A Novel Packet Marking Scheme for IP Traceback

Basheer Al-Duwairi and G. Manimaran
Dependable Computing & Networking Laboratory
Dept. of Electrical and Computer Engineering
Iowa State University, Ames, IA 50011
{dbasheer,gmani}@iastate.edu

Abstract

Recently, several schemes have been proposed for IP traffic source identification for tracing denial of service (DoS) attacks. Most of these schemes require very large number of packets to conduct the traceback process, which results in lengthy and complicated procedure. In this paper, we address this issue by proposing a new scheme, called Probabilistic Pipelined Packet Marking (PPPM), which employs the concept of "pipeline" for propagating marking information from one marking router to another so that it eventually reaches the destination. The key benefit of this pipeline process lies in drastically reducing the number of packets that is required for the traceback process. We evaluate the effectiveness of the proposed scheme for various performance metrics through combination of analytical and simulation studies. Our studies show that the proposed scheme offers high attack source detection percentage, and attack source localization distance of less than two hops under different attack scenarios.

1 Introduction

Current Internet infrastructure is exposed to many serious threats that can affect the availability of important Internet services. Denial of Service (DoS) attacks and a more complicated version known as Distributed DoS (DDoS) are the most common. These attacks deny regular Internet services from being accessed by legitimate users either by blocking service completely or by disturbing it such that users become not interested in the service anymore (for example causing significant delay in accessing an airline reservation web site). Today's Internet has witnessed several incidents that confirm the devastating effect of such attacks. For example, in October 2002, eight out of the thirteen root DNS servers were brought down as a result of severe flooding denial of service attack [7].

Recently, the research community has paid a lot of attention to overcome DoS attacks. These efforts concentrated on attack prevention [4, 9, 6] and attack traceback [1, 2, 3, 10]. Prevention schemes aim to drop attack packets as soon as they are discovered based on the fact that attack packets hold forged IP source addresses. For example, in ingress filtering [4] routers are configured to block packets that arrive with illegitimate source addresses. Another example is the SAVE protocol [6], which is designed to provide routers with the information needed for source address validation.

Attack traceback, the subject of this paper, has emerged as a promising solution to DoS attacks. Tracing attacks back to their sources is a challenging problem for the following reasons. In DDoS attack, the attacker's machine (the master) instructs previously compromised innocent machines (the slaves) to aggressively overwhelm the victim by high volume streams of flooding packets with faked IP source addresses leaving the victim with no clue about the true sources of these packets. In large scale DDoS, it is not necessary for each attack source to send significant number of packets. Instead, the attack is caused by the aggregate traffic converging at the victim site. This distributed anonymous nature of the attack helps the attacker to stay behind the scenes. The stateless nature of the Internet combined with the destination oriented IP routing which does not require any source address validation increases the difficulty of tracing attacks back to their sources. This problem is also complicated by the fact of having millions of hosts connected to the Internet, which implies a huge search space.

In this paper, we propose a novel and efficient packet marking scheme for IP traceback called *Probabilistic Pipelined Packet Marking* (PPPM). The proposed scheme aims to propagate the IP addresses of the routers that were involved in marking certain packet by loading them into packets going to the same destination. Therefore, preserving these addresses while avoiding the need for long term storage at intermediate routers. In the next section, we provide an overview about the related work. In section 3 motivate this work and highlight the main contributions. In section 4, the proposed solution is presented. Section 5 provides some theoretical analysis. In section 6, we describe the simulation studies. Finally, conclusions are drawn in section 7.

2 Related Work

Traceback schemes usually rely on router assistance to determine the path followed by attack packets and eventually identify the attack source. Our work is inspired by the probabilistic packet marking (PPM) approach [10] which we describe here briefly. The main idea in PPM is to let routers mark forwarded packets (probabilistically) hoping that after receiving large number of packets, the path followed by them can be reconstructed based on the marking information obtained in each received packet. PPM employs *edge sampling* technique, in which two static address fields, *start* and *end*, are reserved in each packet to represent the end points (i.e., routers) of a link along the packet path. An additional *distance* field is also reserved to represent the distance of an edge sample from the victim. When a router decides to mark a packet, it writes its own address into the start field and writes a zero into the distance field. Otherwise, if the distance field is already zero then it writes its own address into the end field and increments the distance field. Therefore, forming an edge between itself and the previous router. If the router does not mark the packet, then it always increments the distance field. Major problems of PPM are discussed in section 3.

