

Proving Distributed Denial of Service Attacks in the Internet

Prashanth Radhakrishnan, Manu Awasthi, Chitra Aravamudhan
{shanth, manua, caravamu}@cs.utah.edu

Abstract

In this course report, we present the problem of proving a distributed denial of service attack in the Internet. We propose a solution using probabilistic packet marking by routers, combined with a new technique to track individual DDoS attack flows. Our solution incurs very less overhead in the router's critical path and operates in the presence of legacy routers. We provide evaluation results on our solution's accuracy in the presence of more than 2000 attackers and upto 90% legacy routers in the network.

1 Introduction

E-Crime is on the rise. The annual losses incurred from the spread of digital pests are estimated to be of the order of several billions of dollars. Distributed Denial of Service (*DDoS*) is becoming increasingly frequent due to easy access to malicious software on the Internet. With the burgeoning of *Internet Mafia* [2], an increasing number of Denial of Service attacks are being carried out for economic reasons, by deploying rootkits on the machines of unsuspecting users connected to the Internet. These rootkits allow the attackers to recruit a large army of bots (or zombie) machines to be used for DDoS.

This growing prevalence of criminally motivated DDoS attacks calls for a fundamental rethink on how enterprises approach security. Companies typically bolster the security infrastructure only after they are attacked. But this approach is misguided and costly. Today there are several ways to mitigate the DDoS problem upfront, namely: network ingress filtering, pushback schemes etc. Since none of these solutions are comprehensive, the DDoS insurance model gains significance. A number of insurance companies are now providing computer security related policies. Under some circumstances the insurance model may provide a better return on investment (ROI) than some of the other defence mechanisms. Since the hackers are getting more sophisticated, no single DDoS mitigation scheme could guarantee complete protection.

However, one of the prime issues when dealing with an insurance model is the validation of the client's insurance claim. This is no different in the case of the DDoS security insurance model because the victim could potentially forge an attack on himself by generating the packets. This

necessitates a DDoS proving mechanism which the insurance provider can use to validate the victim's claim. We address this problem of proving a DDoS attack by proposing a solution that uses the technique of probabilistic packet marking by routers, coupled with a DDoS attack flow identification scheme.

Note that a DDoS attack could still be "setup" to fool the insurance company. We consider this scenario outside the scope of our work, because the DDoS attack actually occurs in this case.

The next section discusses existing research that relates to our proposed solution. Section 3 describes our envisioned insurance model usage scenario and the related assumptions that we make. The subsequent three sections discuss the key components of our solution, namely: packet marking, packet verification and identification of DDoS attack flows. The evaluation results are presented in section 7. We conclude in section 8 with a discussion on future work.

2 Related Work

Probabilistic packet marking (*PPM*) at the routers was proposed as an effective strategy for identifying the attack source. Burch and Cheswick [3] were the first to introduce the concept. Their method was however rudimentary in the sense that they traced back by selectively flooding the network links and monitoring changes in the attack traffic. Their method was based on the premise that flooding the links would cause the attack packets to be dropped more than the regular traffic packets and hence by observing changes in all the upstream links, one could achieve traceback.

Probabilistically marking the packets using some of the extra space in the IP headers was proposed by Savage et al [1]. They proposed *Fragment Marking Scheme* (FMS) and suggested the use of the 16 bit IP identification field in the IP header. The receiver reconstructs the routers on the attack path by making use of these markings.

iTrace [4] was proposed by Bellovin et al. *iTrace* involved probabilistic sending of a message to either the source or the destination of the IP packet indicating the IP address of the router (that sent the message). The main benefit of this scheme was that it did not require changes to packets in-flight, but it also suffered from the drawback of generating extra traffic.

Goodrich [5] proposed a marking scheme which marks nodes, instead of links, in the packets. This approach does not use a distance field to identify the distance of a marking router. It has issues with attack graph reconstruction and also doesn't scale to large number of nodes.

Dean et al [6] proposed a scheme to encode the PPM router's IP address as a polynomial in the IP identification field. It also suffers from the inability to scale. A theoretical analysis of traceback was provided by Adler [7]. His work presented a one-bit marking scheme. Its mainly of theoretical interest and also suffers from scalability problems.

SPIE [8] was proposed by Snoeren et al. It is a mechanism which uses router state to track the

path of a single packet. The main advantage of *SPIE* is that it enables a victim to traceback a single packet by querying the state of upstream routers. The major disadvantage of the approach is that it required keeping track of a large amount of state at the routers. This mechanism was enhanced by Li et al [9] with techniques to lower the state-tracking overhead of the routers, at the expense of extra communication overhead.

