

Copyright © 2007 Institute of Electrical and Electronics Engineers, Inc. All rights reserved. Personal use of this material, including one hard copy reproduction, is permitted. Permission to reprint, republish and/or distribute this material in whole or in part for any other purposes must be obtained from the IEEE. For information on obtaining permission, send an e-mail message to [stds-ipr@ieee.org](mailto:stds-ipr@ieee.org). By choosing to view this document, you agree to all provisions of the copyright laws protecting it. Individual documents posted on this site may carry slightly different copyright restrictions. For specific document information, check the copyright notice at the beginning of each document.

## On the Effectiveness of Flexible Deterministic Packet Marking for DDoS Defense

Yang Xiang  
School of Management and  
Information Systems  
Central Queensland University  
Rockhampton, Australia  
y.xiang@cqu.edu.au

Wanlei Zhou  
School of Engineering and  
Information Technology  
Deakin University  
Melbourne, Australia  
wanlei@deakin.edu.au

Zhongwen Li and Qun Zeng  
Information Science and  
Technology College  
Xiamen University  
Xiamen, China  
lizw@xmu.edu.cn

### Abstract

*IP traceback is one of the defense mechanisms for Distributed Denial of Service (DDoS) attacks. However, most traceback schemes consume extensive resources such as CPU, memory, disk storage and bandwidth and require a large amount of IP packets to reconstruct sources, which makes them impractical and ineffective. In this paper, we present a new flexible IP traceback scheme called Flexible Deterministic Packet Marking (FDPM). The flexibilities of FDPM are in two ways, one is that it can adjust the length of marking field according to the network protocols deployed, thus it can work well even in an environment with different network protocols; the other is that it can adjust the marking rate according to the load of participating router, while it still can maintain the marking function. In order to verify the effectiveness of FDPM for DDoS defense in terms of marking efficiency, maximum forwarding rate, and number of packets for reconstruction, we tested FDPM by both simulation and Linux router implementation with an emphasis on the latter. The experiments demonstrate that the built-in overload prevention mechanism, flow-based marking, can isolate and mark the most possible DDoS attack packets, while keeping the load of the participating router in a reasonably low degree. The real hardware implementation confirms that this flexible capability is important when traceback mechanisms are used in a real DDoS defense scenario.*

### 1. Introduction

Many Internet attacks nowadays use IP address spoofing techniques that allow the source address in an IP header to be manipulated and falsified. Distributed

Denial of Service (DDoS) attacks, which prevent legitimate Internet users from using the desired resource [9], are one of such attacks that usually counterfeited source IP addresses to hide the identity of attackers. Therefore, the IP address fields in this case are of no use to identify the attackers.

IP traceback is the ability to trace IP packets to their origins; it provides a system with the ability to identify true sources of the IP packets without relying on the source IP address field of the IP header. Current IP traceback mechanisms [2] [8] include link testing, messaging, logging and packet marking. Unfortunately, most approaches consume extensive resources such as CPU, memory, disk storage and bandwidth and require a large amount of IP packets to reconstruct sources. Some of them are impractical and others are ineffective to find the sources of IP packets quickly, precisely and inexpensively. Among these mechanisms, packet marking schemes, which can be divided further into probabilistic packet marking (PPM) and deterministic packet marking (DPM), are relatively easy to implement, and require a modest computation load and bandwidth. A key issue of packet marking schemes is their effectiveness. Most previous research on effectiveness of packet marking schemes is based on simulation, which has limitations on real challenges such as the maximum number of sources that can be traced in a real network environment, overload problems in the participating routers, and efficiency in the reconstruction of the sources.

In this paper we propose a new scheme called Flexible Deterministic Packet Marking (FDPM) which can solve these challenges and is a practical scheme that can be applied in real implementations. The work described here is the second version of FDPM, which is improved with great flexibility of overload

\* This work was partially supported Fujian natural science grant (A0410004), NCETXMU 2004 program(0000-x07116), Xiamen University research grant (0630-E23011) and Guangdong natural science foundation (06029667).