In [5], the authors proposed a scheme called *Path Information Caching and Aggregation* (PICA). The scheme has the advantage of triggering traceback messages only when the aggregate count of packets going to the same subnet exceeds certain threshold during a specified interval of time. The main problem with PICA is that in large scale DDoS attacks, the specified threshold may never be exceeded, especially at routers far away from the victim. In hash-based traceback [11], bloom filters were used to save packet digests at intermediate routers. The major concern in this scheme is the limited storage available at intermediate routers, which in turn leaves a small window of time through which the saved digests can be collected and successfully traced. DLLT [3] is an efficient scheme for IP traceback that combines the good features of PPM and hash based schemes. The main objective of DLLT is to preserve the marking information done to certain packet by storing them at intermediate routers while establishing a link list between them in a way that facilitates their collection upon attack detection. Long term storage of packet digests represents the major problem of DLLT.

3 Motivation and Contributions

The imminent threats imposed by DoS attacks call for efficient and fast traceback schemes. Here, we list some of the desirable features of a good attack traceback scheme.

1. Providing accurate information about routers near the attack source rather than those near the victim.
2. Recognition and exclusion of false information injected by the attacker.
3. Avoiding the use of large amount of attack packets to construct the attack path or attack tree.
4. Low processing and storage overhead at intermediate routers.
5. If packet information is to be maintained at intermediate routers then collecting this information must be efficient.

In this paper we propose a traceback scheme that enjoys the above features. Also, we try to eliminate the major problems of PPM [10] which we discuss here. PPM lacks many of the desirable features mentioned in the beginning of this section. For example, routers that are far away from the victim have very low chance to pass their marking information to the victim because down stream routers overwrite this information, which leads to the loss of valuable marking information written by routers far away from the victim. This fact is contradictory to our desire to have more knowledge about routers far away from the victim (i.e., the nearest routers to the attackers). PPM requires considerable amount of packets to be collected at the victim before conducting the traceback process. Waiting for a huge number of attack packets to be collected at the victim means admitting the fact of longer denial of service, which conflicts with the need for fast response to stop the attack. This argument may not be valid when a small value (e.g., 0.05) for the marking probability is used in PPM. However, such choice of the marking probability opens a serious vulnerability in PPM that was pointed out in [8], where the attacker has the ability to pass spoofed marking information to fool the victim. *Therefore, PPM is not able to recognize and exclude false marking information inscribed by the attackers if a small value for the marking probability is used. Also, it requires processing of huge amount of packets if large marking probability (e.g., 0.2 or 0.3) is used.*

Motivated by the need to have knowledge about all routers involved in marking certain packet without incurring significant processing and storage overhead, we propose a novel packet marking scheme called *Probabilistic Pipelined Packet Marking (PPPM)*, which enables the destination to identify all marking routers of a given packet. Also, we evaluate the proposed scheme through a combination of analytical and simulation studies.

4 Proposed Solution: Probabilistic Pipelined Packet Marking (PPPM)

4.1 Pipelined Marking Concept

In computer architecture, pipelining is used to allow more than one instruction to be in some stage of execution at the same time. Usually, each instruction passes through sequence of stages during its life cycle. The outcome of one stage is forwarded to the next stage before being replaced by another instruction from the previous stage. The whole process is repeated at each stage until the whole instruction is executed. We propose a similar approach for packet marking, where marking refers to augmenting IP packets with IP address of the marking router. In computer architecture analogy, a router that marks a packet represents a pipeline stage, the marking process represents the instruction execution, and the propagation of marking information from one marking router to another represents the flow of instructions in a pipelined system. *The main idea is to transfer the marking information that belongs to certain packet by propagating them from one marking router to another using subsequent packets going to the same destination.*

We illustrate this idea by the example shown in Fig 1. In this example packets P_1, P_2,.. are forwarded from S to D through the simple path consisting of routers *A*, *B*, *C*. We assume that a marking information field , MI_{Ri}, is allocated in each packet to hold the marking information written by router R for packet i. It is important to notice that the content of this field keeps changing from one marking router to another. Therefore, the marking information written by the last router (i.e., router C) will be the first to reach the destination D. Our objective is to transfer the marking information done by other routers (i.e., A and B) to the destination. Our pipelined scheme is based on buffering the marking information found in certain packet before remarking it, and then sending the buffered information by loading it in subsequent packets going to the same destination.

