ICMP Traceback with Cumulative Path, An Efficient Solution for IP Traceback

Henry C.J. Lee, Vrizlynn L.L. Thing, Yi Xu, and Miao Ma

Institute for Infocomm Research,
21 Heng Mui Keng Terrace, Singapore 119613
{hlee, vriz, yxu, miaom}@i2r.a-star.edu.sg

Abstract. DoS/DDoS attacks constitute one of the major classes of security threats in the Internet today. The attackers usually use IP spoofing to conceal their real location. The current Internet protocols and infrastructure do not provide intrinsic support to traceback the real attack sources. The objective of IP Traceback is to determine the real attack sources, as well as the full path taken by the attack packets. Different traceback methods have been proposed, such as IP logging, IP marking and IETF ICMP Traceback (ITrace). In this paper, we propose an enhancement to the ICMP Traceback approach, called ICMP Traceback with Cumulative Path (ITrace-CP). The enhancement consists in encoding the entire attack path information in the ICMP Traceback message. Analytical and simulation studies have been performed to evaluate the performance improvements. We demonstrated that our enhanced solution provides faster construction of the attack graph, with only marginal increase in computation, storage and bandwidth.

1 Introduction

The Internet is increasingly becoming the pervasive means of communications for all media. At the same time, this has also generated many security problems. In this paper, we look at the issue relating to Denial-of- Service (DoS) [1] and Distributed DoS (DDoS) attacks. In a DoS attack, typically huge quantity of malicious packets are generated and directed towards one or many victims. DDoS is a variation of DoS in that the attacker launches an attack not from one single source, but from several sources that the attacker has already penetrated. As a result, legitimate data traffics are disrupted, servers are compromised, and services are denied to the legitimate users. In such attacks scenario, attackers usually send packets with spoofed IP addresses so as to hide their true network location from the victims and the network infrastructure.

The IP [2] packet contains two addresses: source and destination. The destination address is used by the routing architecture to deliver the packet. The IP network routing infrastructure does not verify the authenticity of the source address carried in IP packets. The source address is used by the destination host to determine the source for message reply. In general, no entity is responsible for the correctness of source address. The scenario is the same as sending a letter using the postal service; the postal service does not care about the correctness or authenticity of the source address, it merely makes sure that the letter is delivered to the correct destination. Consequently, the design of the IP protocol and forwarding mechanism makes it difficult to identify the real origin of a

packet. This characteristic of the Internet is exploited by some malicious users to hide their source and identify. Some mechanisms such as "Ingress filtering" [3] have been proposed to enforce the validity of source IP address originating from a stub network. However, such mechanisms are quite limited as they can only be used in edge networks and the universal enforcement of which is difficult.

The objective of IP Traceback is to determine the actual origin of IP packets, so as to institute accountability. Several approaches have been proposed to address this issue. IP Logging has been proposed in [5], where the intermediate routers log the passage of all IP packets. The log information is then stored in the routers. The combination of this log from the various routers can then be used, if necessary, to trace the path taken by any IP packet. IP marking has been proposed in [6], where the intermediate routers add router's information derived from its address into the IP packets (e.g. in the "Identification" field) with certain probability. The victim of an attack can thus examine this information found in the attack packets so as to construct the path taken, which eventually leads to the true attack origin. ICMP Traceback has been proposed in [4], where intermediate routers generate an ICMP Traceback message probabilistically with every IP packet, and send the ICMP Traceback message to the same destination as the IP packet. The victim of an attack can thus use the received ICMP Traceback messages to construct the attack path.

This paper is organized as follows: section 1 gives an introduction to the paper. Section 2 describes the various current approaches for IP Traceback, namely IP logging, IP marking and ICMP Traceback. Section 3 describes our proposed enhancement to the ICMP Traceback messages. Section 4 describes the comparisons between our approach and the ICMP Traceback. Section 5 describes simulation studies and results. Section 6 concludes the paper.

2 Background

The challenge of IP Traceback is to find an efficient and scalable way to track the source of an arbitrary IP packet. The source can be an Ingress point to the traceback-enabled network, the actual host or network of origin, or compromised routers within the enabled network. It depends on the extent to which the traceback framework is deployed. In an attack, it is possible that some routers may be subverted, hence there is a need to construct the attack path, which comprises the routers traversed by packets from the "source" to the victim. In the case of a DDoS attack where packets come from potentially many secondary sources, there'll be many attack paths. The attack graph is defined as the set of attack paths. The objective of the IP traceback mechanism is to construct the attack graph with the constraint that it should minimize the time that routers spend on tracking and minimize the storage used to keep the tracking information. Lastly, the solution should not adversely impact the privacy of legitimate users.

There are two main approaches to perform traceback: infrastructure scheme and end host scheme. In the first approach, infrastructure scheme, the network is responsible for maintaining the traceback state information necessary for the victim and the network to construct the attack graph. IP logging scheme belongs to this category. In the end host

scheme, the end hosts, which are the potential victims, maintain the traceback state information. IP marking and ICMP Traceback belong to this category.

2.1 IP Logging

In this approach, the network routers log the passage of all IP packets. The key challenge here lies in the potential huge amount of information storage requirement. For example, if a router were to log all the packets in its entirety, each OC-192 link at 1.25 GB/s at the router requires 75 GB of storage for a 1-minute query buffer. The storage requirement quickly becomes prohibitive as the number of router links increases. One solution, SPIE (Source Path Isolation Engine) [5], has been proposed for IP version 4. The mechanism is designed to identify the true source of a particular IP packet given a copy of the packet to be traced and an approximate time of receipt. In order to take care of packets transformation as they are routed from source to destination, the mechanism identified the invariant portions of the 20-byte IPv4 header. The fields that are susceptible to changes include: TOS (Type of Service), TTL (Time to Live), Checksum and Options field. The logging is based on the invariant portion of the IP header and the first 8 bytes of payload. Based on statistics collected, the 28-byte prefix described above results in a rate of collision of approximately 0.00092% in a WAN environment and 0.139% in a LAN environment. To further reduce the storage requirement, instead of storing the entire 28-byte prefix, hashing is performed on it, followed by a Bloom filter processing. The scheme reduces memory storage requirement in the router to 0.5% of link bandwidth per unit time. It also maintains privacy, prevents eavesdropping of legitimate traffic stream.

2.2 IP Marking

The intermediate routers marks the IP packets with additional information so that the victim can use them to determine the attack path. Approaches proposed include node append, node sampling and edge sampling [6]. The node append mechanism is similar to the IP Record Route Option [2], in that the addresses of successive routers traversed by an IP packets are appended to the packets. The victim can thus easily traceback the source of such attack packets. However, this method introduces very high overhead in terms of router processing and packet space. The node sampling approach reduces such overhead by the probabilistic marking of IP packets. The edge sampling approach, as its name imply, marks an edge of the network topology, traversed by the IP packets, instead of just the node. Most algorithms proposed to put the marking information in the Identification field of the IP header. This type of mechanism has an inherent disadvantage in that it affects the format of IP packets. The necessary changes in the IP packet format depends on the algorithm used. The standardization of format for IP marking becomes an issue.

2.3 ICMP Traceback (ITrace)

In the ICMP Traceback mechanism, a new ICMP message type, ICMP Traceback (ITrace), is defined to carry information on routes that an IP packet has taken. As the

IP Marking requires to overload some fields in the IP header, which raises backward protocol compatibility problem, the ICMP Traceback utilizes out-band messaging to achieve the packet tracing purpose.

As an IP packet passes through a router, ICMP Traceback message (ITrace) [4] is generated with a low probability of about 1/20000. Assuming that the average diameter of the Internet is 20 hops, this probability value translates to a net increase in traffic of about 0.1%. This ITrace message is then sent randomly, with equal probability, to the destination or to the origin of the IP packet. In the event of a DoS/DDoS attack, the destination node can then use it to traceback the attack path. On the other hand, the ITrace message provides information for the origin to decipher reflector attacks.

When a router generates an ITrace message, it may generate one of the followings: back link, forward link, or both. Each link element defines a link along which the packet will or has travelled through. The link element comprises of 3 components: the interface name at the generating router, source and destination IP address of the link, and finally a link-level association string that is used to tie together Traceback messages emitted by adjacent routers. On LANs, this string is constructed by concatenating the source and destination MAC addresses of the two interfaces. Finally, each ITrace message contains a variable length RouterID field.