As Yaar et al [11] analyze all the above schemes, they point out the various shortcomings that most of these schemes suffer from. Some of them are enumerated below:

- **Incremental Deployment:** One cannot expect a scheme, which proposes changes to the existing hardware/software architecture to be deployed overnight. Instead, the scheme should be such that it could be deployed incrementally and still give good performance, rather than depending on every router to change configuration before starting to function.
- **Change at Routers:** The change required to the software and the hardware architecture of the routers should be small. This is where schemes such as *SPIE* fail because they require large amount of memory and computation overhead to function properly.
- **Few Packets:** Few packets should be required to traceback to the attacker. Most schemes do not satisfy this requirement.
- **Other Issues:** Another major problem that the traceback schemes need to address is that of a *pollution attack*. Packet marking schemes that do not use a distance field, are susceptible to their data being polluted by malicious attackers, because the marking schemes cannot distinguish between fragments generated by the marking router and the attacker. In this manner, by inserting an appropriate number of *polluted* packets in the stream, the attacker can thwart the traceback scheme.

Our packet marking approach handles incremental deployment, requires minimal change to routers and does not suffer from pollution attack. Further, since our purpose is to prove DDoS, the number of marked packets should bear some correspondence to the magnitude of the DDoS attack.

3 Envisioned Use Model

In our proposed technique, the victim keeps a log of all the packets that it received during its DDoS period. The DDoS attack interval and the packet dump is submitted to the *validator*, a trusted insurance provider entity, as evidence of the DDoS attack.

Some of the routers on the path from the attacker to the victim are capable of probabilistically marking packets in an authenticatable fashion. This requirement that all the routers on the path to the victim need not have the packet marking capability makes this scheme incrementally deployable. The marked packets are authenticated by the *validator*.

We assume that the *validator* has a map of all the upstream routers from the victim, as is done in [10]. To be able to authenticate markings, we also assume that the *validator* shares a secret

key with each PPM router. This sharing of secret with each PPM router may not be impractical, considering that only a small fraction of all routers would have the packet marking capability. However if it does get intractable, we propose to explore the *time-released key chains approach* [10].

As a result of authenticating the packet markings, the *validator* is able to verify that the packets indeed came from the network and were not generated by the victim to fake a DDoS attack. It then calculates the magnitude of the DDoS attack, *i.e.* the total number of packets sent to the victim. The scope of our work ends here and more specifically, we do not characterize whether a certain number of received packets constitutes a potential DDoS attack or not.

4 Packet Marking

In this section we describe our packet marking scheme. We assume the existence of an upstream router map from the victim. This can be relaxed by using the map reconstruction approach proposed in [11]. All PPM routers mark with a fixed probability p . This fixed probability is critical to the attack flow identification algorithm.

We use 30 bits in the existing IP header for packet marking: 16 bits of the *Identification* field and 14 bits of the *Fragment Offset* field. The rationale for using these fields is that IP fragmentation is very rare [1]. We leave out the *Do Not Fragment* bit which should be set to 1, since we want to prevent fragmentation and the *Reserved* bit which should be set to 0. Figure 1 illustrates the components of the marking scheme.

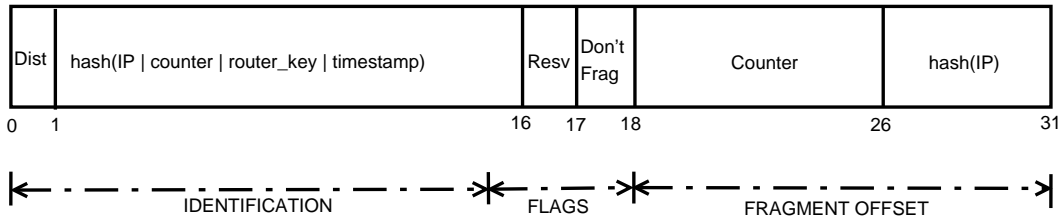


Figure 1: Packet Marking Format

Efficient identification of a PPM router during reconstruction requires encoding the distance of the router as well as its IP. For encoding distance, we use a scheme similar to the one proposed in [11], where just one bit of the marking field is used for representing the router distance. Every PPM router sets the 5 least-significant bits of the packet’s TTL field to a global constant c , and stores the sixth bit of the TTL in the distance field b . When a packet arrives at its destination, the distance at which the packet was marked is computed as: $d = (b|c - TTL[5..0]) \bmod 64$, where $b|c$ denotes concatenation of the one bit distance field b with the five bit TTL replacement constant c . Because legacy routers decrement the TTL, the FIT distance is representative of the exact number of hops from a PPM router. For more details, we refer the readers to [11].

We also include a 5 bit hash of IP address. This is more of a hint for speeding up the identification process and a false positive (i.e. a different IP address having the same hash value) does not affect the correctness.