prevention of participating routers. The work is based on the initial version of FDPM [20] and Deterministic Packet Marking (DPM) [4]. The major improvements of FDPM compared to the previous work are in the flexibilities. The flexibilities of FDPM are in two ways, one is that it can adjust the length of marking field according to the network protocols deployed; the other is that it can adjust the marking rate according to the load of participating routers. To the best of our knowledge, none of the previous work has investigated the overload problem. We are among the first to examine the overload prevention in traceback schemes. In order to verify the effectiveness of FDPM for DDoS defense, we tested FDPM by both simulation and Linux router implementation with an emphasis on the latter. The experiment results demonstrate that the built-in overload prevention mechanism, flow-based marking, can isolate and mark the most possible DDoS attack packets, while keeping the load of the participating router in a reasonably low degree. The real hardware implementation confirms this flexible capability is important when traceback mechanisms are used in a real DDoS defense scenario.

The rest of this paper is organized as follows. In section 2, a short review of the initial version of FDPM is introduced. In section 3, the overload problem of traceback mechanisms is discussed. We propose a flow-based marking scheme to solve the problem. Section 4 provides details of our experiments by real hardware implementation and analyzes the results. Section 5 discusses current related work. A comparison between FDPM and other mechanisms is also given. Finally this article closes with a conclusion in section 6.

## 2. Initial version of FDPM

Flexible Deterministic Packet Marking (FDPM) utilizes many bits in the IP header that has a flexible length. When an IP packet enters the protected network, it will be marked by the interface close to the source of the packet on an edge ingress router. The source IP addresses are stored in the marking fields. The mark will not be changed when the packet traverses the network. At any point within the network, the source IP addresses can be assembled when necessary. Here we give a short review of the initial version of FDPM. More details of it can be found in [20].

Because the maximum length of mark is 25 bits, at least 2 packets are needed to carry a 32-bit source IP address. Each packet holding the mark will be used to reconstruct the source IP address at any victim end

within the network. A segment number is also assigned to the mark, because when reconstructing the packet, the segment order of the source IP address bits must be known. After all the segments corresponding to the same ingress address have arrived to the destination, the source IP address of the packets can be reconstructed. In order to keep a track on a set of IP packets that are used for reconstruction, the identities shown the packets come from the same source must be given. A hash of the ingress address is kept in the mark, known as the digest. This digest will always remain the same for a FDPM interface from which the packets enter the network. It provides the victim end the ability to recognize which packets being analyzed are from a same source, although the digest itself cannot tell the real address. Even if the participating router is compromised by attackers (for example, some marks are spoofed), this scheme will not be affected because the packets with irrelevant digest will be discarded during the reconstruction process.

The packet processing consumes resources such as memory and computing capacity of a participating router. Therefore, it is possible for a router to be overloaded when there are a large number of arrival packets. In this work, flow-based marking is proposed to solve the overload problem. When the load of a router exceeds a threshold, the router will discern the most possible attack packets from other packets then selectively mark these packets. This will alleviate the load of the router while still obtain the marking function.

## 3. Flow-based marking for overload prevention

### 3.1. Overload problem

The possibility of the problem of overload always exists because the resources of a router are limited. All packet marking traceback schemes need the processing power and storage capacity of routers. The encoding process consumes router's resources because it needs to overwrite many bits in the IP header as it is shown in [20]. Therefore, the overload prevention is important to all packet marking traceback schemes because if the router is overloaded, the packet marking scheme can be ineffective. There are many methods to lighten the burden of a router. One is to increase the computing capability of a router, for example, to embed an extended network module (hardware). Another is to apply a flexible algorithm to reduce the load of processing of packets when the load of the router exceeds a threshold.

### 3.2. Flow-based marking

In order to prevent this overload problem, a flow-based marking scheme is proposed in this paper. The idea of flow-based marking is to mark the packets selectively according to the flow information when the router is under a high load. Therefore, it can reduce the load of router; while it still can maintain the marking function. Because one of the major applications of FDPM is DDoS defense system, the flow-based marking mainly deals with the packets in DDoS attacks. For other application, this overload prevention mechanism can be modified accordingly.

The aim of flow-based marking is to mark the most likely attack packets, then let the reconstruction end and reconstruct the source by using a minimum number of packets. This process resembles some congestion control schemes such as the Random Early Detection (RED) [7], which is to isolate the flows that have an unfair share of bandwidth and drop the packets in those flows. In FDPM, the flow-based marking also needs to isolate and mark the flows that have more bandwidth, but not to drop them.