3 ICMP Traceback with Cumulative Path (ITrace-CP)

The current IETF's ITrace message proposal allows routers to generate ITrace messages for the source and destination of IP packets. In the context of DoS / DDoS attack, the victims can make use of the received ITrace messages to construct the attack paths and ultimately identify the attackers. Since each ITrace message only carry one or two links of the entire path, the victim will have to re-construct the various attack paths from the various segments. The task will be especially difficult in the event of a DDoS attack.

This attack graph construction procedure will be facilitated if the ITrace messages are made to carry the entire path information from the routers nearest to the attackers all the way to the victim. With this approach, the victim will only need to identify the attack packets in order to establish the entire attack path or attack graph. In the following section, we will propose and analyze various solutions to encode the path traversed by the attack packets into the ITrace message. Our enhancement only applies to the ITrace messages sent to the destination address.

A simple approach will be to generate ITrace message with the IP Record Route option, so that the subsequent routers will append their addresses to the ITrace message. However, this approach has some drawbacks. Firstly, the ITrace message may not take the same path as the corresponding attack packet to the victim. In this case, the ITrace message will record the wrong route information. Furthermore, the record route option is limited to 9 routers. This is because the header length of IP header is a 4-bit field, limiting the entire IP header to 60 bytes. Since the fixed size of the IP header is 20 bytes, and the RR option uses 3 bytes for overhead, this leaves 37 bytes for the list, allowing up to 9 IP addresses. As such, it may be sufficient in the early days of ARPANET but it is of limited use given the extent of Internet today. Last but not least, most hosts/routers ignore or discard this option.

Our approach constructs ITrace message in a different way. Instead of encoding the path information in the IP packet header record route option, we use an enhanced ITrace message, called the ICMP Traceback with Cumulative Path (ITrace-CP) to store the path information. When a router receives an IP packet, it generates an ITrace-CP message with a certain probability. However, instead of sending the ITrace-CP message to the destination address of the IP packet, it is sent to the next hop router. This "next hop" should be as far as possible the same as the next hop for the corresponding IP packet. The ITrace-CP packet will also contain as much of the IP packet as possible, including the final destination address. In addition, the ITrace-CP message should be sent after the corresponding IP packet.

At the next hop router, the router will process the ITrace-CP message as follows. There are two possibilities:

1. If the ITrace-CP packet is forwarded to the same router as the corresponding IP packet, then the router will generate a new ITrace-CP message and append its own IP address. The new ITrace-CP then forwarded to the next hop router of the corresponding IP packet.
2. Otherwise, the router that processes the ITrace-CP will generate a new ITrace-CP message for the final destination, without making any changes to the payload.

As a result, full or partial path information is stored in the ITrace-CP message when it reaches its destination.

The problem now is how to identify corresponding IP and ITrace-CP messages. The simplest way is for the routers to store the IP packets for a short duration and compare them to the received ITrace-CP messages. However, if we take the example of a Router with 16 OC-192 links at 1.25 GB per second, this would translate to a storage requirement of 2 GB for 100 ms seconds of buffer. We propose three schemes to reduce the storage requirement for matching corresponding IP packets and ITrace-CP messages. In the subsequent analysis, we use 100 ms as the upper bound for the inter-arrival time between an IP packet and its corresponding ITrace-CP message, if any. We will also use the same router configuration as above and assume that the average IP packet size is 256 bytes.

3.1 Scheme 1: Basic Packet Identification (BPI)

Typically, the source of an IP packet sets the identification field to a value that must be unique for that source-destination pair and protocol for the time the packet will be active in the Internet. Hence, the value of the "Identification" field can be used, together with the source and destination addresses and the protocol number to uniquely identify an IP packet in a short time window. In order to take care of possible fragmentation, the flags and fragment offset field of the IP packet can also be included for packet identification. In this way, if fragmentation did occur, the victims will be able to construct paths taken by the fragmented packets, and paths taken by the non-fragmented packets and link them together through the Identification field mentioned earlier that uniquely identifies the packet stream.

3.2 Scheme 2: Hash-based Packet Identification

In this approach, instead of storing the BPI of a packet, the routers determine the hash of the BPI to reduce the storage requirement. The Hash function used must satisfy the following requirements. Firstly, the function must distribute a highly correlated set of input values (i.e. the BPI information) as uniformly as possible over the hash function's output space. Secondly, the hash function should be computationally efficient so as to minimize the computation overload. A hash of 16 bits of hash will result in a collision rate of less than 0.002%. For our router configuration, 100 ms of hash will require a buffer size of less than 16 MB.

3.3 Scheme 3: Hash-based Packet Identification with indicator bit

In addition to scheme 2, this approach set a bit in the IP packets to indicate that an ITrace-CP message has been generated for a specific IP packet. One possibility is to use the first bit of the 3-bit flags field, which is not used currently. This reduces the need to keep those packets, where the bit is not set, or their hashes in downstream routers, hence reducing significantly the storage and processing requirement. The disadvantage is that some changes to the IP packet processing is required.

Let $ITRACE\_CP\_DONE$ be the bit in IP packet that indicate if an ITrace-CP message has been generated for it. The pseudo code of the algorithm at each router is shown as follows. For each IP packet received at a router:

```
If (ITRACE_CP_DONE is set) then {
    Calculate the packet's hash and store it in the buffer
    (this hash will be kept for 100 ms)
    forward the IP packet to R (next hop router)
    (record R in the buffer)
} else {
    generate an ITrace-CP message with a probability p
    set the ITRACE_CP_DONE bit in the IP packet
    forward the IP packet to R (next hop router)
    send the new ITrace-CP message to R
}
```

If an ITrace-CP packet has not been received within 100 ms for an IP packet that has been stored in the buffer, it is possible that the ICMP packet is routed differently as the IP packet or that the inter-arrival time between the IP packet and its corresponding ITrace-CP is higher than 100 ms. In all cases, a new ITrace-CP message will be generated.

Using the bit information, assuming that the max number of hops traversed by the packets is 20, the hash storage requirement is reduced to 16 KB (16 MB * 20 /20000). However, this scheme is vulnerable to another form of exploitation. If the attacker artificially set this bit in all attack packets, then ITrace messages will be generated at the first router for all the packets and all the subsequent routers will construct the cumulative path. Although this mechanism worsens the DoS / DDoS attacks by doubling the attack traffic, the victims will be able to detect the attack, construct the attack graphs and determine the true sources almost instantaneously.

4 Comparison of ITrace-CP with ITrace

We compare the ITrace-SP and Itrace mechanisms in terms of computation overhead, bandwidth and storage overheads. Firstly, in terms of bandwidth, the overhead is minimal as the additional information carried is the IP addresses of the intermediate routers. Assuming that the average path length is 10, the additional bytes carried is only 40 bytes per ITrace-CP message. Given the message infrequency, the overhead is still minimal even if more router information needs to be included. In terms of storage, each router only need to cater less than 16 MB (for ITrace-CP scheme 2) for an inter-arrival time of 100 ms between an IP packet and its corresponding ITrace-CP message. If a bit is used to mark the IP packet (for ITrace-CP scheme 3), the storage overhead will only be 16 KB. In terms of computation, the hash function will introduce minimum overheads. In summary, the additional overheads of ITrace-CP are relatively minimal compared to the ITrace scheme, similarly as compared to other IP Traceback proposals such as IP logging and IP marking.

However, the ITrace-CP scheme will perform much better than the ITrace scheme in its ability to traceback the attack source faster, because more information on the attack path is carried inside the ITrace-CP message. We will now look at a network scenario where the attack path comprises L routers. Let p be the probability of generating an ITrace or ITrace-CP message. We determine their respective performances in attack path construction. The performance metric is expressed as the probability that the full attack path can be constructed with a given number of attack IP packets (N).

For the ITrace-CP scheme, the entire attack path can be constructed by the victim when the router furthest from the victim generates at least one ITrace-CP. Hence, the probability P_E that the full path can be constructed after the victim has received N IP packets is given by (1). Note that P_E is independent of the path length L.

$$P_E = 1 - (1 - p)^N \tag{1}$$

For the basic ITrace scheme, each ITrace message can contain either the forward link, back link or both links. For simplicity, we assume that ITrace messages with either the forward or back link enable the victim to discover two routers addresses on the attack path, whereas the ITrace messages with both links enable the victim to discover three. P_{B1} and P_{B2} denote the full path construction probabilities for ITrace (forward or back link) and ITrace (both links) respectively.

$$P_{B1} = (P_E)^{\frac{L}{2}} \tag{2}$$

$$P_{B2} = (P_E)^{\frac{L}{3}} \tag{3}$$

Figure 1 and 2 plot the Probability of Path Construction as a function of the number of IP packets received. With 20,000 attack packets, the ITrace-CP has a 63% chance of constructing the entire path, versus the ITrace (Forward and Back links) which has chances of 47%, 22% and 10% for 5, 15 and 20 hop attack paths respectively. With 50,000 packets, the probabilities are 92% for ITrace-CP, 87% for 5-hop ITrace, 65% for 15-hop ITrace, and 57% for 20-hop ITrace. The figures show clearly that the ITrace-CP

mechanism requires much less packets to construct the entire attack path as compared to the ITrace.

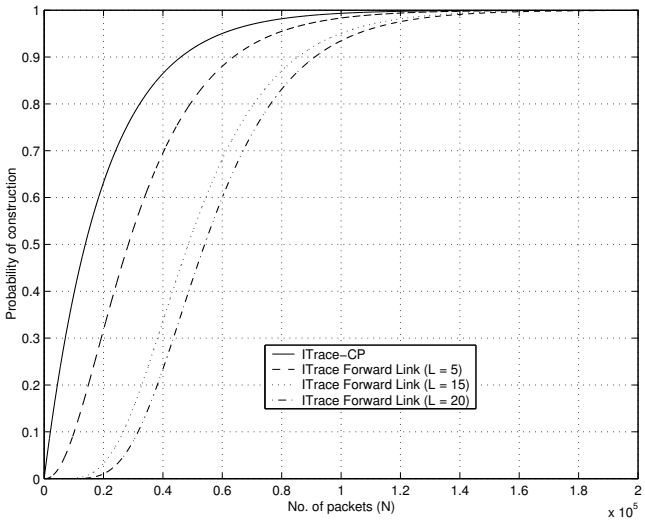

Fig. 1. Performance Comparison between ITrace-CP with ITrace (Forward Link)

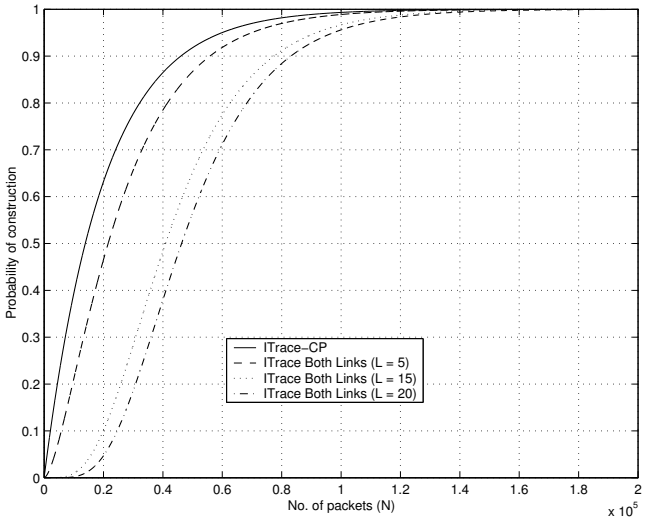

Fig. 2. Performance Comparison between ITrace-CP with ITrace (Forward & Back Link)

5 Simulation studies

Our simulation studies evaluate the effectiveness of the ITrace-CP and ITrace (Forward or back link, both links) mechanism, in terms of time taken to establish the attack graphs in the event of DoS and DDoS attacks. We use the ns-2 network simulation software to model both traceback mechanisms. We constructed the agents for the attacker, the router and the victim.

5.1 Simulations model

As discussed earlier, we assume that, for the ITrace forward or back link, 2 routers would be detected per each ITrace message, and 3 routers in the case of both links encoding option. For the ITrace-CP scheme, the number of routers detected is the number of routers addresses encoded in the message.

Since we are comparing in terms of time taken to establish the attack graphs, evaluation based on false positive was not considered. Therefore, only attack traffic was generated and hash collision was not simulated. In this simulation, each router would generate ITrace or ITrace-CP messages with a probability of 1/20,000 (determined by a random number generator) on the attack traffic they received. When the victim received these messages, it would discover the intermediate routers of the attack graphs. The time taken to detect various numbers of routers on the attack graphs was recorded.

5.2 Network scenario

We performed simulation studies on the two schemes using the linear network topology, for attackers situated 5, 15, and 20 hops away from the victim. The tree topology was not simulated as in the case of multiple attackers at the leaves of the tree, they would have been treated as independent attack paths. This would be similar to simulating the linear topology. For example, in Figure 3, if an attacker sends packets through routers 3 and 2 to the victim node at 1, and that ITrace messages are generated by router 2 based on these packets, it should not be treated as if the router 2 is detected for attack path by another attacker sending packets through router 4 and 2.

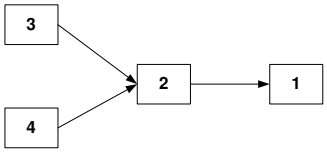

Fig. 3. Tree Topology

In all simulation scenarios, the effective attack traffic arriving at the victim is 1 Mbits/s. However, as we are interested in the relative performances of the two schemes, this number is only indicative.

5.3 Results

The average times (for 30 runs) for the construction of various hops of the attack path for the ITrace (forward link and both links) and ITrace-CP were obtained. The graphs were plotted and shown in Figure 4 to 6. In all the figures, the x-axis represents the number of hops of the attack path discovered while the y-axis represents the time taken in seconds. The 3 figures corresponds to attack paths of length 5, 15 and 20 respectively.

In Figure 4, the performance of the ITrace-CP scheme improved over the ITrace for the forward link option but not the both links option. The average times taken to detect the full path were 27 secs, 19 secs, and 23 secs for the ITrace forward link option, ITrace both links option, and ITrace-CP scheme respectively.

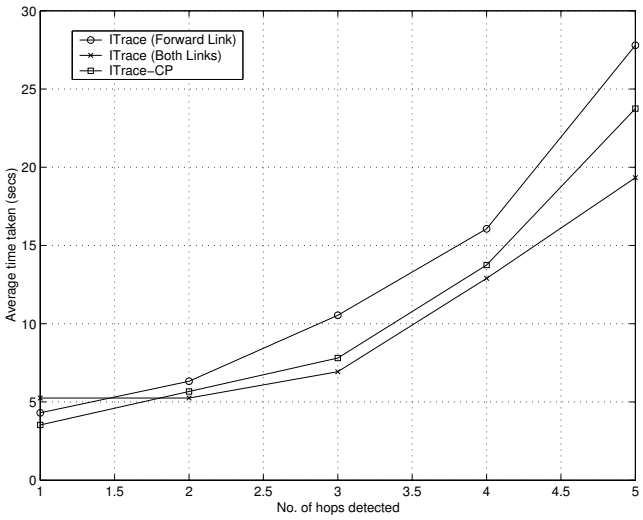

Fig. 4. Average time taken to detect various numbers of hops (5-hop attack path)

In Figure 5, the performance of the ITrace-CP scheme becomes better than the ITrace schemes starting from the detection of 4 hops of the attack path. The window from 10 to 13 hops showed significant improvement by the ITrace-CP scheme; with peak improvement at the 13th hop detection. However, the high average time taken to detect the 14th and 15th hop resulted in the drop in improvement. The average times taken to detect the full path were 40 sec, 27 sec, and 22 sec for the ITrace forward link option, ITrace both links option, and ITrace-CP respectively.

In Figure 6, the performance of the ITrace-CP scheme become better than the other 2 schemes starting from the detection of 4 hops of the attack path. The low gradient of the curve for the ITrace-CP scheme from detection of the 1st to 14th hop indicated that about 14 hops of the attack path could be detected within the same average time taken. The window from 10 to 18 hops showed significant improvement by the ITrace-CP

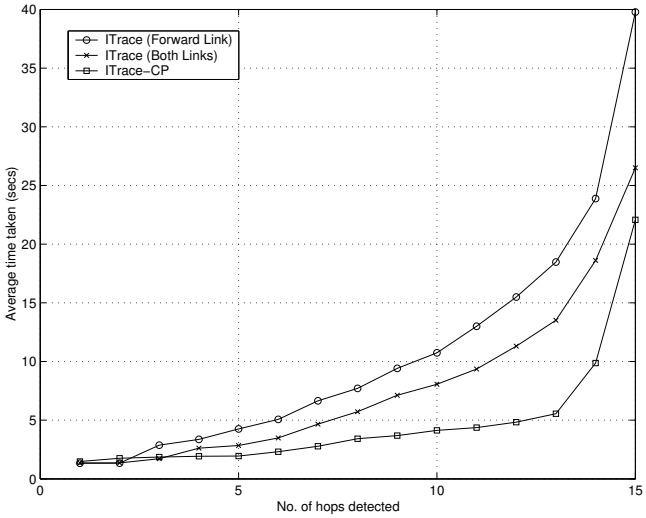

Fig. 5. Average time taken to detect various numbers of hops (15-hop attack path)

scheme; with peak improvement at the 18th hop detection. However, the high average time taken to detect the 19th and 20th hop resulted in the drop in improvement. The average time taken to detect the full path was 39 sec, 24 sec, and 18 sec for the ITrace forward link option, ITrace both links option, and ITrace-CP respectively.