To prove a DDoS attack, we need to identify that the packets came from the network and also verify that they were sent within the victim specified time-interval. For the former, we need to include a keyed hash in the marking. For the latter, routers should encode their current time in the marking. The packet contents need not be unique. Thus, to prevent replay attacks, the router should either not mark more than one packet per timestamp or have some other way of ensuring uniqueness of the packet marking. The granularity of the timestamp represents a trade-off between the number of packets that can be marked in an interval versus the time taken to verify each discrete timestamp value. Seconds granularity is too coarse, whereas the reconstruction complexity with microsecond granularity would be prohibitive. We thus choose millisecond granularity timestamp with an 8 bit counter, that lets routers to mark upto 256 packets in one millisecond. The counter is placed in the marking, in plain-text. This helps improve the number of packets marked within an interval without increasing the reconstruction time.

Note that all marking fields are independent of the contents of the packet. Thus the marking for each interval can be precomputed on the routers, thus taking hash computations out of the routers' critical path.

5 Packet Verification

Packet verification involves: confirming that a marked packet originated from the network and identifying the marking timestamp.

We first identify the potential routers by comparing the *hash(IP)* of all PPM routers at the calculated distance from the victim, with that in the packet. For each such router we next compare the keyed hash result for different values of the timestamp, ranging over the reported DDoS attack period. A match here identifies the PPM router, with very low probability of a false positive. The algorithm, that would be run on the *validator*, follows:

Given:

```
Marked packet: Pkt
Claimed DDoS interval: [reported_DDoS_start_time, reported_DDoS_end_time]
Reporter clock skew: reporter_time_skew
For each Router:
    Router clock skew: router_time_skew
    Packet delay: rtt_router_to_reporter
Window to accomodate a small margin of error in time calculation: DELTA
```

```
iter_start_time = reported_DDoS_start_time + reporter_time_skew - DELTA
iter_end_time = reported_DDoS_end_time + reporter_time_skew + DELTA
dist = (b|c - TTL[5..0]) mod 64
```

```

for (each PPM router at dist)
  for (each router whose hash(IP) matches that in the Pkt)
    for (i from iter_start_time to iter_end_time) {
      t = (i + router_time_skew) - rtt_router_to_reporter / 2
      if (hash(IP, Pkt.Counter, Key, t) matches with that in the Pkt) {
        IP marked the packet Pkt at timestamp t !!
        break
      }
    }
}

```

The algorithm above assumes knowledge of clock skews of all packet marking routers and the victim with respect to the *validator* and the round trip delays between PPM routers and the victim. It first converts the reported DDoS time interval to *validator's* time. And while trying to identify the time of marking, timestamps are translated from the *validator's* time to each router's time, while also accounting for the packet delay.

We realize that this algorithm has much scope for optimizations. For instance, the search space of potential routers could be progressively reduced by correlating them with previously identified routers in the router map. Since our focus is on demonstrating the correctness of our solution, we defer the exercise of optimizing the packet verification algorithm to future work.

6 DDoS Attack Flow Identification

In the previous section we dealt with the problem of identifying the marking routers and the time of marking, from a marked packet. We observe that an inference on DDoS magnitude based on just the total number of valid marked packets in the victim's packet dump, would be inaccurate. Note that the *validator* cannot rely on the non-marked packets from the victim, because they could have been faked. The number of marked packets received at the victim largely depends on the number of PPM routers on the path from the attackers to the victim. For instance, the victim could receive the same number of marked packets in both the following cases. First, there are 5 attackers with one packet marking router in each of their paths to the victim. Second, there is just 1 attacker with all routers on the path to the victim being packet marking enabled. Thus, we need a mechanism to accurately estimate the magnitude of the DDoS attack.

6.1 Estimating packets from each router

We noted earlier that each packet marking router marks with a constant probability p . If x packets were received marked by a particular PPM router during an interval, then the router must have forwarded atleast $\frac{x}{p(1-p)^{h-1}}$ packets during that interval. Here, h is number of PPM router hops from the router to the victim. During DDoS, this packet count would approximately be the total number of DDoS packets forwarded by the router to the victim. This estimate for different routers would overlap when the routers lie on the same attack path. For this reason, we need to track individual attack flows.

6.2 Identifying packet flow

If y marked packets of a single attack flow are received from a router R_1 , then $\frac{y}{(1-p)^h}$ marked packets of the same attack flow will be received from router R_2 , that lies on R_1 's path to the victim and is h PPM router hop-counts away from R_1 . By removing $\frac{y}{(1-p)^h}$ from the actual number of marked packets received from R_2 , we can account for the marked packets belonging to this attack flow. This way, we can handle packet count overlaps and identify unique attack flows. The exact algorithm works as follows, to produce *Total Count*, which is the estimate of the DDoS attack packets.

Let N_i be the number of marked packets from packet marking router i
Let W be a small window to accommodate the inaccuracy of probabilistic values
for (each PPM router i)

$$C_i = N_i$$

```
While (there exists a PPM router with  $|C_i| > W$ ) {
  Let  $i$  be the farthest PPM router with  $|C_i| > W$ 
  for (each PPM router  $j$  on the path from  $i$  to the victim) {
    Let  $h$  be the PPM router hopcount between  $i$  and  $j$ 
     $C_j = C_j - \frac{C_i}{(1-p)^h}$ 
  }
  Let  $x$  be the PPM router hopcount between  $i$  and victim
   $Total\ Count += \frac{C_i}{p(1-p)^{x-1}}$ 
}
```

The above algorithm makes an assumption that there is a unique path from each marking router to the victim. This assumption can be relaxed by adding a few bits of path identification information to packet marking, as is done in [12]. We leave this for future work.

7 Evaluation

To evaluate our work, we performed simulations of the above algorithms.

Our packet marking scheme has mostly been borrowed from existing work [11], that has evaluated this scheme. Although our packet reconstruction algorithm involves identifying timestamps, unlike other existing schemes, it is currently quite inefficient. While we have verified its correctness, we have not evaluated its performance. Thus our focus is predominantly on evaluating the accuracy of the DDoS attack flow identification algorithm, that we have introduced in this report.

7.1 Simulation Framework

Our simulation framework comprises of four modules. The *map generation module* generates an upstream router map from the victim, by randomly creating child nodes, legacy routers and attackers, all driven by specified probabilities. Given an upstream router map from the victim and a list of attackers, the *packet generation module* generates packets (that form the victim's

evidence to the *validator*) by simulating packet transfer from the attacker to the victim. The *packet verification module* performs the role of the *validator* by identifying the marking router and the packet marking time by running the packet verification algorithm. Finally, the *flow identification module* uses the results from the *packet verification module* to identify the DDoS attack flows, by using the algorithm described in the previous section.

7.2 Simulation Parameters

We ran our simulations on contrived datasets designed to exercise some aspects of our algorithms and on those that are randomly generated.

In all the following simulations, unless specified otherwise, the following configuration is implied: router map of 9 levels with a total of about 9840 routers, DDoS packet generation rate of 50 packets/ms, marking probability of 0.04, DDoS attack duration of 5 seconds and absence of legacy routers.

7.3 Effect of Marking Probability

We ran simulations by varying the marking probability from 0.04 upto 0.96 at different attack rates: 50, 100, 200 and 500 packets/ms (and a single attacker).

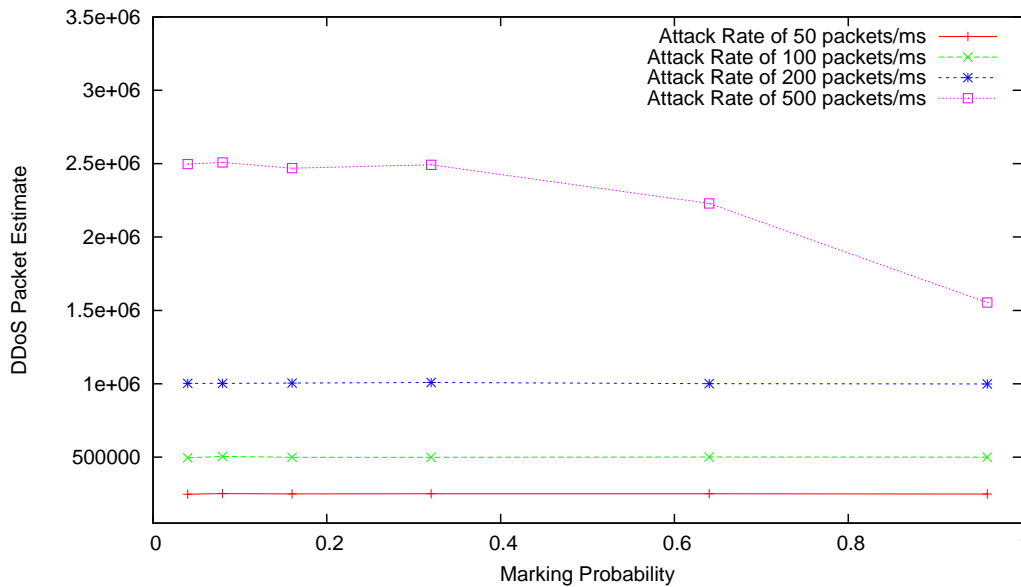


Figure 2: Effect of Marking Probability on DDoS Packet Estimate

Figure 2 plots the estimated number of DDoS packets sent to the victim. The horizontal lines would correspond to the actual DDoS packets sent. For all attack rates other than 500, the estimate follows the actual DDoS packet count with a maximum deviation of 1%. When the attack rate is 500 packets/ms, the accuracy of the estimate degrades with increasing probability (37%

deviation at a marking probability of 0.96). As noted before, each router marks only 256 packets in one millisecond. At large attack rates and very high marking probabilities, the counter gets saturated resulting in poorer estimation of DDoS packets.

