Unified Defense against DDoS Attacks

M. Muthuprasanna, G. Manimaran, Z. Wang

Iowa State University, Ames, IA, USA - 50011
{muthu, gmani, zhengdao}@iastate.edu

Abstract. With DoS/DDoS attacks emerging as one of the primary security threats in today's Internet, the search is on for an efficient DDoS defense mechanism that would provide attack prevention, mitigation and traceback features, in as few packets as possible and with no collateral damage. Although several techniques have been proposed to tackle this growing menace, there exists no effective solution to date, due to the growing sophistication of the attacks and also the increasingly complex Internet architecture. In this paper, we propose an unified framework that integrates traceback and mitigation capabilities for an effective attack defense. Some significant aspects of our approach include: (1) a novel *data cube model* to represent the traceback information, and its slicing along the lines of path signatures rather than router signatures, (2) characterizing traceback as a transmission scheduling problem on the data cube representation, and achieving scheduling optimality using a novel metric called *utility*, (3) and finally an information delivery architecture employing both packet marking and data logging in a distributed manner to achieve faster response times. The proposed scheme can thus provide both per-packet mitigation and multi-packet traceback capabilities due to effective data slicing of the cube, and can attain higher detection speeds due to novel utility rate analysis. We also contrast this unified scheme with other well-known schemes in literature to understand the performance tradeoffs, while providing an experimental evaluation of the proposed scheme on real data sets.

1 Introduction

Denial-of-Service (DoS) attacks are a menace we have come to live with in today's Internet. Distributed DoS attacks on several sites including Yahoo and eBay, and against root DNS servers had virtually paralyzed the Internet in early 2000s. Recent attacks motivated by political and economic reasons on SCO, RIAA, 2Checkout, Blue Security, EveryDNS etc. have established a disturbing trend, and unless we address this issue now, there might soon be an avalanche of DDoS attacks crippling the entire Internet infrastructure. The stateless nature and destination-oriented routing of the Internet makes tracking of attackers, employing source address spoofing, a difficult problem to address [1] [2]. The need of the hour is a technique that not only traces attackers but also aids in effective mitigation of the ongoing attacks [3].

To be realistically applicable on an Internet scale, the proposed schemes must be incrementally deployable, scalable, require minimal changes to existing

hardware, maintain high accuracy during large volume attacks, resist tampering by spoofed data injection, and require very few packets to complete traceback while also simultaneously triggering mitigation, amongst other requirements. The traceback schemes in literature address the problem of collecting information about individual packet forwarding agents and collating this data to obtain attack source/path statistics; while the mitigation schemes address the problem of dropping malicious packets using the concept of path identifiers and do not overtly concern themselves with the identification of the attackers themselves.

We analyze the traceback problem here as independent data representation and data transmission issues. We also propose a novel data model and metric to help us better evaluate traceback schemes. We also exploit the concept of path signatures here so that the proposed traceback scheme can additionally support effective mitigation, thus realistically obtaining the better of the two worlds.

The rest of the paper is organized as follows. Section 2 reviews the different schemes known, while Section 3 presents the basic principle and motivation behind the proposed solution. Sections 4 and 5 discuss the data representation and the data transmission phases respectively. Section 6 analyzes the performance tradeoffs, while Section 7 concludes.

2 Related Work

The earliest work on DDoS defense led to the concept of network traceback [4] by Burch and Cheswick. Bellovin et.al. proposed ICMP-based out-of-band messaging in iTrace [5], while Snoeren et.al. proposed SPIE [2] employing packet logging, which was subsequently improved by Li et.al. in [6]. Belenky and Ansari proposed a deterministic packet marking scheme in [7], while Savage et.al. proposed a probabilistic packet marking (PPM) technique in [1], with subsequent enhancements made by others in [8] [9] [10] [11]. IP address fragmentation for efficient packet marking and their vulnerability to attacker induced noise have been studied in [12] and [13] respectively. Recently, various encoding techniques have been used to progressively improve the performance of PPM schemes, as in Tabu marking [14], Local Topology marking [15], Space-Time encoding [16], Color Coding [17], and the use of Huffman Codes [18], Algebraic Geometric Codes [19] etc. Additionally various architectures for traceback have been explored, such as inter-domain traceback [20] and hybrid traceback [21] [22], in addition to some other radical approaches like in [23].

Research on mitigating DDoS attacks has proceeded in parallel, focusing on network ingress filtering [24], routing table enhancements as in SAVE [25], CenterTrack [26] and intelligent filtering [3]. The concept of path fingerprints was exploited by Yaar et.al. in [27], and subsequently improved in [28]. Various other techniques involving path filtering [29] [30], statistical filtering [31] [32], and rate limiting [33] [34] have also been explored in literature. Of late, the focus has been on unifying these strategies for greater deployment incentives and better defense capabilities for the Internet.

3 Motivation

We discuss a few reasons here that motivate and also form the basis of our proposed solution. Our aim here is to provide a single mechanism that supports both traceback and mitigation simultaneously with no additional overhead.

Unified Operation: Traditional PPM traceback schemes capture per-hop behavior, i.e. they transmit information about each intermediate router to the victim so that a global view of the routing path is eventually obtained over multiple packets. Thus traceback data is incrementally gathered and does not function on a per-packet basis; where each packet corresponds to some router identifier fragment, and state maintenance is needed for mitigation. Similarly, traditional mitigation schemes capture end-to-end behavior, i.e. at some level of abstraction they generate fingerprints that identify the routing paths as opposed to routers themselves. Thus mitigation operates on a per-packet granularity, but lossy abstraction makes it impossible to perform accurate traceback [27].

Our approach entails *slicing* the entire traceback data (later modeled as a cube) along the lines of path signatures instead of individual router identifiers. Thus we obtain multiple signatures per routing path useful for mitigation at lower thresholds on a per-packet basis, and when viewed collectively yields the required traceback data too. Thus we build a homogeneous technique that seamlessly provides both traceback and mitigation capabilities, operating in parallel.

Traditional Restrictions: As the number of bits in the IP headers that can be overloaded to support traceback in the Internet is limited [1], we not only need to reduce the total number of bits to be marked, but also the number of bits marked per packet. Additionally to support faster traceback, we also need to mark these bits across fewer packets. Thus an ideal scheme would need to optimize these three different metrics simultaneously.

Practical Deployment: We assume that the victim has an upstream router graph [11] that serves as a lookup table for suspect routing paths. Hence, the scope of the problem is limited to identifying a unique path (essentially unique set of connected routers) in the router graph and not the entire Internet. As this is not an exhaustive set, we need to obtain only a certain subset of all router identifier fragments, that ensures uniqueness in the router graph. Note that there exist multiple such subsets of different sizes in the router graph.

Use of router identifier fragments as base data unit might lead to transmitting more fragments than required to uniquely identify a certain router, thereby wasting entire packets at times. In contrast, use of path signatures as base data unit ensures that every router has some data to convey to the victim in every packet. Thus we ensure maximum data diversity for the victim by increasing the average information (utility) gained per bit transmitted, and achieve traceback in far fewer (bits) packets due to better (bit) packet space utilization.

We address these issues in two main phases - the *data representation phase* where we address the different theoretical and technical issues by defining a novel data cube model for traceback and a new metric called utility to analyze

the model, and the *data transmission phase* where we address the practical implementation issues by employing limited data logging at intermediate routers.

4 Data Representation Phase

We present our data cube model and other fundamental concepts here that form the crux of our solution. The question we answer here is which bits do we prefer to send, as opposed to how we send them as addressed by Adler in [9].

Data Cube: The total information (in bits) needed by the victim to completely reconstruct a suspect routing path is called the *Traceback Data* and we represent it using the *Data Cube* in Fig.1(a). The *y-axis* in the cube represents the different routers along the routing path, while the *x-axis* represents the different fragments of the corresponding router identifiers. The *z-axis* in the cube represents the different bits in a particular fragment of the router identifier. Thus the cube represents the complete traceback data about the suspect Internet routing path. The shaded region in Fig.1(a) shows the router identifier for a particular router, including the different bits in all the different fragments. This 3-D representation helps us analyze the optimality of the different traceback schemes known.

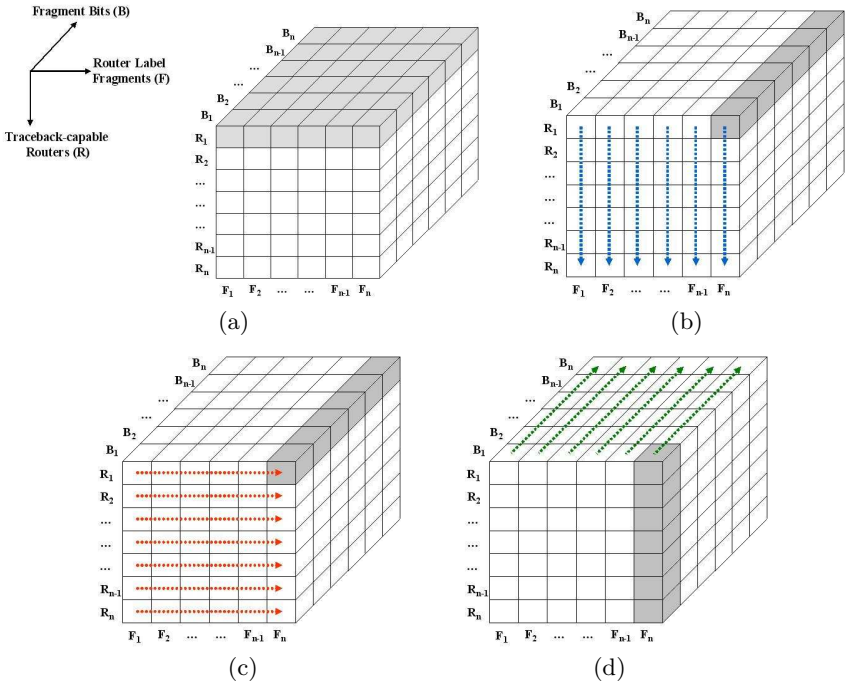

Fig. 1. Traceback Data Cube Model & Different Transmission Schedules

When a DoS attack is launched, the search space for the suspect routing path is the entire Internet (or rather the entire upstream router graph). The victim

then prunes this domain to a unique path, based on the traceback data it receives in an incremental fashion. It is obvious that the speed of the traceback operation is thus determined by the rate of receiving useful incremental information. Thus the transmission schedule used to transmit the entire traceback data to the victim plays a critical role in determining its response time to attacks. We define a metric called *Utility*, which helps us choose an optimal transmission schedule for the traceback data cube. The basic idea is to associate each of the bits in the data cube with different weights (or utilities), such that an objective of maximizing speed (or total utility) yields a natural optimal transmission (sequence) schedule.

Utility: We define *Information Utility* of a particular bit [35] as the ratio of the reduction in the search space due to the receipt of that particular bit to the total original search space (Eqn.1). The search space for a particular router having a k-bit identifier is evidently 2^k. Receipt of a single bit reduces the search space (uncertainty) to 2^{k-1}. A second subsequent bit reduces it to 2^{k-2} and so on.

$$Utility\,(u_i) = \frac{Search\,Space\,Reduction_{\,i}}{Total\,Search\,Space} \tag{1}$$

Thus the utilities of the first, second and m^{th}-bits of a k-bit message are as shown in Eqns.2,3. The terms first, second and so on here do not refer to the digits in the MSB or LSB of the binary number, rather they refer to that bit (any one of the k bits) that was transmitted first, second and so on.

$$u_1 = \frac{2^k - 2^{k-1}}{2^k} = \frac{1}{2} \quad ; \quad u_2 = \frac{2^{k-1} - 2^{k-2}}{2^k} = \frac{1}{2^2} \tag{2}$$

$$u_m = \frac{2^{k-(m-1)} - 2^{k-m}}{2^k} = \frac{1}{2^m} \tag{3}$$

The total utility of all bits of a message is shown in Eqn.4. The $\frac{1}{2^k}$ represents the *self-information* of a k-bit message. In other words, it represents the probability of finding that particular k-bit message in the entire search space, and is the implicit information it possesses. Thus the total utility of the data cube having r intermediate routers is $\approx r$, where each router identifier has utility ≈ 1.

$$\sum_{i=1}^{k} u_i = \sum_{i=1}^{k} \frac{1}{2^i} = 1 - \frac{1}{2^k} \tag{4}$$

We thus see that not every bit of information in the traceback data cube has the same utility, as has been assumed by researchers to date. *The utility of a bit is governed by how much information has already been transmitted to the victim about a particular slice (router identifier) in the cube.* Higher the utility achieved in a certain fixed number of packets, smaller the search space for the suspect routing path, and hence faster the traceback process. We now use these novel concepts of data cube and utility, to derive higher utility rate transmission schedules for the data cube and hence faster traceback schemes.

Utility Analysis: We analyze the basic PPM based traceback scheme [1] here, as it has been the most widely studied technique in literature, and also as its analysis holds true for all the different flavors of PPM schemes proposed to date. We quantify the utility rate of the basic PPM scheme clearly demarcating its operating region in the utility space, thus illustrating its sub-optimal performance.

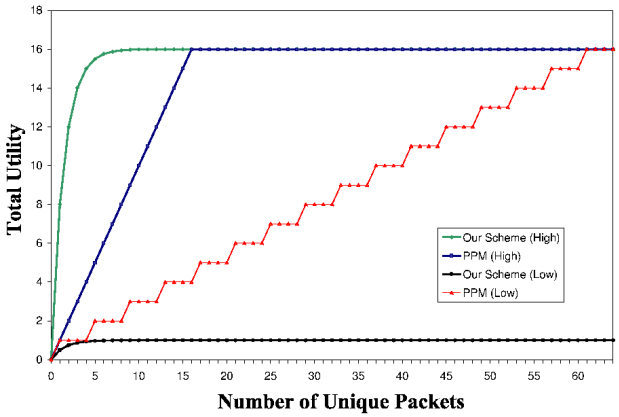

Fig. 2. Utility Rate Comparison

The 3-D data cube when viewed as a 2-D surface (xy-plane), each unit cell corresponds to a vector (z-axis) as indicated in Figs.1(b),1(c). The PPM scheme now reduces to randomly choosing a unit cell (with replacement) from the 2-D surface and sending the corresponding bit-vector to the victim. The receipt of all bit-vectors according to some transmission schedule concludes the traceback process. If the r router identifiers are each split into k fragments, each m bits long, then the utilities of the first, second and t^{th} fragments for each router identifier are as shown in Eqns.5,6.

$$u_{F_1} = (1 - \frac{1}{2^m}) \quad ; \quad u_{F_2} = (\frac{1}{2^m}) * (1 - \frac{1}{2^m}) \tag{5}$$

$$u_{F_t} = \sum_{i=m(t-1)+1}^{mt} u_i = (\frac{1}{2^{m(t-1)}}) * (1 - \frac{1}{2^m}) \tag{6}$$

Hence, highest (lowest) utility rate is achieved when all high (low) utility fragments are scheduled first for transmission. Consider the transmission schedule in Fig.1(b) where all routers send their first fragment, then their second fragment, and so on. Also consider another transmission schedule in Fig.1(c) where a particular router sends all its fragments before the next router is allowed to transmit, as in an implicit token passing system. These represent the highest and lowest utility rate transmission schedules respectively. It is easily seen that any other transmission schedule attains a utility rate in the operating region bounded by the inner two curves in Fig.2.

Proposed Transmission Schedule: Now consider a different transmission schedule, where we again view the data cube as a 2-D surface (xz-plane), where each unit cell corresponds to a vector (y-axis) as indicated in Fig.1(d). We thus *logically* move away from the concept of individual router identifiers to that of path signatures. All the routers now send their first bit, then their second bit and so on in each packet. Any packet now has only one bit from each router in order, and is hence classified as a path fingerprint (signature). The utilities for the first, second, and t^{th} packets are as shown in Eqn.7.

$$u_{P_1} = \frac{m}{2} \quad ; \quad u_{P_2} = \frac{m}{2^2} \quad ; \quad u_{P_t} = m * \frac{1}{2^t} = \frac{m}{2^t} \tag{7}$$

The operating region in utility space of the proposed transmission schedule is bounded by the outer two curves in Fig.2. The lower bound here is caused by the *Leader Selection Problem* discussed later. In Fig.2, we assume r=16 (routers), k=4 (fragments), m=16 (bits), for illustrative purposes only.

5 Data Transmission Phase

Implementing the proposed scheme requires distributed coordination by all the intermediate routers to resolve mainly 2 issues. Firstly, each router needs to know the exact bit position in the IP headers that it is supposed to mark based on its distance from the victim. Also, the first router closest to the attacker, needs to decide which one of the multiple path signatures is to be marked on a certain packet by all the subsequent routers along the suspect routing path.

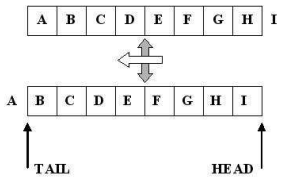

Fig. 3. Traceback Buffer Implementation

We adopt a queue implementation to resolve the first problem [28]. Any new router mark is pushed at the buffer head, causing some bit at the tail to overflow (Fig.3). Thus we ensure that without any explicit messaging, we can retain the marks of the last k routers in the traceback field in the IP packet header.

To prevent the queue from being hijacked by attacker injected spoofed data in the absence of overflow, we would need the first router along the routing path to reset the traceback buffer. Additionally, it would also need to probabilistically choose a particular path signature to be marked on a certain packet. As there is no consensus on how we determine the first router in the research community, we dilute the definition of a *first router* to represent the router farthest from the victim in a trust domain (ISP PoP, gateway router etc.) [12]. Although this limits the depth of traceback that can be achieved, the mitigation accuracy is enhanced due to virtually no data tampering by the attackers.

Leader Selection Problem: As an alternative to the trust domain, we could also *sub-optimally* allow each router to probabilistically choose to be the leader (or first router). This probabilistic selection was first explored in [1]. The probabilistic worst case where the router closest to the victim always chooses to be the leader results in the lower bound for the proposed scheme in Fig.2.

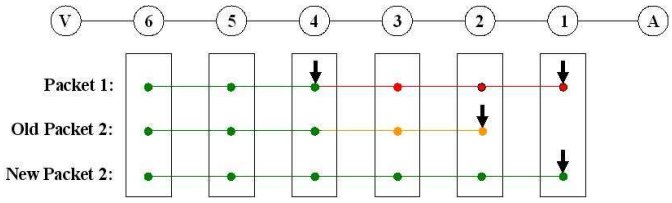

Fig. 4. Hybrid Traceback Architecture

Hybrid Architecture: To improve the average case performance of *Probabilistic Leader Selection (PLS)*, we now propose the hybrid traceback architecture (Fig.4), employing both packet logging and marking techniques [21]. Here, we additionally require the leader to cache the current traceback buffer and the path signature index, before overwriting them. Every other (non-leader) router during marking compares the current traceback buffer and index with its cache (if present), and on a prefix match caches and then marks the longer prefix.

The example in Fig.4 shows a packet where both routers 1 and 4 contend to be the leader, and consequently marks from routers 1,2,3 are lost (but cached at router 4). In a subsequent packet, where router 2 is the leader, router 4 can augment the traceback buffer with cached mark of router 1 also. To keep the storage overhead small, routers can optionally also choose to turn this caching off. Thus selective pipelined distributed network storage of traceback data yields a significant improvement in achieving higher transmission utility rates, and hence faster traceback using far fewer packets.

6 Analysis & Experimental Evaluation

We compare the basic PPM based traceback scheme and the proposed utility based scheme here, while also evaluating the effect of the hybrid architecture.

Number of Packets (w.r.t. PPM): Let there be r routers along the routing path, each having a marking probability of p, and whose identifiers each have k m-bit fragments. The probability of receiving a mark from a router i hops away is $P_i(M_m)$ (Eqn.8(a)). If we conservatively assume that marks from all routers appear with same likelihood as the furthest router, then probability that a packet delivers a mark from some router is $P(M_m)$ (Eqn.8(b)). From generalized Coupon Collector Problem [36] [37] in Probability Theory, and detailed analyses in [1] [17], number of packets X_m needed to reconstruct the routing path from all the different fragments for PPM based scheme is given by Eqn.9.

$$P_i(M_m) = p(1-p)^{i-1} \;\; ; \;\; P(M_m) \geq rp(1-p)^{r-1} \tag{8}$$

$$E[X_m] < \frac{k * log_e(kr)}{p(1-p)^{r-1}} \tag{9}$$

For the proposed utility based scheme (known leader), we have km (r-bit) fragments (vectors), and hence the number of packets X_r needed to reconstruct the routing path from all the different packets is given by Eqn.10.

$$E[X_r] < \frac{k * log_e(km)}{p(1-p)^{r-1}} \tag{10}$$

If we choose $r{=}m$, then our proposed scheme performs no worse than the PPM based schemes in reconstructing the suspect routing paths from the information obtained from all the different fragments (packets).

Utility Rate (w.r.t. PPM): Often we need to shortlist a few candidate suspect routing paths and not identify all unique attack paths during high volume DDoS attacks, as the immediate need is always to choose optimal (upstream) network locations where mitigation may be deployed. The constraint is usually to identify the most likely suspect paths in the top 2-5 percentile. Such a constraint not only reduces the effect of the inaccuracies of the traceback process, but also reduces the response time and operational complexity of the mitigation process.

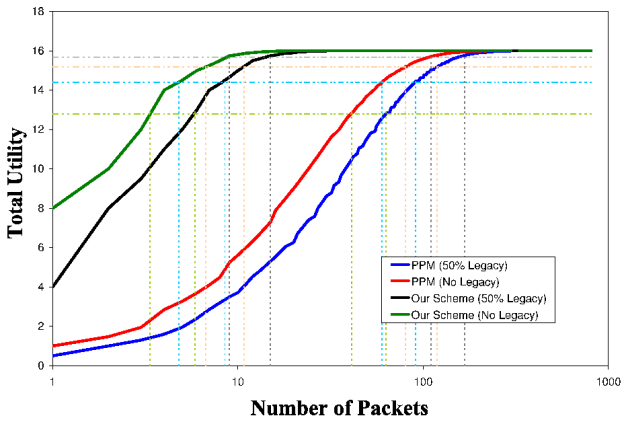

Fig. 5. Top k percentile Filtering

We performed an extensive evaluation of the basic PPM based scheme and the proposed utility based traceback scheme, first router being defined at the trust boundary, over a candidate router graph obtained from Rocketfuel [38]. We simulated 10,000 independent instances of DDoS attacks to obtain a high precision average case performance for the two schemes. Along the lines of the probabilistic calculations above, Fig.5 (Table 1) shows actual number of packets required to filter down to the top $2, 5, 10, 20$ percentile suspect routing paths for the two schemes. We evaluate 2 scenarios namely, *No Legacy* where all intermediate routers are traceback capable, and *50% Legacy* where half of them are legacy routers, incapable of traceback. It is to be noted that the performance degradation even in the face of limited deployment is very minimal. To trigger the

constraints, the number of packets required by our proposed scheme is roughly an order of magnitude lesser than those required by the PPM based scheme.

Thus the proposed utility based scheme achieves significantly faster response times to DDoS attacks, while still retaining the same worst case performance (where we construct the unique attack path) as PPM based traceback schemes.

Percentile (16-bit pkt)	PPM (50% Legacy)	PPM (No Legacy)	Our Scheme (50% Legacy)	Our Scheme (No Legacy)
2	167	111	15	9
5	119	80	11	7
10	91	60	9	5
20	63	41	6	4

Table 1. Number of Packets for Traceback

Mitigation Strategy: We do not delve into this aspect in detail due to space constraints. The proposed scheme transmits some path signature in every packet to the victim. The victim can then choose to maintain independent thresholds for packet filtering for each of these signatures. Additionally, as it can correlate different path signatures as belonging to fixed suspect paths once traceback completes [11], these independent thresholds can then be appropriately scaled to reflect the detected correlation. Thus our unified framework not only provides both traceback & mitigation capabilities, but also speeds up traceback process.

Hybrid Architecture Effect: In an ideal situation, every protected server would be associated with some network trust boundary or leader, and we would obtain the best performance then as described above. In cases where this is not true, PLS (possibly with hybrid architecture) would provide a working alternative, although sub-optimal in nature to the scheme analyzed above.

The PPM schemes and the proposed scheme with known leader, assure us that every packet contains the entire corresponding bit-vector. However, with PLS, the leader always chooses to erase the traceback buffer, and hence the buffer contains only a subset of that bit-vector. For simplicity, we assume that PLS can equi-probably choose any router, and hence average bit-vector length would be half the number of routers along routing path ($\frac{r}{2}$). Hence, for the proposed scheme with PLS chosen leader, we have km (r-bit) fragments, and the number of packets X_r^1 needed to reconstruct routing path is given by Eqn.11.

$$E[X_r^1] < r * \frac{k * log_e(km)}{p(1-p)^{r-1}} \tag{11}$$

Using the router maps and parameters as above, we also evaluate the effect of the hybrid traceback architecture on the proposed utility based traceback scheme employing PLS. Fig.6 shows the actual number of packets needed to filter down to the top $2, 5, 10, 20$ percentile suspect routing paths for this scheme. As expected, we notice that the number of packets needed goes down sharply, as the required percentile accuracy reduces. We also notice that the number of packets

needed here is much higher than the basic utility based scheme with known leader, due to the probabilistic nature of selection as described in Eqns. 9, 10, 11.

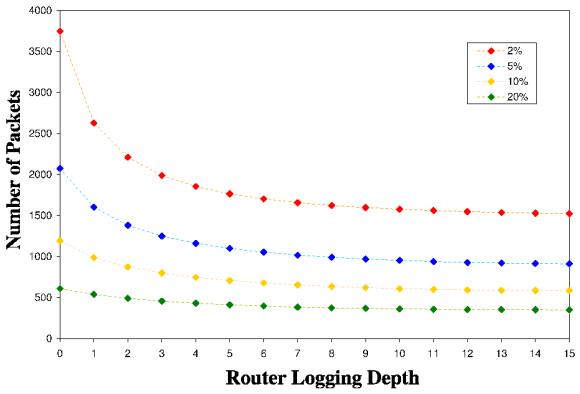

Fig. 6. Hybrid Architecture Effect

We additionally capture a metric called *Router Logging Depth*, which indicates number of upstream routers whose (bits) signatures are being cached. While depth 0 indicates no caching, depth n ($n \gg 0$) indicates maximum caching. It is easily seen that additional router storage for traceback logging increases linearly with increasing logging depth. However, we see diminishing returns with increasing logging depths in Fig.6, and thus individual routers can tune the cache size appropriately, without affecting the speed of the traceback process.

7 Conclusions

As malicious entities unleash an increasing number of DDoS attacks on the Internet, it has become imperative to not only track them to hold them liable (traceback), but also to limit their capabilities and render them ineffective (mitigation). In this paper, we propose an unified framework that provides both traceback and mitigation capabilities simultaneously, operating in parallel. We view traceback as essentially a data transmission problem where we send a data cube across, and analysis of this model using novel metrics helps us understand and design faster traceback schemes. We also exploit the concept of path signatures and hence support attack mitigation in an implicit manner. Thus our proposed defense mechanism unifies both traceback and mitigation paradigms into a single protocol yielding better deployment incentives and greater defense capabilities in today's Internet. It is easily seen that any traceback optimization in literature can easily be cast as an optimization along a certain dimension of the proposed data cube model. The orthogonality of the different dimensions indicates that these enhancements need not necessarily be mutually exclusive. As part of our future work, we plan to evaluate a concurrent deployment of the different orthogonal optimizations, to achieve even faster attack response times.

References

1. Savage et. al., "Practical Network Support for Traceback," in *SIGCOMM*, 2000.
2. A. C. Snoeren et. al., "Hash-Based IP Traceback," in *SIGCOMM*, 2001.
3. M. Sung, J. Xu, "Intelligent Packet Filtering: A Novel Technique for defending against DDoS Attacks", in *IEEE TPDS*, 2003.
4. H. Burch, B. Cheswick, "Tracing Anonymous Packets to their approximate source", in *Proc. USENIX LISA*, Dec. 2000.
5. S. M. Bellovin, "ICMP traceback messages," *Internet Draft*, Mar. 2000.
6. Li et. al., "Large-Scale IP Traceback in High-Speed Internet: Practical Techniques & Theoretical Foundation", in *IEEE Symp. on Security & Privacy* 2004.
7. A. Belenky, N. Ansari, "IP Traceback with Deterministic Packet Marking", in *IEEE Communication Letters*, Apr 2003.
8. D. Song, A. Perrig, "Advanced and Authenticated Marking Schemes for IP traceback", in *Proc. IEEE INFOCOM*, 2001.
9. M. Adler, "Tradeoffs in Probabilistic packet marking for IP traceback", in *Proc. STOC*, pp. 407-418, 2002.
10. T. Peng, C. Leckie, K. Ramamohanarao, "Adjusted Probabilistic Packet Marking for IP Traceback", in *Proc. Networking*, 2002.
11. A. Yaar, A. Perrig, D. Song, "FIT: Fast Internet Traceback", in *INFOCOM* 2005.
12. I. Hamadeh, G. Kesidis, "Performance of IP Address Fragmentation Strategies for DDoS traceback", in *Proc. IEEE IPOM*, 2003.
13. M. Waldvogel, "GOSSIB vs Traceback Rumors", in *ACSAC*, 2002.
14. M. Ma, "Tabu Marking Scheme for Traceback", in *IPDPS*, 2005.
15. B. Al-Duwairi, T. Daniels, "Topology Based Packet Marking," in *ICCCN* 2004.
16. M. Muthuprasanna, G. Manimaran, "Space-Time Encoding for DDoS Attack Traceback", in *Proc. IEEE GLOBECOM*, 2005.
17. M. Muthuprasanna, G. Manimaran, Mansoor Alicherry, Vijay Kumar, "Coloring the Internet: IP Traceback", in *Proc. ICPADS*, 2006.
18. K. Choi, H. Dai, "A Marking Scheme using Huffman Codes for IP Traceback", in *Proc. ISPAN*, 2004.
19. C. Bai et.al., "Algebraic Geometric Code Based IP Traceback", in *IPCCC* 2004.
20. Y. Sawai, M. Oe, K. Iida, Y. Kadobayashi, "Performance Evaluation of Inter-Domain IP Traceback", in *Proc. IEEE ICT*, 2003.
21. B. Al-Duwairi, G. Manimaran, "Novel Hybrid Schemes employing Packet Marking & Logging for Traceback", in *IEEE TPDS*, 2005.
22. Gong et.al., "IP Traceback based on Packet Marking & Logging", in *ICC* 2005.
23. M. Walfish, M. Vutukuru, Hari Balakrishnan, D. Karger, Scott Shenker, "DDoS Defense by Offense", in *Proc. SIGCOMM*, 2006.
24. P. Ferguson, D. Senie, "Network ingress filtering: Defeating denial of service attacks which employ IP source address spoofing," in *RFC 2267*, 1998.
25. J. Li, J. Mirkovic, M. Wang, M. Reiher, L. Zhang, "SAVE: Source address validity enforcement protocol", in *Proc. of INFOCOM*, 2001.
26. R. Stone, "CenterTrack:An IP overlay network for tracking DoS floods", in *Proc. USENIX Security Symposium*, 2000.
27. A. Yaar, A. Perrig, D. Song, "Pi: A Path Identification Mechanism to defend against DDoS Attacks," in *Proc. IEEE Symposium on Security and Privacy*, 2003.
28. A. Yaar, A. Perrig, D. Song, "StackPi: New Packet Marking Filtering Mechanisms for DDoS & IP Spoofing Defense", in *JSAC*, pp.1853-1863, Oct. 2006.
29. C. Jin, H. Wang, K. G. Shin, "Hop-Count Filtering: An effective defense against spoofed DDoS traffic", in *ACM CCS*, 2003.
30. A. Keromytis, V. Misra, D. Rubenstein, "SOS: An architecture for mitigating DDoS attacks", in *IEEE JSAC*, pp. 176-188, Jan. 2004.
31. Y. Kim, W. Lau, M. Chuah, J. Chao, "PacketScore: A statistical-based overload control against DDoS attacks", in *Proc. IEEE INFOCOM*, 2004.
32. T. Peng, C. Leckie, K. Ramamohanarao, "Protection from DDoS attacks using history-based IP filtering", in *Proc. IEEE ICC*, 2003.
33. J. Ioannidis, S. M. Bellovin, "Implementing Pushback: Router-based defense against DDoS attacks", in *Proc. NDSS*, 2002.
34. D. Yau, J. Lui, F. Liang, "Defending against DDoS attacks with max-min fair server-centric router throttles", in *Proc. IWQoS*, 2002.
35. C. E. Shannon, "A mathematical theory of communication I & II", in *Bell System Technical Journal*, vol. 27, pp. 379-423 &623-656, 1948.
36. H. von Schelling, "Coupon Collecting for Unequal Probabilities", in *Proc. American Mathematical Monthly*, 1954.
37. S. Lu, S. Skiena, "Filling a Penny Album", in *Proc. CHANCE*, 2000.
38. N. Spring, R. Mahajan, D. Wetherall, "Measuring ISP Topologies with Rocketfuel", in *Proc. SIGCOMM*, 2002.