To simplify the discussion, we assume that only packet P_1 is being marked by all routers (A, B, C), while other packets are not marked (The actual algorithm is slightly different). The figure shows the marking information buffered at each router as packets traverse the path. It also shows the marking information being written in each packet along the path. Packets P_2 and P_3 are used to transfer the marking information that belongs to P_1. It is clear that after receiving three packets, the destination is able to collect the marking done for packet P_1 (i.e., MI_{A1}, MI_{B2}, and MI_{C1}). It is important to mention that the marking information includes an ID that uniquely identifies the packet such that the received marking information can be identified as belonging to one particular packet.

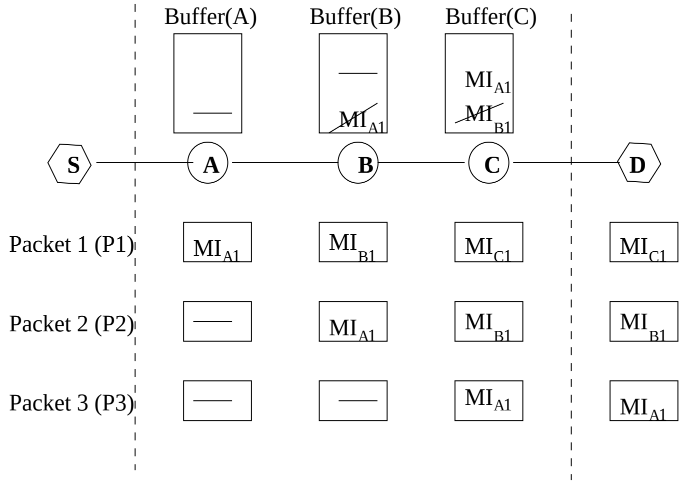

Figure 1. An example of pipelined-based marking

4.2 Details of Probabilistic Pipelined Packet Marking (PPPM)

In this subsection, we provide details about PPPM. We start by describing the fields need to be allocated in each packet for marking purposes, the information need to be buffered at each marking router, then we describe the pipelined marking and buffering algorithm, followed by attack source identification algorithm.

Marking Information Carried in Each Packet: The following fields are allocated in each packet to carry the marking information from one marking router to another:

- 32-bit IP address: represents the IP address of the marking router.
- 8-bit $T\bar{T}L$: This field is used to obtain the actual distance between the marking router and the victim. $T\bar{T}L$ is normally set to the value of TTL found in the packet before remarking it except when the packet is used to transfer a buffered marking information that belongs to another packet, say P_x. In which case, the $T\bar{T}L$ is set to $(TTL + T)$, where T represents the distance between the previous marking router of P_x and the current router.
- k-bit ID: this is an ID that uniquely identifies the marked packet. It is chosen randomly by the first marking router (i.e., the first router to mark certain packet).

Marking Information Buffered at Each Marking Router: The marking information found in certain packet are buffered at the receiving router before remarking it. The buffer can be implemented as a destination based table (i.e., buffer entries are maintained per destination). For each destination, the most recent marking information is buffered. This includes the same marking fields found in the marked packet except that instead of storing the $T\bar{T}L$ value, the difference T=$T\bar{T}L$-TTL is stored . This difference represents the distance between the previous marking router and the current marking router. The advantage of this field is to determine the distance of certain marking router from the destination.

Determining The Distance of Each Marking Router from The Destination: Fig. 2 shows the benefit of using the $T\bar{T}L$ field in determining the distance of certain marking router from the destination. Let packet P_1 be sent from S_1 along the shown path to D. Packet P_2 is sent from different source, S_2, to the same destination D. We assume that P_1 was sent before P_2, and we assume that P_1 was marked by routers R and W respectively, and that its ID was set to x by router R; while P_2 was marked by routers F and W and its ID was set to y by router F. The figure shows the values of (packet ID, marking router, TTL, $T\bar{T}L$) for both packets after they leave each router along their paths. Also, it shows the content of the buffer at router W.

