Scalable Hash-based IP Traceback using Rate-limited Probabilistic Packet Marking

Minho Sung, Jason Chiang, and Jun (Jim) Xu

***Abstract*— Recent surveys show that DDoS attack is still one of the major threats to the Internet security. Many techniques have been proposed to trace the origin of attacking packets, known as IP traceback problem, using either hash-based packet logging or probabilistic packet marking. However, both approaches have scalability problems under the heavy DDoS attacks in terms of the space and computational overheads. In this paper, we propose a novel scalable IP Traceback scheme by utilizing the advantage of both packet logging and marking to balance the overheads at routers and at the victim, hence scalable for both sides. The baseline idea of our approach is to sample a very small percentage (e.g., 1%) of packets at the routers, and save the digests of only sampled packets. At the same time, the routers mark their signature using very simple marking scheme into the marking field of sampled IP packets to send out the "information of logging" to the victim in probabilistic way to help the traceback procedure. We also propose a heuristic technique to improve the performance of the marking scheme. In the result, the number of attacking packets the victim should collect for the traceback procedure to achieve high level of traceback accuracy is much less than the numbers in previous PPM schemes, and also the computational and storage overhead in routers are much less than previous packet logging approach.**

I. Introduction

In recent years, large number of Distributed Denial of Service (DDoS) attacks have been launched to various targets such as high-profile commercial websites [1], IRC chatting servers [2], Internet infrastructure (root DNS servers) [3] or contents distribution systems [4]. Despite the large efforts to defend against, recent surveys [5], [6] show that DDoS attack is still one of the major threats to the Internet security. One of the reasons that keep the frequency of the DDoS attack high is that tracing and punishing the attacking sources are very hard under the current Internet. This is because the Internet only uses destination IP address to make the routing decision and does not maintain any information about the routed packets. Therefore, when the attackers use the forged source IP addresses in the attacking packets, there is no way to find the origins of the attacks.

Recently, many techniques are proposed to trace the origin of an IP packet, known as IP traceback problem. Most of them requires the Internet to maintain certain level of inevitable additional information about the routed packets. These algorithms can be categorized into two groups according to the place to maintain the additional information. First approach is to use the small limited marking field in IP packet header [7], [8], [9]. In this approach, routers randomly select the packets and mark partial path information into the marking fields in selected packets (referred as PPM(Probabilistic Packet Marking) approach). Then the victim can reconstruct the attack paths using the information in marking fields of many attacking IP packets received. Second approach is to store the information about the routed packets in each router's local storage [10] (referred as packet logging approach). Each router stores packet digests in the form of Bloom filters at its local disks, and this information is used for traceback procedure later.

However, both approaches have scalability problems under the heavy DDoS attacks in terms of the overheads in the algorithms [11]. PPM approach requires the victim prohibitively high number of attacking packets to be collected and high computational overhead for traceback procedure, mostly due to the high complexity of coding/decoding of the path information using limited marking space. On the contrary, packet logging approach requires prohibitive computational and storage overhead to the routers with very high speed links.

A. Our new scalable approach

In this paper, we propose a novel IP traceback scheme which balances the overheads at routers and at the victim, hence scalable for both sides. The scheme presented in this paper is a new approach from our previous work on scalable IP traceback scheme [11], which can achieve high accuracy of the IP traceback and can scale very well to a large number of attackers. Our previous approach is to sample and log a small percentage (e.g., 3.3%) of packets using an efficient sampling scheme utilizing only 1 bit flag in IP header. In the new approach, we investigate the possibility to further decrease the sampling and traceback overhead in the cost of using more marking space (e.g. 16 bits) in IP header.

The baseline idea of our approach is to sample a very small percentage of packets at the routers, and save the digests of only sampled packets. At the same time, the routers mark their signatures using very simple marking scheme into the marking field of sampled IP packets to send out the "information of logging" to the victim in probabilistic way. Once the victim collects at least one signature from each of the routers on attacking paths in high probability after receiving enough number of attacking packets, it can initiate the traceback procedure using recursive query to the upstream routers[1].

We also propose a heuristic technique to further improve the performance of our approach or any PPM-based scheme, observing the following intrinsic problem at any PPM-based scheme. The probability that a marking from a router reaches to the victim without being overwritten by any other routers on the path is geometrically smaller the further away the router is from the victim. In the result, in any time period, the victim receives much larger number of marks from the nearer routers

Minho Sung and Jun (Jim) Xu are with College of Computing, Georgia Institute of Technology, Jason Chiang is with Telcordia Technologies. E-mails: {mhsung,jx}@cc.gatech.edu, chiang@research.telcordia.com

[1]This procedure is similar with the ones in [10], [11], but different in the selection of the set of attacking packets for the traceback query.

than the ones from the further routers, so the time to collect all the marks from the furthest router is the main bottleneck of PPM schemes. We introduce a novel heuristic technique to improve the performance as following. We use a special simple data structure at the routers to control the rate of marking to a certain destination during a given time epoch. Using this technique, the nearer routers may decrease the rate of their marks to the victim while the further routers may have more survival probability of their marks to the victim. One salient point of this approach is that more performance improvement can be achieved as the intensity of a DDoS attack is increasing, because the gain of the data structure is also increasing.

Extensive simulation results based on real-world network topologies show that our new approach can improve the performance of the traceback drastically. First, the routers only have to maintain very small amount of information about the packets going through them. With a very low sampling rate (e.g. 1%), the storage and computational cost becomes much smaller than existing packet logging approach or our previous scheme. Second, the average number of attacking packets required for traceback procedure to achieve high level of traceback accuracy can be much less than existing PPM-based schemes or our previous approaches. This improvement comes from the simplicity of our marking scheme. Each router transmit only one piece of information using marking field without any information fragmentation which causes the significant traceback overhead to the victim. The effectiveness of our heuristic technique is the other reason of this improvement. Third, the overhead for the traceback query message transmission at the victim can be drastically decreased. In our previous scheme, the size of a query message to a router is up to a few megabytes. However, in the new scheme, this size can be decreased to one or two kilobytes due to the packet selection procedure, which means that a few number of IP packets is enough to transmit a query message. Finally, it can eliminate the marking field spoofing problem, which is a serious drawback in previous PPM approach allowing some false marking information from the attackers, by comparing victim's set of attacking packets as an evidence with the logging information at routers.

The rest of this paper is organized as follows. An overview of the proposed algorithm is presented in the next section. This is followed by a detailed description of the design in Section III. Parameter tuning of the proposed schemes and the analysis of traceback overhead is presented at Section IV. Evaluation of algorithms using simulations on traffic traces is presented in Section V. We review related work in Section VI and conclude the paper in Section VII.

II. Overview

In this section we first present an overview of our new scalable traceback approach. Then, we explain an intrinsic problem of PPM-based approach, and propose our new heuristic technique to overcome the problem for further improvement of the traceback performance.

A. Overview of the scheme

In this paper, we propose a new IP traceback scheme that has scalable algorithms at both the routers with high link speeds and the victim under the heavy DDoS attack. We assume that each router has storage resource to save the digests of the small percentage (e.g., 1%) of traffic. We also assume that the 16-bit IP fragmentation field in packet IP header can be used to store the mark of the routers throughout this paper. [2]

As introduced in Section I, the baseline idea of our approach is to require Internet routers to record Bloom filter digests of packets going through them, like [7]. However, unlike [7], which records 100% of packets, our scheme only samples a small percentage (e.g., 1%) of packets at the routers, and save the digests of only these sampled packets.

The attack graph reconstruction can be initiated by the victim using recursive queries to the upstream routers, similarly to [7]. However, it is no longer possible to trace one attacker with only one packet because each router has bloom filter digests for different set of sampled packets. Instead, in our scheme, for sending the traceback query to a router, the victim uses a set of packets which have high possibility that they are logged at the router being queried, and different set of packets will be used for querying to different router. To help victim's packet selection, routers send out the information about their logging of sampled packets using the marking fields of those sampled packets. This is done with very simple marking method. The routers use the hash value of their IP addresses as their signature, and mark this value into the marking field of all sampled packets.

Once received a set of the attacking packets which contains at least one signature from each of the routers on attacking paths in high probability, the victim can initiate the traceback procedure using recursive query to the upstream routers as follows. When the victim send a traceback query to a router R, it first select the small subset of packets, which contains the signature of R, out of the collected set of attacking packets. Then the victim asks to router R whether it has seen any of those packets, and R answer the the question using the Bloom filter test. If the answer is yes, the victim will include the router R to the attack graph reconstruction.

B. New heuristic technique to improve Probabilistic Packet Marking

In existing PPM approach, every router uses the same marking probability p. This gives the simplicity of the implementation, but it has an intrinsic problem as follows. The probability that a mark from a router is reached to the victim not being overwritten by other routers on the path is $(1-p)^{d-1}$, where d is the number of hops from the router to the victim. So, a router is expected to mark $\frac{1}{p(1-p)^{d-1}}$ packets in average to deliver its mark to the victim. This number of required packets geometrically increases as d increases. In the result, in any time period, the victim receives much larger number of marks from the nearer routers than the further routers.

[2] The IP fragmentation field has been reused in the PPM-based IP traceback schemes. The "backward compatibility" issues has been discussed in [7].

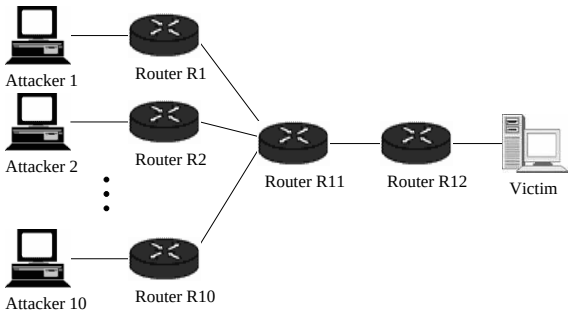

Fig. 1. An example of attack graph

Probabilistic Logging and Marking procedure at router R (given sampling rate p):
1. **for** each packet pkt
2. **if** $(random(0,1) \leq p)$ **then**
3. Bloom_filter_insert($pkt.invariants$);
4. $pkt.hopcount$:= 0;
5. $pkt.label$:= $f(R.IP)$;
6. **else**
7. Increase $pkt.hopcount$;

Bloom_filter_insert(x): (given # of hash functions k):
8. Set bits $A[h_1(x)]$, ..., $A[h_k(x)]$ to 1;

Fig. 2. Probabilistic Logging and Marking (PLM) scheme

This phenomenon makes the total number of attacking packets for the victim to collect all kinds of the marks very large. Note that it is important for the victim to collect at least one instance of each mark, and the duplicate occurrences of the same mark does not help but only degrades the traceback performance. This problem is aggravated under the situation of heavy DDoS attack because nearer routers will deliver more attacking traffic than the further routers to the victim, hence mark more packets. Figure 1 shows an example of attack graph when ten hosts attack the victim simultaneously. If we assume that all attackers send the same rate T of attacking traffic to the victim, the victim will receive $10 \cdot T \cdot p$ marks from the router R_{12}, but only $T \cdot p \cdot (1-p)^2$ from each of routers $R_1, \cdots, R_{10}$, which is 11 times less rate of marks when we use 0.05 as the value of p (typical value in existing PPM schemes).

In our scheme, we propose a new heuristic technique to improve the performance of the marking scheme as follows. We can divide the rate of mark $T \cdot p \cdot (1-p)^2$ from each of routers $R_1, \cdots, R_{10}$ in upper example into two parts. First part is $T \cdot p$, which represents the rate of initial marking from each router. Second part is $(1-p)^2$ which is the probability that the initial mark is overwritten by the other routers on the path to the victim. To increases the rate of mark reached at the victim, we use higher value of p (e.g., 0.2) than the previous PPM scheme, to increases the initial mark rate $T \cdot p$. Then, to decreases the probability that this initial mark is overwritten, each router limits the number of marking to a certain destination to only one during a certain time window. For this purpose, routers maintain a bit array as the set of flags. We will refer this very simple data structure as "rate-limiter". When a packet arrives, with probability p, router hashes the destination IP to get an index of one flag. If selected flag is not set, the router logs and marks the packet, and set this flag. For the next sampled packets to the same destination within the same time window, no logging or marking will be performed. At the end of time window, the flag will be reset. In the result, the rate of duplicated marks from the nearer routers will be much decreased, and this can give high significant improvement of the performance by improving the chance of the marks from the further router to reach the victim without being overwritten. Note that increasing the value of p does not mean the increment of overall overhead, because we can control the average rate of marking, which is much less than the value of p, by configuring the parameters of the rate-limiter. More detailed description of this procedure will be presented at the next section.

III. Design

In this section, we present our basic scheme and reconstruction procedure in Section III-A and Section III-B, respectively. Then we explain the advanced marking scheme using rate-limiter in Section III-C in detail.

A. Probabilistic Logging and Marking (PLM) scheme

Our Probabilistic Logging and Marking (PLM) scheme is presented in Figure 2. As described in Section II, all particpating routers sample the packets using predefined marking probability p. The sampled packets will be saved in the form of Bloom filter digests, and also will be marked to send the router's signature to the victim. Detailed description about packet logging and marking procedure is following.

Packet logging: For every sampled packet, the router records the packet digests (line 3 and 8 in Figure 2) using a space-efficient data structure known as Bloom filter [12], like in [10]. A Bloom filter representing a set of packets $S = \{x_1, x_2, \cdots, x_n\}$ of size n is described by an array A of m bits, initialized to 0. A Bloom filter uses k independent hash functions $h_1, h_2, \cdots, h_k$ with range $\{1, \cdots, m\}$. During *insertion*, given a packet x to be inserted into a set S, the bits $A[h_i(x)]$, $i = 1, 2, \cdots, k$, are set to 1. To *query* for a packet y, i.e., to check if y is in S, we check the values of the bits $A[h_i(y)]$, $i = 1, 2, \cdots, k$. The answer to the query is *yes* if **all** these bits are 1, and *no* otherwise. Same as in [10], the first 24 invariant bytes of an IP packet is used as hash input including the invariant portion of the IP header (16 bytes) and the first 8 bytes of the payload.

A Bloom filter guarantees not to have any false negative, i.e., returning "no" even though the set actually contains the packet. However, it may contain false positives, i.e., returning "yes" while the packet is not in the set. The *capacity factor*, denoted as c, of a Bloom filter is defined as the ratio of m to n. In this paper, we assume the Bloom filter at each router is paged to disk before c decreases to $k/\ln 2$. Then according to [12], the false positive rate of the Bloom filter is no more than 2^{-k}.

Encoding procedure at router R
(given sampling rate p):
1. **for** each packet pkt
2. **if** $(random(0,1) \leq p)$ **then**
3. $index := f_1(pkt.Dest\_IP)$;
4. **if** $(flags[index] = 0)$ **then**
5. $flags[index] := 1$
6. Bloom_filter_insert($pkt.invariants$);
7. $pkt.hopcount := 0$;
8. $pkt.label := f_2(R.IP)$;
9. **else**
10. Increase $pkt.hopcount$;
11. **else**
12. Increase $pkt.hopcount$;

Fig. 3. Rate-limited Probabilistic Logging and Marking (RPLM) scheme

Packet marking: We use the 16-bit marking space which consists of two fields. We use the first 5 bits as the *hopcount* in a same way as in the existing PPM schemes [3] The remaining 11 bits is used to mark the *label*, defined by the output of hash function with the router's IP address. When a packet is sampled, the router writes its 11-bits label to the marking field of the packet (line 5 in Figure 2), and sets the hopcount field to 0. For the packets not sampled, routers saturately increases the hopcount field.

B. Traceback processing

When the victim detects a DDoS attack, it will trigger a traceback procedure. The victim will first collect a decent number of attack packets. Then it will use these packets to track down the attackers. We denote the set of packets that is used for traceback as L_v.

The traceback procedure starts with the victim checking all its immediate neighbors. To check any router S which is one hop away from the victim, the victim first sift out the subset of packets L_S from L_v using two conditions : $pkt.hopcount = 1$ AND $pkt.label = f(S.IP)$, i.e, packets should have the hopcount from the victim to S and should have the signature of S. Then, the victim will query the corresponding (right date and time) Bloom filter at S with the set L_S. The router S is added to the reconstructing attack tree if at least one match is found. Each neighbor R of S will then be queried by new set of packets L_R, sifted from L_v with new conditions : $pkt.hopcount = 2$ AND $pkt.label = f(R.IP)$. Again, if at least one match is found, R will be added to the attack tree. Otherwise, no more neighbor routers of R will be queried. This process is repeated recursively until it cannot proceed.

C. Rate-limited Probabilistic Logging and Marking (RPLM) scheme

Our PLM scheme can get a high accuracy of traceback result with very low overhead at routers at the victim, but the performance improvement is limited by the geometric problem of PPM approach mentioned in Section II-B. Our Rate-limited Probabilistic Logging and Marking (RPLM) scheme applying new heuristic technique to further improve the performance is presented in Figure 3. To limit the rate of packet logging and marking in a time window to a certain destination, we use rate-limiter consisting of a bit array, which is a very simple data structure. Two important parameters of rate-limiter are the size of bit array l and the length of epoch e for operation. In every e seconds, all flags in the rate-limiter are reset to zero. When a packet is arrived and sampled, the router select a random bit, indexed by the hash value of the destination IP in the packet (line 3 in Figure 3). If the value of selected bit is zero, the router will run the logging and marking algorithm as in the basic scheme. If it is not zero, it may mean that another packet to the same destination is already logged and marked in the same time epoch. In this case, logging and marking procedure will be skipped for this packet. Due to the possible hash collision to select a random bit for different packets with their destination IPs, logging and marking procedure can be mistakenly skipped for some packets. To decrease the frequency of this mistakes while keeping the level of performance improvement high, the value of e and l should be tuned appropriately. We present how to tune these two parameters given a target level of storage overhead in Section IV-A. We also show that the performance of our RPLM scheme can be much better than the performance of PLM scheme in Section V.

IV. TUNING PARAMETERS OF RATE-LIMITER AND OVERHEAD ANALYSIS

In this section, we first present the way to tune the parameters of rate-limiter given a target level of storage and computational overhead at the routers for logging and marking operation. Then we analyze the average number of attacking packets required for the traceback procedure.

A. Tuning parameters of rate-limiter given a target level of storage overhead

In our PLM scheme, the maximum sampling and logging rate P_m in a router will be equal or larger than the value of marking probability p because all randomly selected packets will be logged and marked. However, in our RPLM scheme, P_m is controlled by two parameters of rate-limiter: the size of limiter l and the length of epoch e. This is because the maximum possible number of logging and marking operation during an epoch is limited to the bit size of rate-limiter. This in turn implies that if we have a target level of storage overhead, the value of l and e can be tuned appropriately, as following.

Let P_m be the target level of storage overhead. The smaller the value of P_m the better, as long as it is enough to get a high level of traceback accuracy. And let L be the maximum number of packets per second at a link with conservative assumption of 1000 bits average packet size. Then the value of P_m is equal to $l/(L \cdot e)$. If we target the value of P_m as

[3]This 5 bit requirement can be decreased to 1 bit utilizing Time-To-Live(TTL) field modification scheme in [13]. In this case, the performance of our scheme can be improved by increasing the size of remaining marking space.

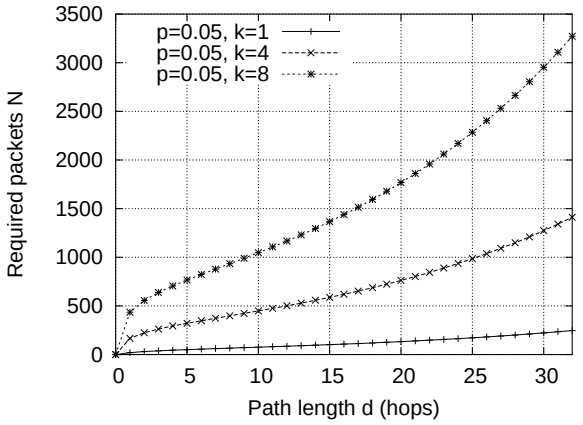

Fig. 4. Expected number of packets N need for reconstructing a path of length d

low as 1% [4] and use 10 milliseconds as the value of e, we can set the value of l as $0.01 \cdot L \cdot e$. For example, in a OC-192 link (10 Gbps), we can use 1000 bit rate-limiter. Simulation results in Section V show that the small 1000-bit rate-limiter can improve the performance drastically.

B. Number of attacking packets required for the traceback procedure

If the victim collects enough attacking packets, which contains at least one marked packets from each routers on the attacking paths in high probability, the traceback processing can be initiated. Therefore, the number of attacking packets required for the traceback procedure can be estimated by using the coupon collector's problem as following. Here we provide the theorem from our previous work [14] with the modification of the parameters for the scheme in this paper. Refer [14] for the proof.

Theorem 1: Let p be the probability of packet marking by an Internet router. Let d be the hop distance from the victim to the related router and k be the number of information fragments from the router in marking procedure. Then, the average number of packets N that need to be received for the victim to reconstruct the path of length l (in number of hops) is

$$N = \frac{1}{\sum_{i=1}^{d} p(1-p)^{i-1}} \int_0^{\infty} \left[1 - \prod_{j=1}^{d} \left(1 - e^{-\lambda_j t}\right) \right] dt$$

$$where \quad \lambda_j = \frac{p(1-p)^{d-1}}{\sum_{l=1}^{d} p(1-p)^{l-1}}$$

Figure 4 shows the comparison of the value of N as d increases. Our scheme uses 1 as the value of k while existing PPM-based schemes typically uses 4 or 8. Three curves in the graph show the increment of N when k=1,4,8 respectively. The graph clearly shows that our scheme require much smaller number of N compared to the other PPM-based schemes. This advantage comes from the simplicity of our marking scheme, in the additional cost of small logging overhead.

[4]Note that the maximum sampling rate P_m is different from the storage requirement relative to the link capacity. The latter is decided by the bloom filter parameters. It will be 0.005 % of link capacity if we use same parameters for the logging as in [10].

V. Performance Evaluation

In this section, we evaluate the performance of the proposed scheme using real-world Internet traffic traces. We show that our scheme can achieve high accuracy of traceback using a small number of attack packets even when there are a large number of attackers.

A. Simulation setup

Simulating network-wide IP traceback scheme requires highly scalable network simulation tool because thousands or millions of nodes can be involved. However, the current popular network simulation tools such as ns2 [15] or OPNET Modeler [16] are not scalable to very large scale network simulation. Therefore, we have developed our own event-driven simulation tools (available at [17]) based on high level of abstraction in network events, which is not directly related to DDoS attack or IP traceback, for scalability.

In a simulation, we first load the topology information from the real-world Internet topologies. Then we randomly choose the attackers and legitimate users to transmit attacking and normal traffic to the victim respectively.

1) topologies: The following three real-world network topologies are used in our simulation study.

- Skitter data I – collected from a CAIDA-owned host (a-root.skitter.caida.org) on 11/28/2001 as a part of the Skitter project [18]. This data contains the traceroute data from this server to 192,900 destinations.
- Skitter data II – collected from another CAIDA host (e-root.skitter.caida.org) on 11/27/2001, containing routes to 158,181 destinations.
- Bell-lab's dataset – collected from a Bell-labs host [19], containing routes to 86,813 destinations. We merged six route sets originated from the same host into one and trimmed incomplete paths.

All three topologies are routes from a single origin to many destinations in the Internet. In our simulation, we assume that this origin is the victim, and the attackers and the legitimate clients are randomly distributed among the destination hosts.

2) Traffic modeling: In our simulation, we model the network traffic during a given time epoch e as following. Each legitimate clients randomly generate the packets to the victim. The average inter-arrival time of the packets from a client follows exponential distribution with rate T_l. Each attackers generate the random attacking traffic to the victim in a same way, but with much higher average inter-arrival time T_a.

Each router on the attack path compiles the packets from legitimate clients and the attackers during the same epoch, and then randomly save the digests and mark using the proposed scheme. The background traffic is also considered in the simulation because it can affect the performance of the proposed scheme due to the hash collision in the rate-limiter. To simulate the background traffic at a router during a

given time epoch, we randomly choose a part of the following real packet header traces and apply the proposed algorithm to simulate the effect of the hash collision.

- Trace from University of North Carolina (UNC) (courtesy of Prof. Jeffay) – collected at the 1 Gbps access link connecting the campus to the rest of the Internet, on April 24, 2003.

- Trace from University of Southern California (USC) (courtesy of Prof. Papadopoulos) – collected at their Los Nettos tracing facility on Feb 2004.

3) Performance metrics: To evaluate the accuracy of the attack tree constructed by the traceback scheme, we use two metrics: *false negatives ratio* (FNR) and *false positive ratio* (FPR). We call the attack paths that is not detected by the traceback scheme as *false negatives*. FNR of an attack tree constructed by the traceback scheme is defined as the ratio of the number of false negatives to the number of actual attack paths. Traceback scheme may identify attack paths that are not actually on attack graph, because of the nature of Bloom filters. We call these paths *false positives*. The FPR of an attack tree constructed by the traceback scheme is defined as the ratio of the number of false positives to the total number of attack paths in the attack tree.

B. Performance of our scheme

Figures 5(a,b,c) show the FNR of the proposed scheme against the average number of attack packets from an attacker, under the three aforementioned Internet topologies. Similarly, Figures 6(a,b,c) show the FPR values. In all figures, we use 1000 bits rate-limiter and 10 millisecond epoch length. The three curves in each figure correspond to 1,000, 2,000 and 5,000 attackers, respectively. All curves in Figures 5(a,b,c) show that as the number of attack packets used for traceback increases, FNR value decreases sharply. For example, more that 98% of the attacking path can be identified with only 150 packets from each attacker in average, in all three curves. This result shows that the performance of the proposed scheme is much better than previous PPM-based schemes which require more than thousands of the packets from each attacker. Another salient point is that our scheme performs better if the number of attacker increases. This shows the effectiveness of using rate-limiter to improve the performance of traceback scheme by reducing duplicated marking information from the routers near the victim, and hence increasing the chance to reach to the victim of the marking information from further routers. As shown in Figures 6(a,b,c), the FPR value increases very slowly as the average number of packets increases, and is always reasonable.

The two curves in each Figures 7(a,b,c) compares performance when we use (RPLM scheme, upper curve) and not use (PLM scheme, lower curve) the rate-limiter under the attack from 1000 attackers. Clearly, rate-limiter can decrease the curve more sharply, and the victim can get much better traceback results with a given number of average packets from an attacker.

Figures 8(a,b,c) shows the performance result when we change the marking probability p from 0.1 to 0.4. All graph with three different topology show that we can get the best result when we use 0.2 as the marking probability. If the marking probability is lower than 0.2, the amount of initial marking each router becomes too small to be delivered to the victim without mark overwriting. If the marking probability become too large, the mark in packets from a router will be overwritten by the following routers on the path to the victim with high probability.

VI. Related Work

Two main classes of solutions has been proposed to address IP traceback problem. One class is PPM approach [7], [8], [9] and the other is packet logging approach [10]. However, both approaches have a scalability problems under heavy DDoS attack due to their own algorithm overhead.

Many techniques to reduce the overheads of the two approaches are proposed. Yaar *et al.* [13] propose to utilize Time-To-Live(TTL) field in IP header to reduce the complexity of the packet marking scheme in [8]. Using their scheme, the size of counter field in any packet marking scheme which uses the counter field can be reduced from 5 bits to only 1 bit. This scheme can be also combined with our proposed scheme in this paper to improve the performance. Technique to improve packet logging approach is proposed in our previous work [11]. Our previous approach is to sample and log a small percentage of packets using an smart and efficient sampling scheme utilizing only 1 bit flag in IP header to drastically reduce the space and computational overheads in routers. Basheer and Manimaran [20] proposes a hybrid scheme to utilize the advantages of both packet marking and logging, similarly to ours. In their work, routers on the attacking path create distributed linked list using their packet marking and logging scheme. Then the victim can traceback the origins of the packets by tracing the linked list by sending queries to the upstream routers.

VII. Conclusion

In this paper, we have presented a novel IP traceback approach utilizing both packet logging and PPM approach. In this work, the overheads at the routers and the victim can be drastically decreased relative to existing PPM approaches or packet logging approaches. We have also introduced a heuristic technique to further improve the performance of the packet marking scheme. Our simulation results show that the proposed scheme performs and scales very well to get the high accuracy of IP traceback result with a very small number of attack packets at the victim and very small packet logging overhead at routers.

References

[1] L. Garber, "Denial-of-service attacks rip the Internet," *IEEE Computer*, vol. 33, no. 4, pp. 12–17, Apr. 2000.
[2] M. Delio, "Irc attack linked to dos threat," http://www.wired.com/news/culture/0,1284,41167,00.html, Jan. 2001.
[3] D. McGuire and B. Krebs, "Attack on internet called largest ever," http://www.washingtonpost.com/wp-dyn/articles/A828-2002Oct22.html, Oct. 2002.
[4] R. Lemos and J. Hu, "'zombie' pcs caused web outage, akamai says," http://news.zdnet.com/2100-1009_22-5236403.html, June 2004.

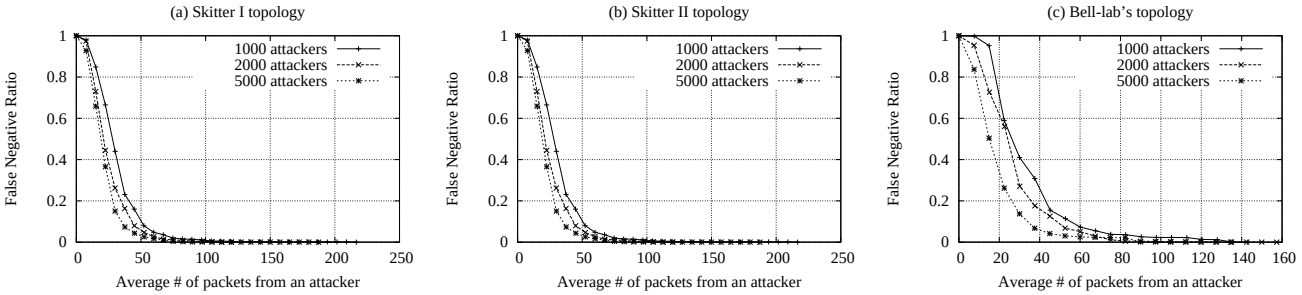

Fig. 5. False Negative Ratio of our traceback scheme on three different topologies

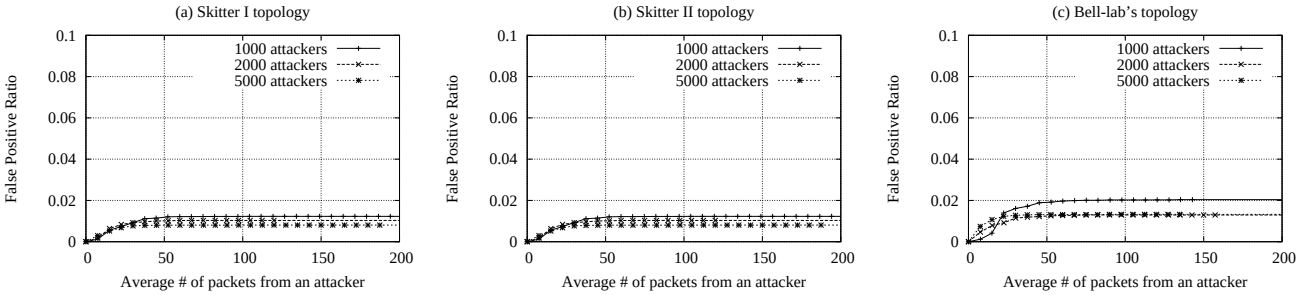

Fig. 6. False Positive Ratio of our traceback scheme on three different topologies

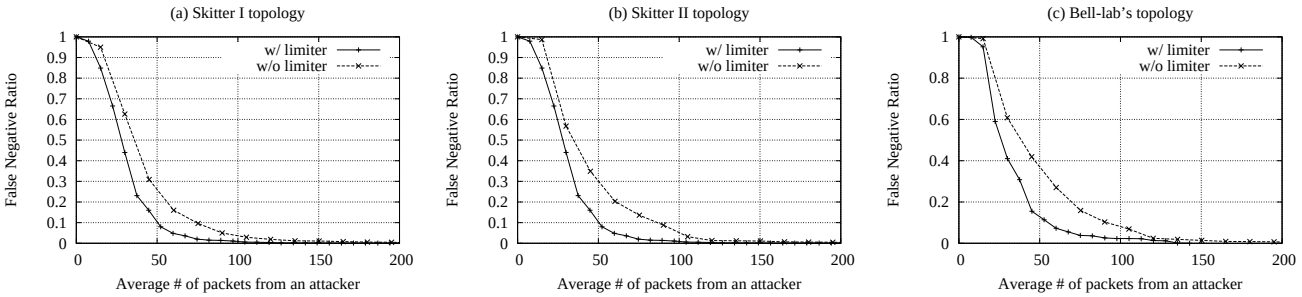

Fig. 7. False Negative Ratio of our traceback scheme with and without rate-limiter on three different topologies

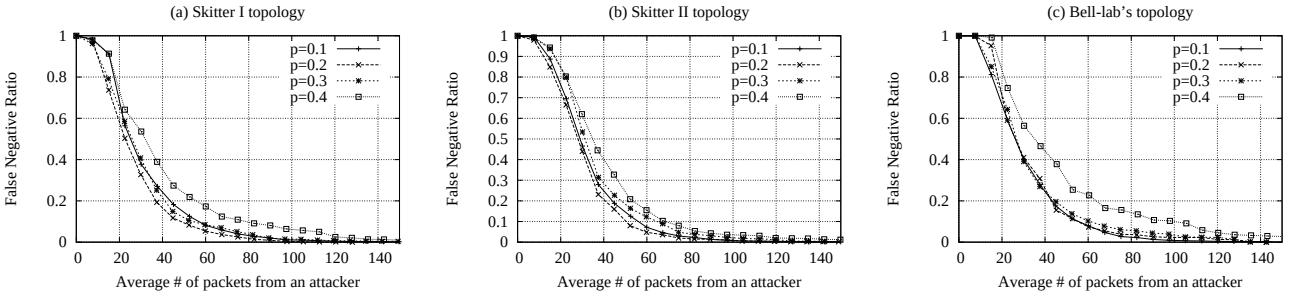

Fig. 8. False Negative Ratio by varying marking probability p on three different topologies

[5] Computer Security Institute, "2005 CSI/FBI computer crime and security surver," http://www.gocsi.com.

[6] D. McPherson, C. Labovitz, and F. Jahanian, "Backbone attack detection and mitigation methodologies," in *Tutorial at ACM SIGCOMM*, 2005.

[7] S. Savage, D. Wetherall, A. Karlin, and T. Anderson, "Practical network support for IP traceback," in *Proc. ACM SIGCOMM*, Aug. 2000, pp. 295–306.

[8] D. Song and A. Perrig, "Advanced and authenticated marking schemes for IP traceback," in *Proc. IEEE INFOCOM*, Apr. 2001, pp. 878–886.

[9] M. T. Goodrich, "Effi cient packet marking for large-scale IP traceback," in *Proc. ACM CCS*, November 2002, pp. 117–126.

[10] A. Snoeren, C. Partridge, *et al.*, "Hash-based IP traceback," in *Proc. ACM SIGCOMM*, Aug. 2001, pp. 3–14.

[11] J. Li, M. Sung, J. Xu, and L. Li, "Large-Scale IP Traceback in High-Speed Internet: Practical Techniques and Theoretical Foundation," in *Proc. IEEE Symposium on Security and Privacy*, May 2004.

[12] B. Bloom, "Space/time trade-offs in hash coding with allowable errors," *Communications of the Association for Computing Machinery*, vol. 13, no. 7, pp. 422–426, 1970.
[13] A. Yaar, A. Perrig, and D. Song, "FIT: Fast Internet Traceback," in *Proc. IEEE INFOCOM*, Mar. 2005.
[14] M. Sung and J. Xu, "IP Traceback-based Intelligent Packet Filtering: A Novel Technique for Defending Against Internet DDoS Attacks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 14, no. 9, pp. 861–872, Sept. 2003, preliminary version appeared in Proc. 10th IEEE ICNP.
[15] "The network simulator - ns-2," http://www.isi.edu/nsnam/ns/.
[16] "Opnet modeler," http://www.opnet.com/products/modeler/.
[17] M. Sung, "DDoS attack simulator," http://www.cc.gatech.edu/~ mhsung/ddos_simulator/, June 2006.
[18] "CAIDA's Skitter project web page," Available at http://www.caida.org/tools/measurement/skitter/.
[19] B. Cheswick, "Internet mapping," Available at http://cm.bell-labs.com/who/ches/map/dbs/index.html, 1999.
[20] D. Basheer and G. Manimaran, "Novel hybrid schemes employing packet marking and logging for ip traceback," *IEEE Transactions on Parallel and Distributed Systems*, vol. 17, no. 5, pp. 403–418, May 2006.