The data structures include a dynamic flow table  $T$  and a FIFO queue  $Q$  as it is in figure 1. Each record in  $T$  stands for a flow. Here the flow means the group of packets that have some defined specific subset of identifiers and are in the  $Q$  at a certain time. In order to simplify the problem, packets are classified into different flows according to the destination IP address in the IP header. The flow records in  $T$  are hashed values of the destination IP addresses and the number of packets from this flow in the queue  $Q$ . The algorithm of flow-based marking is shown below. There are two load thresholds  $L_{max}$  and  $L_{min}$ .  $L_{max}$  is the threshold that controls the whole packet marking, which means the router will not mark any packets if its load exceeds this value. The load threshold  $L_{min}$  means if the load exceeds this value, the router can still work, but it must reduce the marking load. These thresholds can be set according to different real situations in routers.  $max\_pkts$  is a threshold to control whether to mark the packet or not. The flow-based marking algorithm is shown in figure 2.

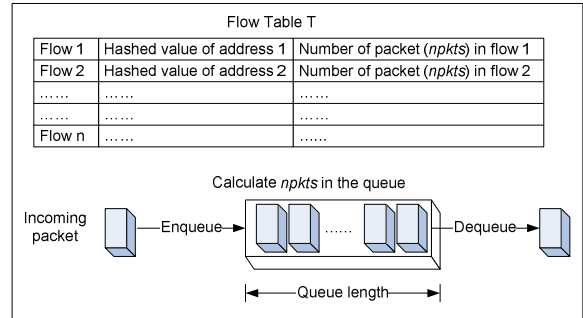


Figure 1. Dynamic flow table  $T$  and FIFO queue  $Q$

```

If (load of router R > the threshold  $L_{max}$ )
    Do not mark any packets;
    Turn on congestion control mechanisms;
Else if (load of router R > the threshold  $L_{min}$ )
    Turn on flow-based marking at R, edge interface A, in network N;
    for each attacking packet p
        check the number of packets npkts from T in the Q;
        if(npkts == 0, means no such flow in T)
            add a new entry in T, set its npkts = 1;
        else if(npkts < threshold, max_pkts)
            npkts ++;
        else
            mark the packet according to the encoding procedure;
        endif
        insert this packet into Q;
        if Q is full
            dequeue;
        endif
    endif
else
    Mark each packet at R, edge interface A, in network N;
endif

```

Figure 2. Flow-based marking algorithm

## 4. Experiments and results

### 4.1. Simulation and Linux router implementation

In order to test the effectiveness of FDPM, we conducted both simulation and Linux router implementation. We used the data generated by SSFNet [18] simulator and the embedded DDoS tools [6] in project Distributed Denial of Service Simulators at Deakin University. In the project, two DDoS tools, TFN2K and Trinoo, are adopted and integrated into SSFNet to create virtual DDoS networks to simulate the attacks. The TFN2K and Trinoo are ported from C to Java to be embedded into SSFNet. Using the DDoS simulators, we can simulate the launch of any DDoS attack with different features such as duration, protocol, attack rate, etc. Based on the initial version of FDPM, flow-based marking Java module was embedded.

Currently most of the previous work on traceback was based on simulation. It is difficult to test the real performance of the traceback scheme if only simulation is used. We used Click modular router [12] to implement FDPM on real hardware. Click is flexible and configurable router software, which is assembled from packet processing modules. FDPM encoding element, reconstruction element, flow-based marking control element, and other associated measuring elements were added to Click.

## 4.2. Marking efficiency

When the load of router exceeds a certain threshold  $L_{min}$ , the router has to reduce the marking rate in order to alleviate the load. If the packets are marked in a random manner (the possible attacking flows are not marked selectively, all packets receive the same probability to be marked), the reconstruction end will use more packets to reconstruct the sources than the flow-based marking.