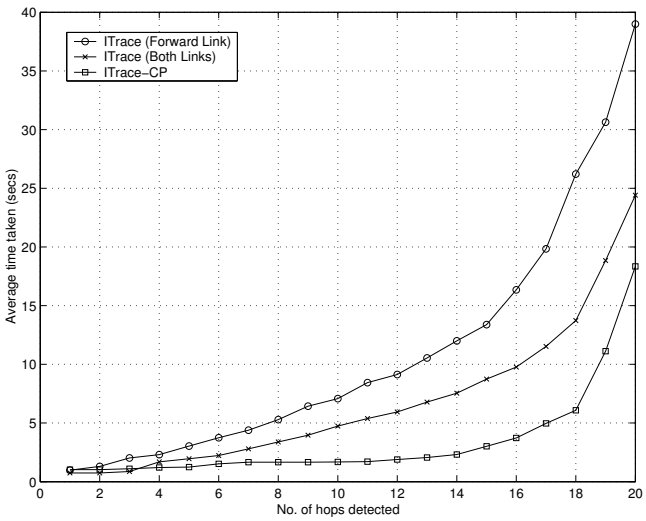

Fig. 6. Average time taken to detect various numbers of hops (20-hop attack path)

6 Conclusion

The objective of IP Traceback is to determine the true source of DoS/DDoS attacks. This paper first gives an overview of the respective approaches for IP Traceback. We then proposed an enhanced ICMP Traceback scheme, called ITrace-CP (ICMP Traceback with Cumulative Path) that encodes cumulative attack path information. We described the ITrace-CP protocol and the mechanism for constructing ITrace-CP messages so that it contains the addresses of all the routers on the attack path. As part of the ITrace-CP protocol, we proposed three schemes for the routers to match corresponding IP and ITrace-CP messages. We have carried out qualitative comparison of the ITrace-CP scheme with the ITrace scheme in terms of storage, bandwidth and computational requirements. We deduced that the ITrace-CP introduces marginal overhead in terms of storage and bandwidth and acceptable computational overhead. Analytical studies were done to compare the performances in terms of the probability of attack path construction as a function of number of attack packets and attack path length. We found that the performance of the ITrace-CP is independent of the attack length and that the probability of path construction of ITrace-CP is significantly higher than that of the ITrace, for all hop lengths. Simulation studies have also been conducted to further evaluate their relative effectiveness in constructing the DoS and DDoS attack paths. Our simulation showed that the ITrace-CP mechanism performs better than the ITrace mechanism and takes significantly less time to construct the entire attack path in longer attack paths.

7 Future Work

In the ICMP Traceback proposal, it is recommended that ITrace messages be generated with a probability of 1/20,000 so as to limit the increase in data traffic to less than 0.1%. However, in ITrace-CP scheme, given that path information are generated, it is more logical to generate ICMP messages nearer to the attackers, or in other words further from the victim. We will investigate how the probability can be determined to further improve the performance. Also, we have assumed an upper-bound of 100 ms for the packets inter-arrival time. A more rigorous study of this will enable a more accurate determination of the buffer allocation as well as optimal performance of ITrace-CP.

References

1. K.J. Houle, G.M. Weaver, “Trends in Denial of Service Attack Technology”, CERT Coordination Center, Oct 2001. http://www.cert.org/archive/pdf/DoS_trends.pdf
2. J. Postel, “Internet Protocol”, Request for Comments 0791, Internet Engineering Task Force, 1981.
3. P. Ferguson, D. Senie, “Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing”, Request for Comments 2827, Internet Engineering Task Force, May 2000.
4. Steve Bellovin et al, “ICMP Traceback messages”, IETF Internet Draft “draft-ietf-itrace-04.txt”, Feb 2003. Work in progress.
5. Alex C. Snoeren et al, “Hash-Based IP Traceback”, ACM SIGCOMM 2001, August 2001.
6. Stefan Savage et al, “Practical network support for IP traceback”, ACM SIGCOMM 2000.