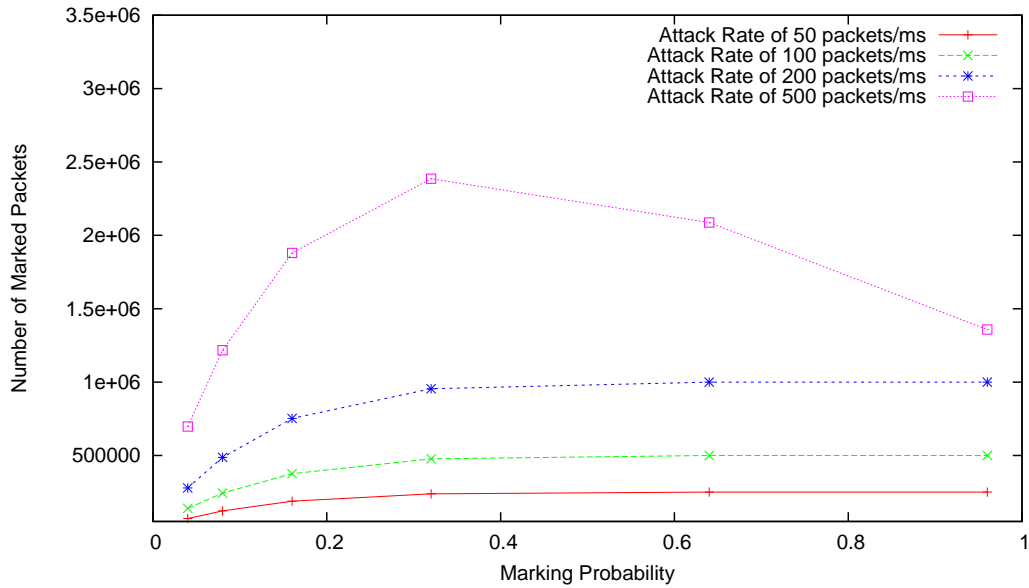


Figure 3: Effect of Marking Probability on Marked Packets Received

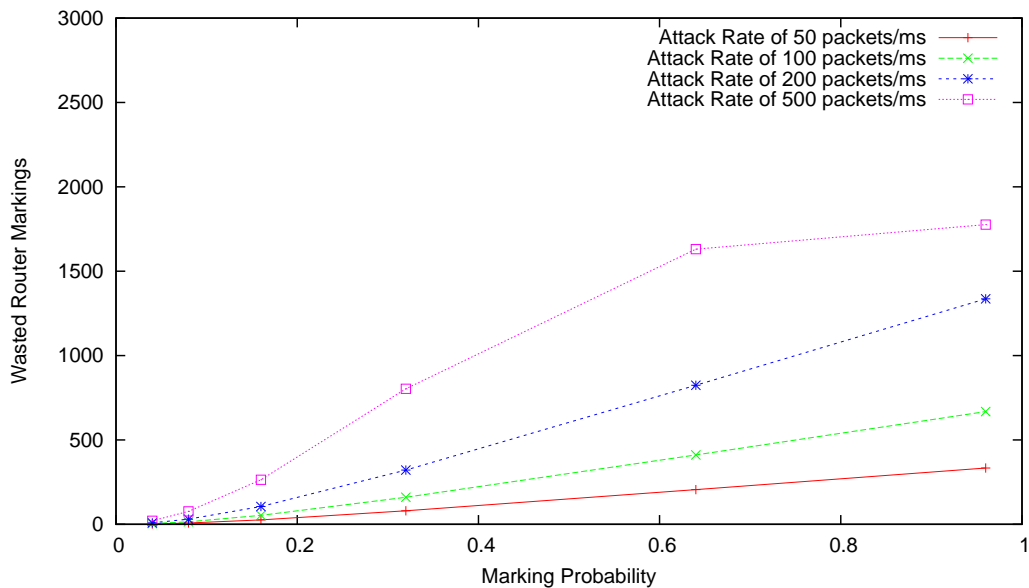


Figure 4: Effect of Marking Probability on Wasted Markings

Figure 3 plots the number of marked packets received. This number increases with increasing probability, as expected, and reaches saturation as the number of marked packets gets closer to

the total packets sent. The 500 packets/ms case shows anomalous behaviour. As the marking probability increases beyond 0.375, the number of marked packets drops. This is possibly due to the marking clashes, i.e. multiple routers marking the same packet and wasting their millisecond counts. To verify this, we plot the number of such wasted markings in Figure 4.

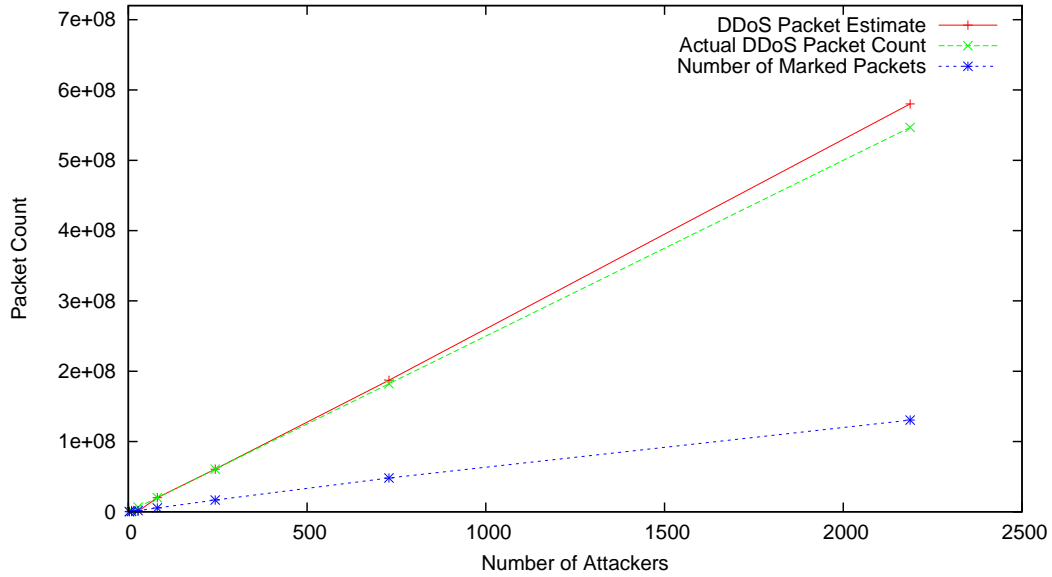


Figure 5: Effect of Number of Attackers

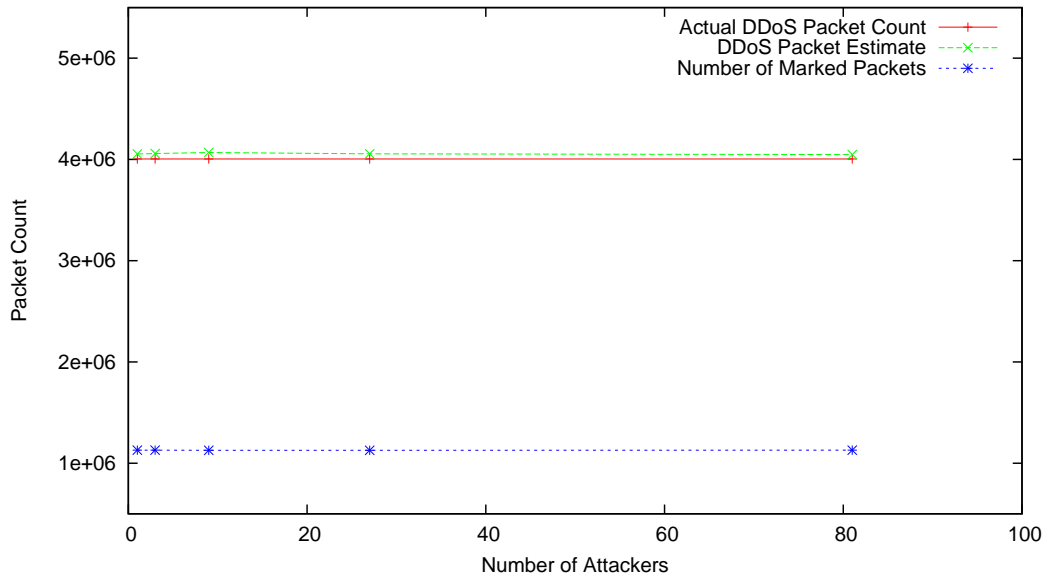


Figure 6: Effect of Distribution of Attack Flows

7.4 Effect of Number of Attackers

We ran simulations by varying the number of attackers upto 2187. Figure 5 plots the estimated number of DDoS packets and the number of marked packets received. The results show that our DDoS flow detection algorithm has about 94% accuracy even in the presence of 2187 attackers. As expected, the number of marked packets increases with attacker count. But the rate of increase falls progressively (for instance, between 3 and 9 attackers the number of marked packets increased by 3.01. Whereas between 729 and 2187 the increase falls to 2.71). This is because of millisecond count saturation, especially at routers closer to the victim, where different attack flows converge.