Figure 3 (a) shows in SSFNet simulation, when the router uses 2 packets to carry a source IP address ( $k=2$ ), 10% of the packets are attack packets, the marking efficiency (that is measured by the number needed to reconstruct a source IP address and the marked rate of all the packets passing through the router) in flow-based marking and random marking. Figure 3 (b) shows when the router use 8 packets to carry a source IP address ( $k=8$ ), 50% of the packets are attack packets, the marking efficiency in flow-based marking and random marking. From these figures we can see the random marking can not control when to mark and which packets to mark because it randomly selects packets to mark. Therefore, both attack packets and normal packets receive the same possibility to be marked. On the other hand, by using flow-based marking, the attack packets have more chances to be marked. Thus in the reconstruction end, less number of packets are needed to reconstruct the source.

Figure 4(a) shows in Linux Click router implementation, when the router use 2 packets to carry a source IP address ( $k=2$ ), 10% of the packets are attack packets, the marking efficiency in flow-based marking and random marking. Figure 4(b) shows when the router use 8 packets to carry a source IP address ( $k=8$ ), 50% of the packets are attack packets, the marking efficiency in flow-based marking and random marking. From figure 3 and figure 4 we can see the simulation and real hardware implementation show the same trend. This clearly demonstrates the capability of the FDPM to selectively mark the most likely DDoS packets in case of high load of routers.

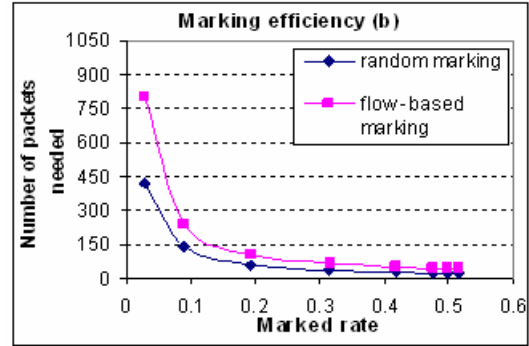
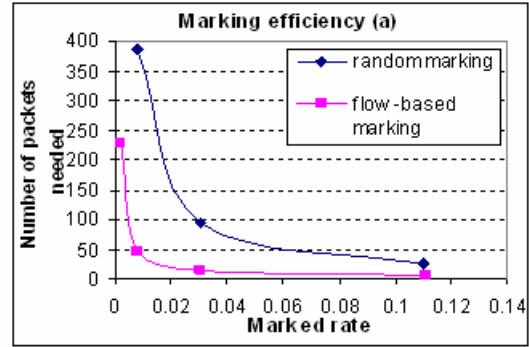


Figure 3. Marking efficiency in simulation

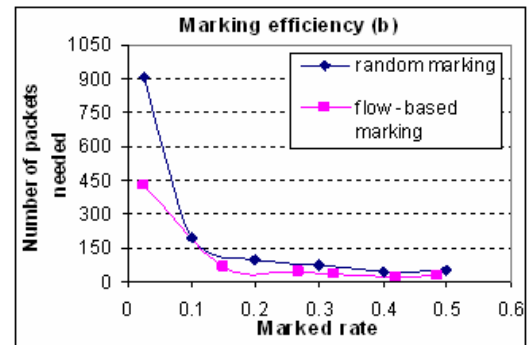
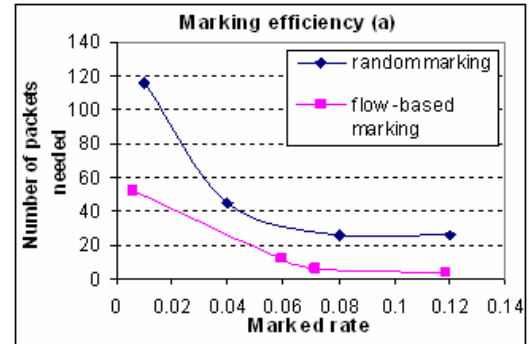


Figure 4. Marking efficiency in Linux Click router implementation

Figure 4 also shows in real case, we do not have to mark all the packets to make the traceback function work. For example, in figure 4(a), if 10% of the packets are marked, on average only about 4 packets are needed to reconstruct one source; even if only 1% of the packets are marked, on average only about 50 packets are needed. This capability of FDPM greatly relieves the router from the packet processing load.

#### 4.3. Maximum forwarding rate

This section evaluates FDPM's performance for forwarding IP packets under different conditions. The metric we use is the maximum forwarding rate. It is the rate at which a router can forward 64-byte packets over a range of input rates. In simulation, it is difficult to measure this rate. Therefore, