A Marking Scheme Using Huffman Codes for IP Traceback

K. H. Choi and H. K. Dai
Computer Science Department, Oklahoma State University
Stillwater, Oklahoma 74078, U. S. A.
{kyu, dai}@cs.okstate.edu

Abstract

In (Distributed) Denial of Service attack ((D)DoS), attackers send a huge number of packets with spoofed source addresses to disguise themselves toward a target host or network. Various IP traceback techniques such as link testing, marking, and logging to find out the real source of attacking packets have been proposed. We present a new marking scheme (with marking and traceback algorithms) in which a router marks a packet with a link that the packet came through. Links of a router are represented by Huffman codes according to the traffic distribution among the links. If the packet runs out of space allotted for the marking field in the packet header, then the router stores the marking field in the router's local memory along with a message digest of the packet. We analyze the memory requirement of routers to store marking fields, compare the new scheme with other existing techniques, and address practical issues to deploy the new scheme in the Internet. The new scheme marks every packet, therefore IP traceback can be accomplished with only a packet unlike in probabilistic markings; also it requires far less amount of memory compared to logging methods and is robust in case of DDoS.

1. Introduction

The anonymous nature of the Internet Protocol (IP) makes it difficult to identify the true source of an IP packet if the source uses fake address, hence distributed denial of service (DDoS) attacks have become more prevalent recently due to the relative ease of acquiring and executing such attacking tools and their near untraceability to the attacker. There are two ways to deal with (D)DoS attacks, the one is to detecting and discarding attacking packets on the way to their destinations, and the other is IP traceback to find the real source of attacking packets and then possibly make the attacker responsible. If a victim could find the path of attacking packets in real-time, it would be much easier to quickly stop the attack, and the possession of capability to trace back would somewhat deter attackers from launching (D)DoS attacks. The problem of traceback of spoofed packets has become a topic as a measure against (D)DoS attacks in the Internet world.

Various existing techniques for IP traceback have been reported in the literature (Internet Control Message Protocol (ICMP) traceback messages [1], link testing[2], marking [6] [8], and logging [7]). We propose a new technique that uses Huffman codes to mark packets with router's information as packets traverse routers during the journeys to reach their destinations. Simulation results and practical issues are also presented.

2. New Marking Scheme Using Huffman Codes

The new idea utilizes the following facts. First, routers are able to know on which physical network interface port packets arrive, this ability is used in ingress filtering and input debugging of routers. Second, each router is connected with not so many adjacent routers, in a router-level Internet map the average degree (the number of neighboring routers of a router) is 3.15 [4].

There are two differences in the proposed method from other marking methods. Firstly when a router marks a packet with address information, the information is not of the router that is marking but of a router that sent the packet to the current router, and secondly it uses a special table called link table, which shows all the links between the router and its adjacent routers. The router appends to the marking field a Huffman codeword representing the link number of the link (router) through which the packet arrived. When the marking field of a packet becomes short of space left to append the corresponding Huffman codeword for the link number, the router stores the content of the marking field with a message digest of the packet into the router's local memory, and then clears the field and appends the codeword. The stored link sequence can be retrieved via the message digest of the packet from the intermediate router during an IP traceback procedure.

2.1. Encoding of Marking Field

Figure 1 shows encodings of 32-bit marking field, in format (a) marking field is divided into a 1-bit saved flag (*sf*), a 26-bit link sequence (*ls*), and a 5-bit length of link sequence (*lls*), and in format (b) it is into a 1-bit saved flag (*sf*) and a 31-bit link sequence (*ls*). To reduce the possibility that the marking field has to be stored at intermediate routers' local

Proceedings of the 7th International Symposium on Parallel Architectures, Algorithms and Networks (ISPAN'04)
1087-4089/04 $ 20.00 © 2004 IEEE

memory, it is required to assign a longer field to the link sequence. So instead of using *lls* to specify the length of bit-string (sequence of link codes) in the field *ls*, we use bit 1 as a delimiter with leading 0s to designate the start position of the valid bit-string. When a packet passes through a router, *ls* is augmented with a codeword that represents a link through which the packet came in. Before appending the reversal of the codeword at the right end of *ls*, the router checks if there is enough bit-space left in *ls* to append the codeword by counting the leading 0s before the delimiter in *ls*. Figure 2 presents the marking algorithm at a router.

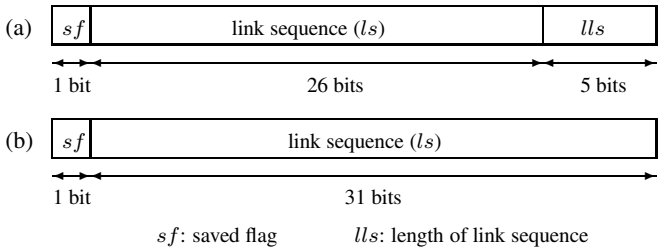


Figure 1. Encoding of the marking field. Format (a) uses *lls* to specify the length of bit-string in *ls*, while format (b) uses a delimiter bit 1 at the leftmost of the valid sequence of link codes in *ls*.

2.2. Storing Marking Fields at Intermediate Routers

Because of the limited space of *ls* in the marking field we may not be able to store the complete link sequence of a path. After a router has determined that the bit-space left in the *ls* is not enough for appending a codeword, the router stores the contents of the marking field in its local memory, which is indexed by the message digest of the packet, denoted by MD(packet). After saving, *l*s will be cleared by setting with 0x01 having only delimiter bit 1 at the rightmost bit, and *sf* is set to 1 indicating that the marking field is stored.

Possible Packet Transformations
If a packet undergoes a transformation after the marking has been saved (when *sf* is 1), then a router can not retrieve the stored marking field unless it knows the message digest of the packet before the transformation. Therefore when *sf* is 1 and a transformation happens, a router should store a pair of digests of old and new packets MD(new packet):MD(old packet) along with the marking field, and clear *ls* by setting with 0x01, but *sf* remains 1.

2.3. Traceback Procedure

Starting from a router that is directly connected with a victim, the victim can traceback a packet by decoding the link sequence (*ls*) in the marking field of the packet. When decoding a codeword the victim consults the link table of current router to find the upstream router that forwarded the packet to the current router. After a codeword has been decoded, *ls* will be right-shifted times of the length of the decoded codeword. When *ls* become 1 (only with a delimiter at the rightmost bit) and *sf* is 1, the stored marking field should be retrieved via the message digest of the packet. Now the upstream router becomes current router and the traceback continues until *ls* becomes 1 and *sf* becomes 0. Figure 2 presents the traceback algorithm at a victim.

2.4. Representation of Links

To reduce the length of the marking field in the IP packet header and the times link sequence has to be stored in intermediate routers due to the lack of space left in the marking field during the marking procedure, we use Huffman codes, which is widely used to compress data by assigning shorter codewords to higher-frequency characters and longer codewords to lower-frequency characters, to represent the link numbers. For a router, each link between itself and one of its adjacent routers has a relative number (frequency) of packets coming into the router through the link, and using the frequencies of packets we can assign a Huffman codeword to each link. Table 1 shows an example where the number of links is 5 and the average number of bits to represent a link with unequal distribution is 2.04 while fixed-length representation requires 3 bits. Figure 3 illustrates two Huffman trees each with equal and unequal distribution of packets among 5 links of a router of Table 1.

Table 1. An example of distribution and corresponding codes for links with degree 5.

link number	1	2	3	4	5
unequal distribution	45	34	10	8	3
equal distribution	255	255	255	255	255
fixed-length codes	000	001	010	011	100
Huffman codes for unequal distribution	1	00	011	0100	0101
Huffman codes for equal distribution	110	111	00	01	10

2.5. Organization of Link Tables

The link table of a router is a file that is supposed to be accessed by a victim to decode a Huffman codeword to find an upstream router on the attacking path. All routers must have agreed structure for their link tables. Figure 4 shows a possible structure of a link table. In the structure, number of links is the number of adjacent routers directly connected with a router and the frequency of each link is the relative number of packets coming through the link. The number of links and frequencies of each link are represented by one byte for each, and IP addresses of routers are 4 bytes long.

2.6. Encoding of Marking Field in the IP Header

Some fields of IP header must be used as the marking field. The Option field of IP packet looks most adequate but in [6] the Identification field of IP header is used to store path information on account of that less than 0.25% of packets undergo fragmentation [9]. If the IP Identification field is used for marking then the original function (reassembling fragmented packets by inspecting the Identification field of packets) of the field will be impeded. Using the Option field is

Marking procedure at a router with a packet P

```
Determine the link that packet P came from and its Huffman codeword representation by consulting the link table;

if (sf == 1 and packet P (old_P) transformed into a different packet (new_P))
   then store MD(new_P):MD(old_P)(sf, ls)     // store at local memory
        ls = 0x01                             // clear ls by setting with 000...01

Count the leading 0s (space_left) in ls to know how many bits are available in ls;
if (space_left < length(codeword))      // not enough space left in link sequence ls
   then store MD(P):(sf, ls)            // MD(P): message digest of packet P
        sf = 1, ls = 0x01               // marking field saved, ls cleared

Append codeword to ls;                  // append codeword to the link sequence ls
```

Traceback procedure at a victim with a packet P

```
Starting at the closest router (current router) with which the victim is directly connected;

while (1) {
  Print current router;

  Construct Huffman tree with the link table of current router;
  Decode one Huffman codeword from the right end of ls by using the tree;
  Find the router that the decoded codeword represents;

  if (ls == 0x01 and sf == 1)    // marking field is stored at current router's memory
     then retrieve MD(P):MD(pre_P)(sf, ls) or MD(P):(sf, ls)
          reset sf, ls with retrieved values

  if (ls == 0x01 and sf == 0)
     then break                 // stop traceback, no more link sequence to decode

  Set current router with the found router;
}
```

Figure 2. Marking and traceback procedures at a router and a victim, respectively, with a packet.

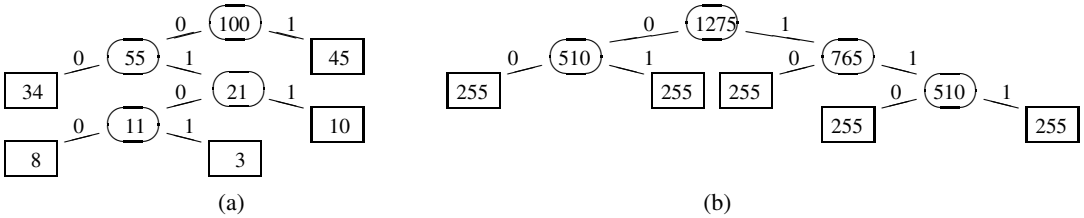


Figure 3. The Huffman trees for the distributions of Table 1: (a) Unequal distribution: Frequencies of packets arriving through each link are respectively 45, 34, 10, 8, and 3. The compression rate over the fixed-length representation is 61.2% $((45 \times 1 \text{ bit} + 34 \times 2 \text{ bits} + 10 \times 3 \text{ bits} + 8 \times 4 \text{ bits} + 3 \times 4 \text{ bits})/(45 + 34 + 10 + 8 + 3) \times 3 \text{ bits})$; (b) Equal distribution: Frequencies of packets arriving through each link are all the same with 255. The compression rate over the fixed-length representation is 80% $((20 \times 2 \text{ bits} + 20 \times 2 \text{ bits} + 20 \times 2 \text{ bits} + 20 \times 3 \text{ bits} + 20 \times 3 \text{ bits})/(45 + 34 + 10 + 8 + 3) \times 3 \text{ bits})$.

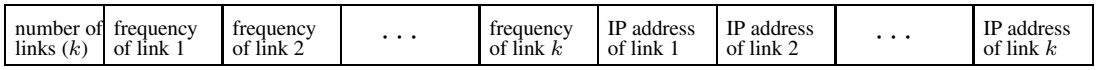


Figure 4. Structure of link table.

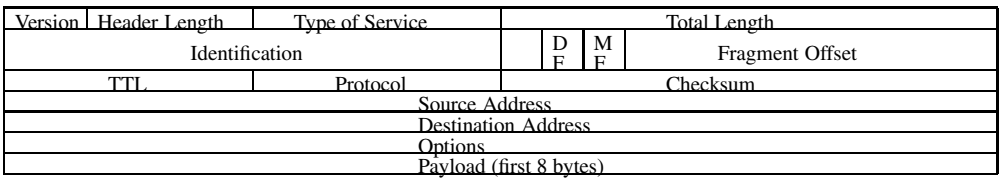


Figure 5. (Adopted from [7]) The fields of an IP packet. The fields: Type of Service (ToS), Time to Live (TTL), Checksum, and IP Options are masked out before digesting.

not supported practically because the Option field has rarely been used in reality and most of routers that are running currently in the Internet cannot handle the Option field. Even though they could handle the Option field there remain still other problems like increasing possibility of fragmentation of packets due to increased size by using the Option field, because basically the Option field does not have an assigned fixed length space in IP header as its name means literally. Therefore this study does not propose a certain field to use for the marking field.

2.7. The Message Digest Algorithm

When routers store a marking field it will index the marking field with a message digest of the packet. If we choose a message digest algorithm with longer output (64-bits or 128-bits) then routers need to have more memory space to store the digest with along a marking field. Using one of existing digest algorithms with adequate output length is the easiest way. There are many message digest algorithms and if we adopt MD5 [5] then we can use only 32 bits of 124 output bits, for example by selecting every forth bit of the output. As explained later in the memory requirement section of the results of simulation, at a high-end router with capacity of 1 Tera bits per second (bps) with 1-minute period of keeping of marking fields, the number of 32-bit marking fields that has to be stored is 1200 Mega. Therefore with 32-bit marking field, the probability that a message digest collides with a stored one is $1.2/4$ because there are 4 Giga digests with 32 bits long.

Fields of IP Packets Used as Inputs of Message Digest
When routers compute a digest MD(P) of a packet P, the input of the digest algorithm is not the whole packet, only some parts of the packet are used to reduce the processing time and because some fields of IP header changes as the packet passes through routers. In [7] as shown in Figure 5, ToS, TTL, Checksum, and Options of IP header will be masked out before digesting, and the first 8 bytes of the payload are used as input. But for the new marking scheme, in addition to the fields masked out in Figure 5, fields that are used for marking field are masked out too before digesting.

2.8. Compromised Intermediate Routers

In a (D)DoS, there are possibly compromised intermediate routers on the path of attacking packets, and they could mess up or carefully manipulate the marking field of a packet. However compromised routers cannot affect the marking after them, and the victim can traceback correctly at least up to a compromised router that is the closest to the victim on the path.

3. Simulation

Simulation has been done to see whether new idea works correctly and analyze mainly memory requirement of the new idea. To imitate a packet flow in the Internet, first a packet, not a real IP packet but a data structure having a marking field, is created, then this packet traverses a certain number of routers (hops). Before the packet reaches a router the router is created by assigning an IP address, a link table including degree (number of links), IP addresses of neighboring routers that links connect with the router, and distribution of packets (frequencies) among the links, then Huffman codes for the links are constructed by creating a Huffman tree using the distribution, and one of the links is chosen randomly assuming that the packet comes in through the chosen link. Finally it marks the packet with a Huffman codeword representing the chosen link. When a packet reaches its destination, IP traceback of this packet may be accomplished from the last router. During a traceback, at each router, a Huffman tree is created with packet distribution, and one codeword is decoded from the marking field of the packet. With the decoded information the next upstream router's IP address is found consulting the link table of the current router. The traceback continues at the found router until there is no link sequence left in the marking field.

For the analysis of memory requirement of routers, all information about created routers and Huffman codes for the links, such as distance, degree, length of Huffman codes, and length of complete link sequence were collected.

3.1. Average Lengths of Huffman Codes and Link Sequences

The average length of codewords increases in proportion to the average degree. Table 2 and Figure 6 show the average length of codewords for degrees from 2 to 6. For average degree 3, the average length is 1.44 and 1.56 bits for unequal distribution and equal distribution respectively, and for average degree 4 it is 1.77 and 1.95 bits. In equal distribution all links of a router were given 255 the same frequency of incoming packets, and in unequal distribution the frequency of incoming packets through each link is differently given in the range of $1\sim255$ by random.

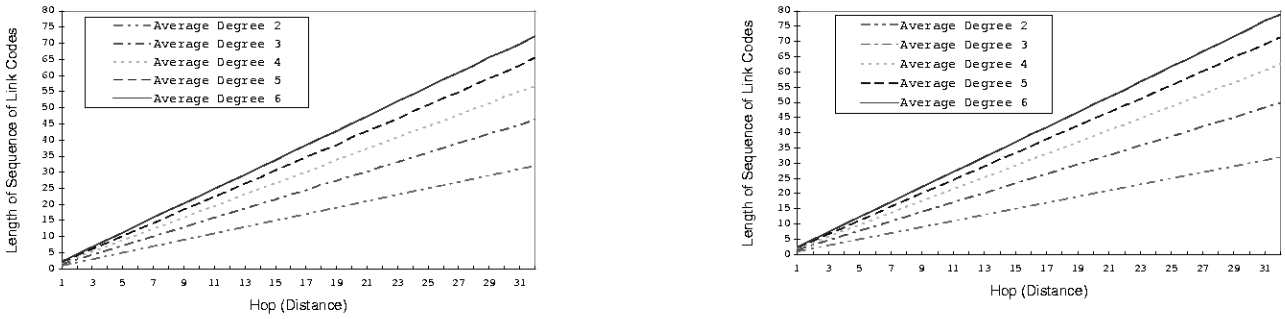
Table 2. Average length of Huffman codewords with average degrees 2, 3, 4, 5, and 6.

average degree		2	3	4	5	6
average length of codewords	equal distribution	1.00	1.56	1.95	2.23	2.47
	unequal distribution	1.00	1.44	1.77	2.03	2.26

Figure 7 shows the average length of complete link sequence with unequal and equal packet distribution among links. As like average length of Huffman codes, the average length of link sequence increases in proportion to the average degree and the distance (hop). The average length of link sequence with average degree 3, distance 16, and unequal packet distribution is 23.11 bits, and 24.95 bits with equal distribution.

3.2. Memory Requirement for Routers to Store Marking Fields

Memory requirement was analyzed for each 32-bit and 16-bit long marking field with equal incoming packet distribu-



(a) Unequal Distribution of Packets

(b) Equal Distribution of Packets

Figure 7. Average length of sequence of link codes: (a) With unequal distribution of packets among links: Links were given different frequencies in the range of 1∼255 randomly and one percentage of packets transformed; (b) With equal distribution of packets among links: Every link was given 255 the same frequency of packets and one percentage of packets transformed.

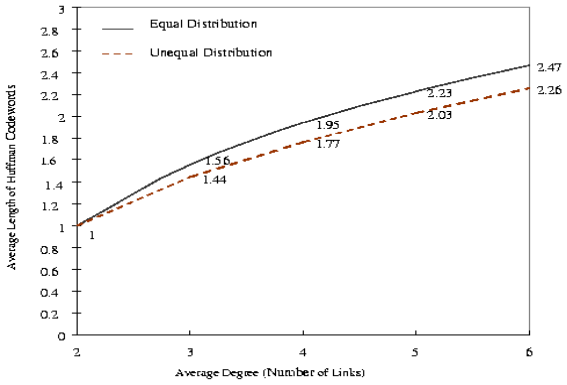


Figure 6. Average length of Huffman codewords for average degrees 2, 3, 4, 5, and 6 with one percentage of packet transformation.

tion among links because we cannot ensure that the routers' link tables are optimally tuned with the actual packet distribution. But the simulation has been done with each equal and unequal distribution. In equal distribution all the links were given the same frequency 255 of incoming packets, and in unequal distribution the links were given different frequencies in the range of 1∼255 randomly.

32-Bit Marking Field

Almost all paths are less than 32 hops and the average length of path (number of hops) is around 16 [10], and the average degree (average number of neighbors of a router) is slightly larger than 3 [4]. Therefore using 32 bits for marking field, with distance 16 hops and average number of links 3, the average length of complete sequence of link codes is 23.11 bits in Figure 7(a) and the probability that the marking field has to be stored is 0.002 in Figure 9(a) with unequal packet distribution among links. But with equal distribution where all the frequencies are same with 255, the average length of sequence of link codes is 24.95 bits in Figure 7(b) that is a little larger than that with unequal distribution, while the probability of saving of the marking field is 0.001 in Figure 9(b) because the length of Huffman codes is a little longer with $\lceil \log_2 n \rceil$ bits or $\lceil \log_2 n \rceil - 1$ bits for n links with same frequencies (255) of packets among n links while the average length of Huffman codes with unequal distribution is always less than with $\lceil \log_2 n \rceil$ bits.

In Figure 7(b) with average degree 3 at distance 21, the average length of complete sequence of link codes exceeds 30 bits that is the length of *ls* in the marking field and consequently the link sequence has to be stored. Therefore the probability that the marking field of a packet with average distance 16 and average degree 3 is stored at least once on the way to its destination is about $12/32$ (distance 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, and 32 out of 32 distances) that is 0.375 (the area in Figure 11(a) is about $1/3$). And the probability that a router on the path of 16 routers stores the marking field approximates to $0.375/16$, which means that 2.34% of marking fields are stored at a router.

With average degree 4, the probability that marking field is stored at least once is $16/32 = 0.5$ (distance 17, 18, . . ., 32 out of 32 distances) because at distance 17 the average link sequence exceeds 30 bits and the area in Figure 11(a) is about $1/2$. The probability of saving of marking field at a router out of average 16 routers is $0.5/16 = 3.1\%$.

Since the actual average degree in the Internet is between 3 and 4, the percentage of packets whose marking field is stored at a router is inferred less than 3%. Furthermore, as the distances of paths are distributed around 16 as shown in Figure 8 if we apply different weights according to their distribution the actual percentage will drop to less than 2%.

A high-end core router with capacity of 1 Tera bps that is 1 Giga packets/second with assumption that average packet size is 1 Kb will store 20 Mega marking fields per second. The memory required for the router is 160 MB/second (20 Mega $\times$ 64 bits $/8$ bits), which is 0.128% of router's ca-

pacity for keeping of marking fields for a second, and 7.68% for a minute. For a low-end router with capacity 1 Giga bps the requirement is 9.6 MB/minute.

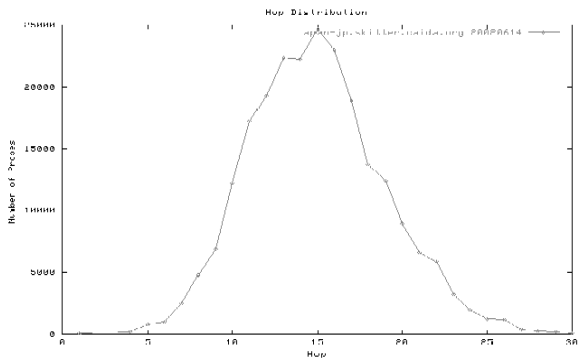


Figure 8. (Adopted from [3]) Hop distribution.

16-Bit Marking Field
Figure 10 shows the average count of savings of 16-bit marking field with each unequal and equal distribution, and Figure 11(b) shows only with degrees 3 and 4 with equal distribution of incoming packets among links. The average count of savings of marking field of a packet is about 2 according to Figure 11(b) with degree between 3 and 4. And even if we apply hop distribution (Figure 8) to Figure 11(b), the actual average count of savings will be some 2. The average count of savings of a packet at a router of 16 routers is $2/16$, which is 0.125 meaning that marking fields of 12.5% of packets that a router forwards are stored at this router. The memory requirement for a router with capacity of 1 Giga bps is 750 KB/second (0.125 Mega $\times (32+16)$ bits $/8$ bits) that is 0.6% of router's capacity for a second keeping and 36% for a minute.

Transformations
Transformation did not affect on the average length of codewords and the memory requirements due to the percentage of packets that undergo transformation is generally low, in the simulation the percentage were 0.1, 0.5, 1, 2, and 3%.

3.3. Comparison with Other Methods

1. Can trace both during an ongoing attack or postmortem: Like other marking methods this new method allows a victim traceback a packet both during an attack and after an attack has been completed, provided that the link sequence still remains in intermediate routers in case that intermediate routers stored the sequence.

2. Can construct a path of any packet correctly: With this method we can construct a path of any packet regardless of whether it is an attacking packet or not, meaning that the method does not require amount of packets to construct a path, only one packet is enough.

3. Can construct all paths of DDoS attacks correctly: Packets of different attack paths will have different link sequences and each sequence can be decoded into a different attacking path.

4. Requires less computation to traceback: Compared to probabilistic markings or hash based logging, the new method can easily construct a path of a packet provided that it can access link tables of intermediate routers.

5. Requires smaller amount of space than other loggings: This method requires about a third of amount of space required in hash based logging to store marking fields along with message digests in intermediate routers.

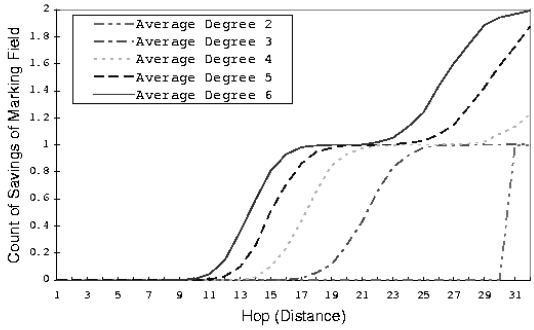
6. Requires local memory to store marking fields: It is essential for routers to have enough memory to store marking fields even though the memory requirement is less than that of other method. The requirement increases in proportion to the period of keeping of marking fields.

7. Adds overhead of marking to routers: It is a load for routers to maintain a link table and that the table must be correct and well optimized, and it is an issue how to enforce or impose an obligation of keeping and managing the table to all routers.

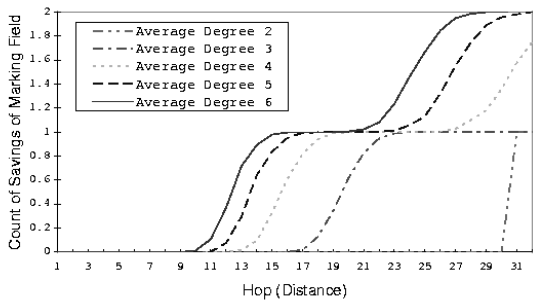
8. Vulnerable to 1-bit error: The new idea requires all routers a packet pass through to mark, and if one of internal router does not mark or if there is at least a 1-bit error in the marking field of a packet then the traceback of the packet will fail. It is a characteristic of a variable length codes like Huffman codes that if one of bit is inverted or missing by error then correct decoding (expanding) of the encoded (compressed) bit string is impossible from the codeword including the bit error.

4. Practical Issues for the New Scheme

1. Management of link tables: Each router must maintain a correct link table and provides victims with the table when asked for access. Link tables should be optimized as well as possible to reflect the correct distribution of packets coming into the routers from its adjacent routers. It will be an issue how to enforce or impose an obligation of keeping and managing tables to all routers. We may authorize a certain system or an organization to collect all the link tables and manage them: checking correctness and controlling access to the tables. This collection of tables will be a whole router-level Internet map. If the configuration of links of a router changes then the link table should be updated promptly and previous link table must be preserved for some period of time such that a victim can access the previous table and decode a link code marked by the router. Each previous table must be annotated with starting and ending dates and times.

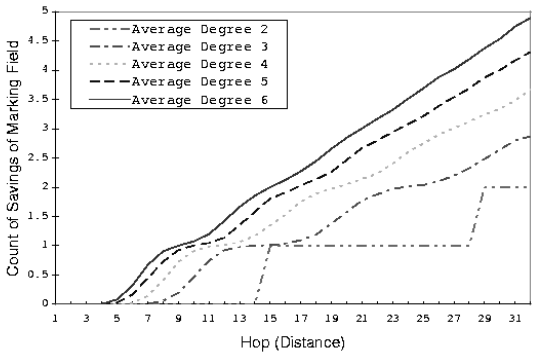


(a) Unequal Distribution of Packets

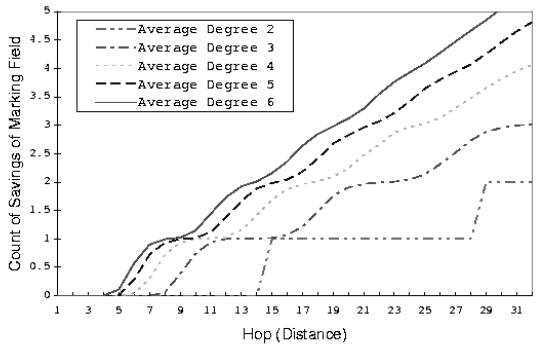


(b) Equal Distribution of Packets

Figure 9. Average count of savings of 32-bit marking field of a packet during its travel: (a) With unequal distribution of packets among the links: Links were given different frequencies in the range of 1∼255 randomly and one percentage of packets transformed; (b) With equal distribution of packets among the links: Every link was given same frequency of packets with 255 and one percentage of packets transformed.

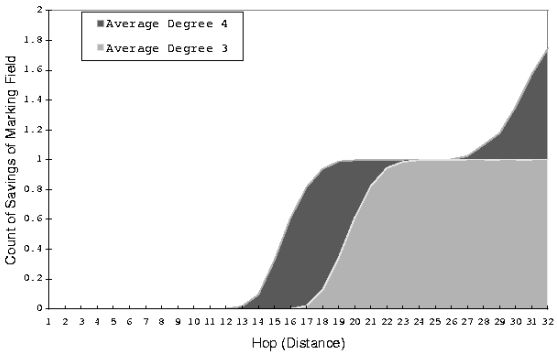


(a) Unequal Distribution of Packets

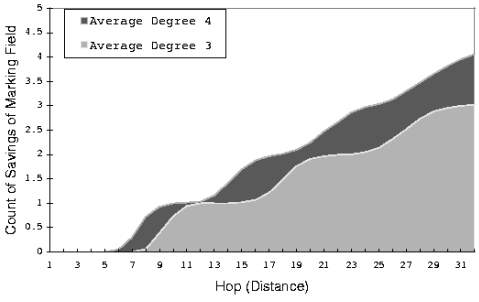


(b) Equal Distribution of Packets

Figure 10. Average count of savings of 16-bit marking field of a packet during its travel: (a) With unequal distribution of packets among the links: Links were given different frequencies in the range of 1∼255 randomly and one percentage of packets transformed; (b) With equal distribution of packets among the links: Every link was given same frequency of packets with 255 and one percentage of packets transformed.



(a) 32-Bit Marking Field of a Packet



(b) 16-Bit Marking Field of a Packet

Figure 11. Average count of savings of the marking field of a packet with average degrees 3 and 4, and with one percentage of packet transformed and equal packet distribution among links: (a) for 32-bit marking field; (b) for 16-bit marking field.

2. Local memories of routers: Many current routers are not equipped with a hard disk and do not have enough main memory to store marking field for a certain period of time, that is to say one minute or so.

3. The ability to know on what link packets arrive: Every router can satisfy the assumption that is capable to know from which link packets arrive on. But the function to figure out from which neighbor sent a packet to it must be done automatically upon arrival of the packet, and the packet must be tagged with the link information until it is marked by the embedded program of a router.

4. Packet transformations: A packet can be transformed more than once during its journey by internal routers. To trace a packet that was transformed from another packet back up to routers before the transformation, the marking field of old packet must be copied into new packets. Some protocols of transformation like ICMP copy the contents of the IP header of previous packet into the data field of new packets.

4.1. Conclusion and Future Work

It is difficult to trace a packet back to its source with current IP version 4.0. From the beginning the Internet was not designed and implemented with tracebacks in mind, needless to say when people started building the Internet they had not imagined situations where tracebacks are needed.

Current IP header is not appropriate for marking, using either the Identification field or the Option field of IP header has its own limitation. Therefore this study does not suggest specific fields in IP header to use for marking, but suggests and analyzes a new marking technique with two different sizes of marking field,16-bits and 32-bits.

The new idea proposed in this study requires routers to have enough memory space regardless of whether it is a hard disk or a main memory to store marking fields for a certain period of time in accordance with the amount of traffic. However most of routers have been doing their jobs without a local hard disk or even with a small main memory, so they have to be equipped with a secondary memory to store marking fields. In hash based logging [7] they attach a Data Generation Agent to a router to store information of packets the router forwards.

The scheme presented in this study is to mark every packet at routers so that every packet will have information about intermediate routers between source (attacker) and destination (victim). It may be worth thinking over whether it is necessary to generate IP traceback information for all packets regardless of whether it is a marking or a logging. In probabilistic markings routers do not mark all packets but sample packets to mark because packets cannot keep all the router's IP information due to the limited space of the marking field in IP header. To lessen the marking load of routers and to decrease the size of the marking field in IP header probabilistic marking can be applied to the new scheme.

Moreover it is necessary to deliberately select fields of IP header to use as an input of the message digest so that routers do not need to store MD(oldP):MD(newP)(*sf*, *ls*) in case that a transformation does not change the fields of IP header that are used as inputs of the message digest. For instance the Identification field is used in fragmentation transformation, and if routers do not use the Identification field and data portion as inputs when calculating the digest to store a marking field because of lacking of space in the link sequence field (*ls*), routers do not need to store MD(oldP):MD(newP)(*sf*, *ls*) in case of fragmentations.

References

[1] S. Bellovin, M. Leech, and T. Taylor. The ICMP traceback message. RFC 2026, Internet Engineering Task Force, March 2000, expired April 2002.

[2] H. Burch and W. Cheswick. Tracing anonymous packets to their approximate source. In *Proceedings of the USENIX Large Installation Systems Administration Conference*, pages 319–327. USENIX, December 2000.

[3] K. C. Claffy and D. McRobb. Measurement and visualization of Internet connectivity and performance. URL: http://www.caida.org/tools/skitter.

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

[5] R. L. Rivest. The MD5 Message-Digest Algorithm. RFC 1321, Internet Activities Board, Internet Privacy Task Force. April 1992.

[6] S. Savage, D. Wetherall, A. Karlin, and T. Anderson. Practical network support for IP traceback. In *Proceedings of the 2000 ACM Special Interest Group Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM 2000)*, pages 295–306. Association for Computing Machinery, August 2000.

[7] A. C. Snoeren, C. Partridge, L. A. Sanchez, C. E. Jones, F. Tchakountio, S. T. Kent, and W. T. Strayer. Hash-based IP traceback. In *Proceedings of the 2001 ACM Special Interest Group Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM 2001)*, pages 3–14. Association for Computing Machinery, August 2001.

[8] D. X. Song and A. Perrig. Advanced and authenticated marking schemes for IP traceback. In *Proceedings of the 20th Annual Conference of IEEE Communications and Computer Societies (INFOCOM 2001)*, pages 878–886. IEEE Communications and Computer Societies, April 2001.

[9] I. Stoica and H. Zhang. Providing guaranteed services without per flow management. In *Proceedings of the 1999 ACM Special Interest Group Conference on Data Communication (SIGCOMM 1999)*, pages 81–94. Association for Computing Machinery, August 1999.

[10] W. Theilmann and K. Rothermel. Dynamic distance maps of the Internet. In *In Proceedings of the 19th Annual Conference of IEEE Communications and Computer Societies (INFOCOM 2000)*, pages 275–285. IEEE Communications and Computer Societies, March 2000.