7.5 Effect of Distribution of Attack Flows

The purpose of this simulation was to see the effect of distributing the DDoS packets across attack flows, on the accuracy of DDoS packet estimation. We proportionately decrease the rate of attack with increasing number of attackers (up till 81), to keep the number of attack packets constant. Figure 6 shows that the estimate is quite accurate throughout, with a maximum deviation of 1.58% from the actual value.

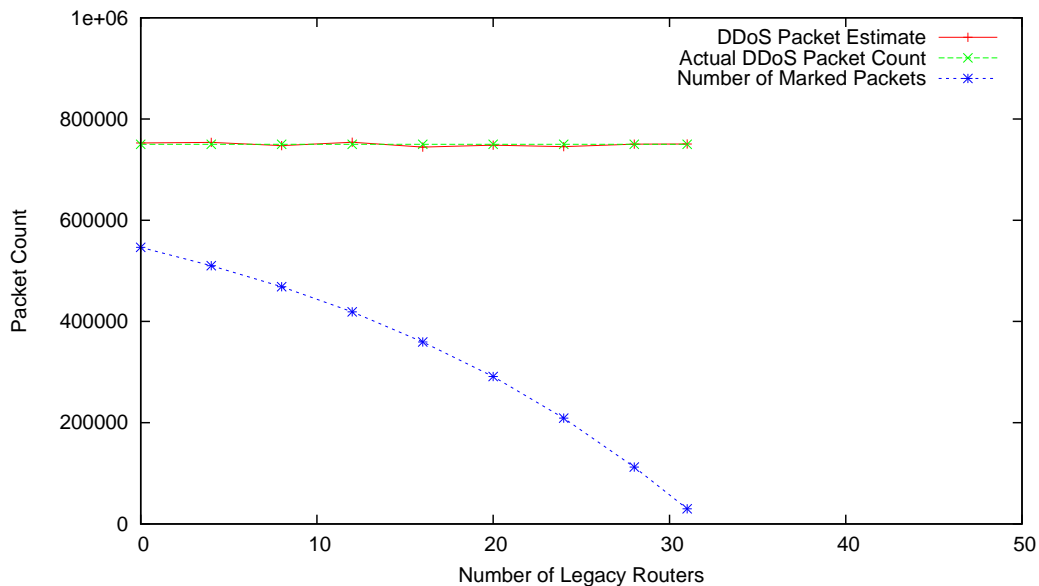


Figure 7: Effect of Legacy Routers in the Attack Path

7.6 Effect of Legacy Routers in the Attack Path

We performed this experiment on a contrived graph with a single attacker 32 hops away from the victim. The estimate of DDoS packets and number of marked packets received was measured with varying number of legacy routers (Figure 7). The results were again accurate, with a maximum deviation of 0.7% from the actual value.

7.7 Effect of Randomized Placement of Legacy Routers and Attackers

Real-world is characterized by randomized placement of legacy routers and attackers. Some attack flows in the DDoS attack might not have any PPM router in their path to the victim. We evaluate the performance of our algorithms in such randomized environments.

The experiments were all run in the same 9840-router graph, but with varying ratio of legacy routers and number of attackers. In all these experiments, we neither had control nor knowledge of placement of the attackers with respect to the PPM routers.

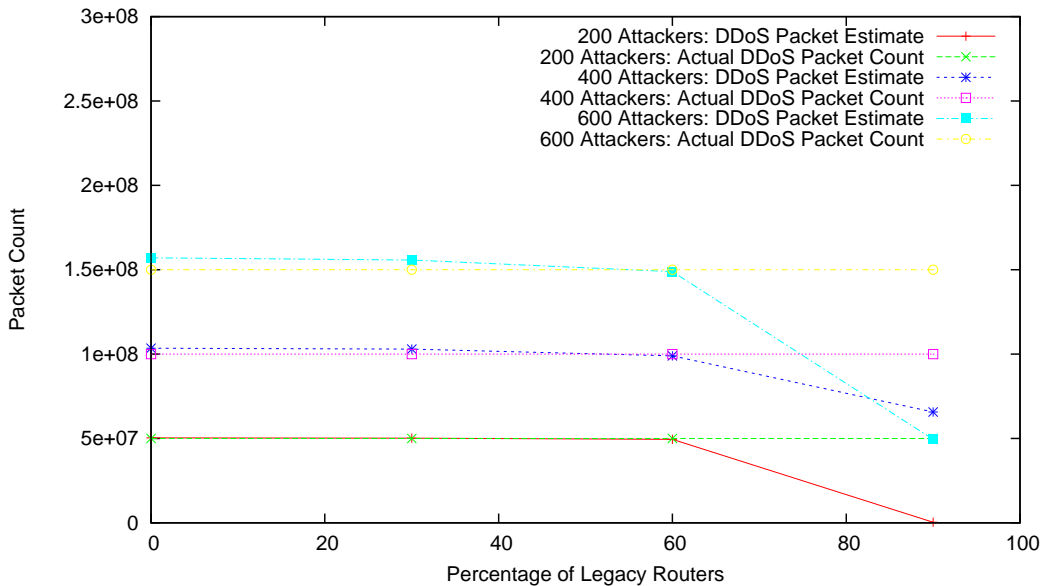


Figure 8: Effect of Randomized Placement of Legacy Routers and Attackers

Figure 8 plots the measured estimates of DDoS packets along with the actual number of DDoS packets. It is seen that until about 60% of legacy routers in the network the estimates are consistently accurate. Beyond that, accuracy steadily degrades and at 90% legacy routers its quite untractable.

Another interesting result is that the accuracy at 30% or even at 60% of legacy routers is always better than 0% legacy routers. We think this is due to the millisecond counter saturation at routers. That the inaccuracy steadily increases with increasing number of attackers lends credence to this.

8 Conclusions and Future Work

Our contributions in this work were two-fold. First, we have identified the problem of proving DDoS, discussed a potential usage scenario where it could be important and applicable and have

proposed a solution to the problem. Second, we have proposed a DDoS attack flow identification algorithm that estimates the DDoS attack magnitude with very high accuracy even in the presence of more than 2000 attackers and at 60% legacy routers in the network.

As future work on this, we would like to relax some restrictive assumptions that we have made and also try optimizing some algorithms. Here is a summary of the future work items, along with the approach we propose to adopt:

- **Relax the assumption on upstream router map.** Explore using the map reconstruction approach in [11].
- **Optimize the packet verification algorithm.**
- **Relax the assumption on a unique path from PPM routers to victim.** Add a few bits of path identification information in packet marking like in [12].
- **Design a practical key management scheme between the *validator* and PPM routers.** Explore using the *time-released key chains approach* in [10].

References

- [1] Stefan Savage, David Wetherall, Anna R. Karlin and Tom Anderson, *Practical network support for IP traceback*, SIGCOMM, 295-306, 2000.
- [2] <http://www.eweek.com/article2/0,1895,1947561,00.asp>
- [3] Hal Burch and Bill Cheswick, *Tracing anonymous packets to their approximate source*, Unpublished paper, December 1999.
- [4] S. Bellovin, M. Leech, and T. Taylor, *The ICMP traceback message* Internet-Draft, October 2001. Work in progress, available at <ftp://ftp.ietf.org/>
- [5] Michael Goodrich. *Efficient packet marking for large-scale IP traceback*, In Proceedings of the 9th ACM Conference on Computer and Communications Security, November 2001.
- [6] Drew Dean, Matt Franklin, and Adam Stubblefield. *An algebraic approach to IP traceback* ACM Transactions on Information and System Security, May 2002.
- [7] Micah Adler. *Tradeoffs in probabilistic packet marking for IP traceback*, In Proceedings of 34th ACM Symposium on Theory of Computing (STOC), 2002.
- [8] Alex C. Snoeren, Craig Partridge, Luis A. Sanchez, Christine E. Jones, Fabrice Tchakountio, Stephen T. Kent, and W. Timothy Strayer. *Hash-based IP traceback*, In Proceedings of ACM SIGCOMM 2001, August 2001.
- [9] J. Li, M. Sung, J. Xu, and L. Li. *Large-scale IP traceback in high-speed Internet: Practical techniques and theoretical foundation*, In Proceedings of the IEEE Symposium on Security and Privacy, May 2004.

- [10] Dawn X. Song, Adrian Perrig. *Advanced and Authenticated Marking Schemes for IP Traceback* In Proceedings of the IEEE Infocomm 2001.
- [11] A. Yaar, A. Perrig and D. Song. *FIT: Fast Internet Traceback*, IEEE Infocom 2005
- [12] A. Yaar, A. Perrig and D. Song. *Pi: A Path Identification Mechanism to Defend against DDoS Attacks*, In IEEE Symposium on Security and Privacy, May 2003