data Management; in cooperation with ACM Special Interest Group on Management of Data / Priniciples of Database Systems*, 2001.

[20] V. Kuznetsov, A. Simkin, H. Sandstrom, An evaluation of different IP traceback approaches, *in proc. of Fourth International Conference on Information and Communications Security*, 2002, pp. 37-48.

[21] C.Gong and K.Sarac, IP traceback based on packet marking and logging, *in the proc. of ICC*, 2005, pp. 1043-1047.

[22] R. Chen, J.-M. Park, and R. Marchany, RIM: Router interface marking for IP traceback, *IEEE Global Telecommunications Conference (GLOBECOM '06),* 2006, pp 1-5. [23] Marvin Oliver Schneider and Jacques Calmet, A Logical Fibering Approach to Denial of Service Prevention, *WSEAS Transactions on Systems*, Issue 3, Vol. 6, 2007, pp. 570-576.

[24] A. Kulkarni and S. Bush, Detecting DDoS using Kolmogorov Complexity Metrics, *Journal of Network and Systems Management*, Vol 14, No. 1, 2006.

[25] Ming Li and Wei Zhao, A principle of a data synthesizer for performance test of anti-DDoS flood attacks, *in the proc. of the 7th WSEAS Int. Conf. on Applied Computer and Applied Computational Science (ACACOS '08)*, 2008, pp. 254-258.

[26] Ming Li, Jun Li, and Wei Zhao, Simulation Study of Flood Attacking of DDoS, *in the proc. of IEEE 3rd International Conference on Internet Computing in Science and Engineering, IEEE CS Press*, 2008, pp. 28-29.