In this example, P_2 is used to transfer the marking information that belongs to packet P_1 which was buffered at router W. After receiving the two packets, the destination can identify that packet x was marked by routers R and W which are 3 and 1 hops away respectively. Determining the distance of each marking router from the destination is incorporated in the marking procedure described in Fig. 3, where a router, R, decides whether to mark a packet or not based on some marking probability q.

Realizing Probabilistic Edge-based Marking: It is to be noted that *probabilistic edge marking* (an edge is composed of two adjacent routers on the packet path) can be realized with a slight modification to the proposed marking and buffering procedure. Whenever a router decides to mark a packet, we force the subsequent router to mark the same packet. This can be achieved by maintaining a 1-bit field called *marking flag* as part of the marking information to be held in the packet. This flag is used to enforce deterministic marking when it is *on*. When it is *off*, the marking becomes probabilistic. With this flag, the probabilistic edge marking in PPPM can be implemented as follows: When a router receives a packet, it checks the marking flag. If it is *on*, it has to do the marking and buffering procedure and then reset the flag. Otherwise (i.e., when the flag is *off*), it takes the marking decision based on some probability q. If the decision outcome is to mark the packet it will do that, and then set the flag to 1 such that the next adjacent router will do

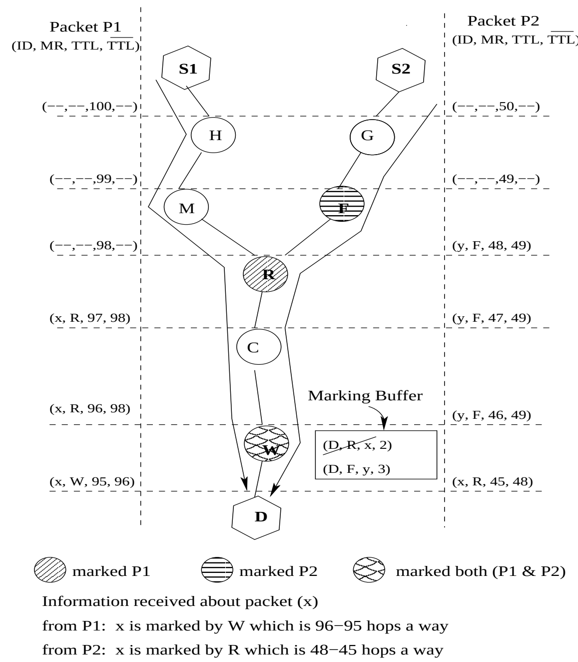

Figure 2. An example of distance calculation

the marking deterministically. *Through out the rest of this paper we adopt this modification since identifying edges on the attack bath would be more useful and leads to fast and efficient source identification.*

Attack Source Identification: Once a DoS attack is detected, the victim starts the source identification process using k attack packets as an input. By realizing that packets holding the same ID carry marking information that belongs to the same packet, the victim can extract the marking information of each of the distinct IDs found in the k packets, by running the simple algorithm shown in Fig 4. The outcome of this algorithm is an ID table with two columns: *Packet ID* and *Marking Routers List* and their *distances* from the victim. The marking routers list obtained for each ID can be converted into marking edges by observing the distance of each marking router. In [10], the authors proposed an edge-based source identification algorithm that we can use directly due to the similarity of input information (i.e., path edges and there distances) in both schemes.

4.3 Security of PPPM

The attacker is expected to behave like a marking router. Therefore, marking its outgoing packets with false information before being marked by any router along their path. For example, it can spoof the IP-marking field. Also, it can write any value in the $T\bar{T}L$ field. If the attacker uses a $T\bar{T}L$ value that is less than the original packet's TTL, then it is very easy to distinguish and drop attack packets

```
Marking and Buffering Procedure at Router R:
for each packet P
1. Choose a random number x in [0,1]
2. if (x>q)
      forward P.
3. else
      a. if (P.destination is NOT in R.MB)
            ● if (P.ID == 0)
                  Mark P (i.e., P.ID = random(), P.T̄TL= P.TTL
                        , P.MR = R.)
            ● else
                  Buffer P. MI (i.e., R.MB.Dest [i]= P.Dest, R.MB.ID= P.ID,
                              R.MB.T=P.T̄TL − P.TTL).
                  Remark P (i.e., P.T̄TL=P.TTL, P.MR=R)
      b. else
            ● save P.MI in temporary variable (i.e., temp. ID = P.ID, temp.MR=P.MR,
                              temp.T̄TL= P. T̄TL)
            ● remark P (i.e., P.ID= R.MB.ID, P.MR= R.MB.MR, P.T̄TL=P.TTL+R.MB.T̄)
            ● Buffer P (i.e., R.MB.ID = temp.ID, R.MB.MR=temp.MR, R.MB.T̄=temp.T̄TL−P.TTL)

Notation:
MB: Marking Buffer
MI: Marking Information
MR: Marking Router
R.MB.T: the value of T at MB of R
R.MB.Dest{i]: the destination stored
              in the ith entry of R.MB
```

Figure 3. Pipelined marking and buffering algorithm

```
for each packet P
1. if (P.ID NOT in the ID Table (IDT))
      add entry for P.ID in IDT
2. append P.MR & (P.TTL − P.TTL) to the ID.marking routers list
```

Figure 4. ID table construction

before reaching the victim, because $T\bar{T}L$ must be larger than the TTL value of a given packet. If the $T\bar{T}L$ written by the attacker is larger than the packet's original TTL, then intermediate routers cannot distinguish attack packets. This false information is more likely to be overwritten by subsequent routers if a reasonable marking probability is used (the choice of marking probability is discussed in section 5). Unfortunately, there is nothing that can prevent the false information from propagating to the victim. For a given packet ID, the victim will include the spoofed marking router in the marking routers list for that ID. However, because of the restriction that we imposed on $T\bar{T}L$ (i.e., $T\bar{T}L$ must be larger than TTL), the distance obtained at the victim for the spoofed marking router will be always larger than that obtained for any other marking router of the same packet. Therefore, the victim can easily distinguish and exclude this information.

The attacker may use the same packet ID for all its outgoing packets. Worst than that, he can coordinate with other attackers to use the same packet ID either continuously or from time to time. In PPPM, the first marking router can not recognize itself as the first router to do the marking for a given packet. Therefore, the ID injected by the attacker is used. We suggest the following slight modification to the marking and buffering algorithm shown in Fig. 3, to prohibit attackers from using the same ID for all packets: if a router receives a packet, P, that has identical (P.dest, P.ID) to those buffered for previously marked packet, it will drop the incoming packet. The suggested modification is based on the fact that it is very unlikely for two distinct packets going to the same destination within small window of time to have identical IDs.

5 Analysis

Choice of The Marking Probability: The value of the marking probability, q, plays a direct role on the effectiveness of PPPM. On one extreme, if we set q to 1 (i.e., deterministic marking) then d attack packets would be sufficient to locate the attacker (where d is the number of of PPPM enabled routers in the pipeline between attacker and victim). This choice is excluded in practice to avoid excessive overhead on network routers. On the other extreme, if we set q very close to 0, then larger amount of packets would be required to reconstruct the attack path. In general,the following issues must be considered when choosing q:

1. Ensuring that at least one router along the attack path would mark a given packet. This has the benefit of overwriting any false marking information injected by the attacker when acting as a marking router.

2. Minimizing the number of packets required for attack traceback. This bound is expressed as a function of q (see equation 5), where it can be seen that the bound is minimized by increasing q.

3. Minimizing the overhead on network routers. The amount of overhead imposed by PPPM on network routers can be lowered by decreasing q.

While issues 1 and 2 are consistent, they conflict with issue 3. Therefore, there is a trade off between the efficiency of the proposed scheme and the amount of overhead imposed on network routers. In [8], it was shown that the attacker's ability to hide his location (by injecting false marking information into its attack packets) is curtailed by increasing the marking probability, however in PPM [10] the degree to which the victim can delimit the attacker's injection of false marking information is bounded by sampling constrains. In PPPM, there is no maximum bound on the value of q because information about all marking routers of a given packet will eventually propagate to the victim. In our simulation studies (section 6), we use relatively large values for q to illustrate the efficiency of PPPM. In practice, lower values for q would be used to minimize overhead on network routers. The choice of q is a design issue that involves a trade off between efficiency and overhead.

Number of Attack Packets Required to Identify the Attacker: In this section, we show that the number of attack packets required to identify the attack sources is substantially less than that required by PPM [10]. Our objective is to find a bound on the minimum number of packets that has to be received by the victim such that every router on the path from attacker to victim is involved in marking at least one of these packets with high confidence probability u. Let y represent this lower bound. Let the marking probability at router R be q. Let P_f be the probability that R fails to mark any packet out of the y packets. Clearly, $P_f = (1-q)^y$. Therefore, the probability that R will succeed in marking (we call it the success probability) at least one packet is given by:

$$P_s = 1 - P_f = 1 - (1-q)^y \qquad (1)$$

In multiple attackers case, the success probability is not uniform for all routers. In fact, it increases for routers near the victim because attack packets start to converge at these routers. To simplify the analysis, we assume that attackers are attached to leafs of a d-ary tree rooted at the victim (i.e., each router has d incoming interfaces and one outgoing interface). Also, we assume that each attacker is n hops away from the victim, and each of them injects the same number, y, of packets simultaneously. Our objective is to find the minimum value of y such that all routers along the attack path succeed in the marking procedure. Since we assume that each router has d incoming interfaces and one outgoing interface, the number of attack packets forwarded by a router l hops away from the victim (i.e., $n-l$ hops away from the attacker) can be expressed as:

$$y_l = d^{n-l}.y \qquad (2)$$

Therefore, the success probability at such router becomes:

$$P_{sl} = 1 - P_{fl} = 1 - (1-q)^{y_l} \qquad (3)$$

This means that routers close to the victim has better chance to mark their packets (i.e., higher success probability). However, to obtain the desired bound we can safely assume that the success probability for all the n routers along the attack path is the same and equal to that of the farthest router (i.e., as given in equation 1)

If we define X to be a random variable that represents the number of routers out of n that were successful in the marking process, then X follows the binomial distribution with success probability P_s given in equation 1. We need to find y such that:
$P(X = n) \geq u$ (i.e., the probability that each router succeeds in marking at least one packet is larger than u). But,

$$P(X = n) = (1 - (1-q)^y)^n \geq u \qquad (4)$$

solving for y, we obtain:

$$y \geq \frac{\lg(1 - u^{1/n})}{\lg(1-q)} \qquad (5)$$

Table 1. Simulation parameters

parameter	value
number of attackers (m)	100
number of attack packets used (n)	1500
marking probability (q)	0.15
attack path length (l)	15

For example, for an attack path of length 15 and marking probability of 0.3 and confidence probability 0.95 the number of packets required by PPPM is 16 compared to 1340 in PPM [10]. This significant reduction is due to the mechanism employed by PPPM to propagate marking information of routers far away from the victim.

6 Simulation Studies

We have carried out extensive simulation experiments to evaluate the proposed PPPM scheme. The performance of the proposed scheme was evaluated based on two metrics. We start by defining them. Then, we specify the simulation parameters and present the results.
Performance Metrics: We use the following metrics to evaluate the proposed scheme:

- *Attack source localization distance (ALD)*: This metric defines the distance between the detected attack source and the actual attack source. A value of *zero* for ALD means that the exact attack location is detected.

- *Detection percentage*: This metric defines the percentage of exactly detected attack sources (i.e., detected attack sources with ALD = 0). For example, if j attack sources out of total m attack sources are exactly located, then we express the detection percentage as $\frac{j \times 100}{m}$ %.

Simulation Method: Table 1 shows the default values of the simulation parameters used in our experiments. In each simulation experiment we generated a random attack tree with m attackers and one victim. The attack path length l was the same for all attackers. Packets were marked according to the specified probability q. Attackers were instructed to inject their packets simultaneously with a rate of 1000 packets/attacker. Each of the following results represents the average of 200 independent simulation runs based on the simulation parameters values shown in Table 1 unless otherwise specified. The simulation experiments were designed to study the effect of the above parameters on ALD and attack source detection percentage.

Simulation Results: Attack localization distance is a metric associated with each attack path. Therefore, we performed the simulation experiments in this part based on single attack path (i.e., $m = 1$). Fig. 5 shows the ALD as a function of the number of packets used for traceback process, n, for different values of q. As expected, increasing the number of packets used for traceback results in better performance (i.e., lower ALD), because more information is obtained about the attack path. It is clear that PPPM is very effective even for small values of q. For example, the attack source can be located exactly after receiving 21 attack packets when the marking probability is 0.2.

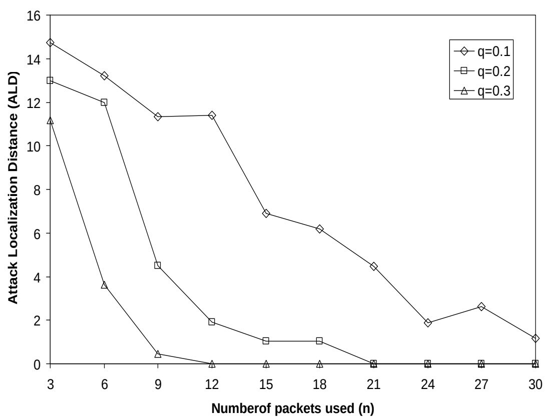

Figure 5. Effect of number of attack packets used (n) on the attack localization distance (ALD) for different marking probabilities (q)

To study the effect of the attack path length l on the attack localization capability of PPPM, we fixed n to 15 while changing l from 10 to 20 in steps of two for different values of q. Fig. 6 shows the results of this study. As l increases, ALD also increases for all values of q. This can be explained by recalling that we are using the same number of packets (i.e., 15) for different values of l. To keep ALD low, larger number of packets must be used as l increases.

PPPM offers very high attack source detection percentage that exceeds 90% when enough packets used in the traceback process. Fig. 7 shows the detection percentage of PPPM as a function of n for different values of q. In this experiment, 100 attackers participated in the attack. It is obvious that the detection percentage is directly affected by the marking probability and the number of packets used in the traceback process. Increasing the values of these two parameters increases the detection percentage. In practice,

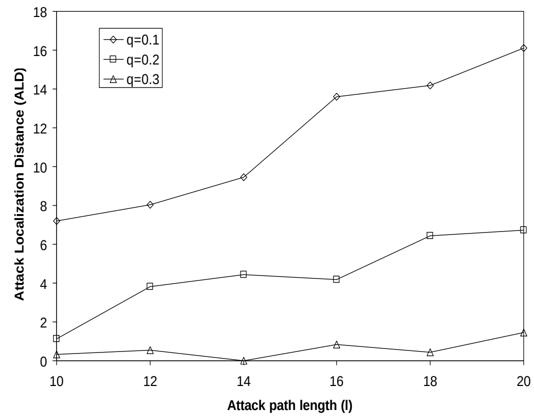

Figure 6. Effect of attack path length on the attack localization distance (ALD) for different marking probabilities (q)

the victim has no control over the value of q. Therefore, n should be increased in order to achieve better performance.

The attack path length, l, seems to have negligible effect on the attack detection percentage. Fig. 8 supports this observation. It shows the results of an experiment conducted using 100 attackers and 1500 packets (i.e., $m = 100, n = 150$). The results are different than what we expected because packets coming from multiple attackers converge at different routers and continue their journey to the victim using the same path. Therefore, regardless of the attack path length, usually enough packets will pass through the path contributing to better detection capability.

7 Conclusion and Future Work

In this paper, we proposed an efficient traffic source identification scheme called *Probabilistic Pipelined Packet Marking* (PPPM). The proposed scheme aims to propagate the IP addresses of the routers that were involved in marking certain packet by loading them into packets going to the same destination. Therefore, preserving these addresses while avoiding the need for long term storage at intermediate routers. Such scheme is very useful in identifying Denial of Service (DoS) attack sources. The main advantage of employing this technique in PPPM, is to significantly reduce the number of packets needed by the victim in the traceback process, which leads to faster and scalable identification of attack sources. Therefore, stopping the attack at early stages, and detecting attackers that have low indi-

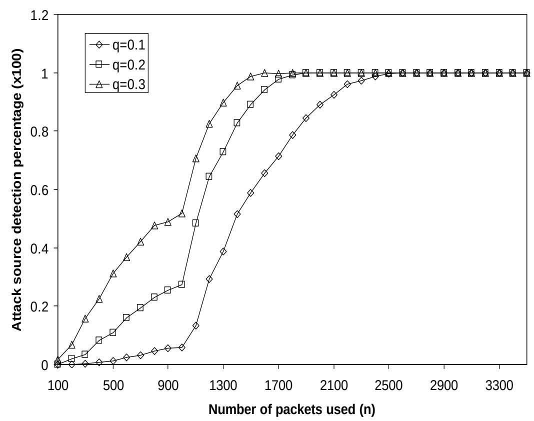

Figure 7. Attack source detection percentage as a function of the number of packets used n **for different marking probabilities** q

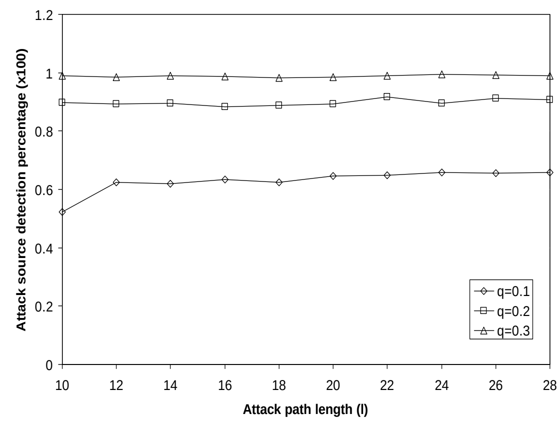

Figure 8. Attack source detection percentage as a function of the attack path length (l**) for different marking probabilities (**q**)**

vidual contribution of attack packets. In addition, PPPM enjoys many features that were listed in section 3 including its immunity to many kinds of attacks.

We evaluated the proposed scheme through a combination of theoretical analysis and simulation studies. In our theoretical analysis, we showed that the number of attack packets needed by the victim to conduct the traceback process is substantially less than that required in PPM [10]. Several issues in PPPM still need to be investigated in more depth. For example, current architecture of PPPM assumes the allocation of 57-bits in the IP header for marking purposes, which is unrealistic assumption. Currently, we are developing efficient encoding schemes to overcome this problem. Also, more research needs to be done to address the issue of IP packets fragmentation and reassembly. Our future work will also focus on the issue of state maintenance at intermediate routers to support this scheme.

References

[1] H. Burch and B. Cheswick, “Tracing anonymous packets to their approximate source,” in *Proc. 2000 USENIX LISA Conf.*, Dec. 2000, pp.319-327.

[2] D. Dean, M. Franklin, and A. Stubblefield, “An algebraic approach to IP traceback,” in *Network and Distributed System Security Symposium (NDSS ’01)*, Feb. 2001.

[3] B. Duwairi, A. Chakrabarti, and G. Manimaran, “An Efficient Packet Marking Scheme for IP Traceback,”. *in Proc. of Networking 2004*, Athens, Greece.

[4] P. Ferguson and D. Senie, “Network ingress filtering: Defeating denial-of-service attacks which employ IP source address spoofing,” RFC 2827, 2000.

[5] F. Hsu and T. Chiueh, “A Path Information Cashing and Aggregation Approach to Traffic Source Identification,” in *Proc. of IEEE International Conference on Distributed Computing Systems (ICDCS)*, May 2003.

[6] J. Li, J. Mirkovic, M. Wang, P. Reiher, and L. Zhang., “SAVE: Source address validity enforcement protocol,” In *Proc. of IEEE INFOCOMM 2002*, April, 2002.

[7] D. McGuire and B. Krebs, “Attack on Internet called largest ever,” in *www.washingtonpost.com*, Oct. 2002. http://www.washingtonpost.com/wp-dyn/articles/A828-2002Oct22.html.

[8] K. Park and H. Lee, “On the Effectiveness of Probabilistic Packet Marking for IP Traceback under Denial of Service Attack,” in *Proc. of IEEE INFOCOM 2001*, Mar. 2001.

[9] K. Park and H. Lee, “On the Effectiveness of Route-Based Packet filtering for Distributed DoS Attack Prevention in Power-Law Internets,” in *Proc. ACM SIGCOMM ’01*, pp. 15-26, 2001.

[10] S. Savage, D. Wetherall, A. Karlin and T. Anderson, “Practical network support for IP traceback,” in *Proc. of ACM SIGCOMM*, Aug. 2000, pp. 295-306.

[11] A. C. Snoeren, C. Partiridge, L. A. Sanchez, C. E. Jones, F. Tchhakountio, S. T. Kent, and W. T. Strayer, ”Hash-Based IP TraceBack,” in *Proc. of ACM SIGCOMM*, Aug. 2001.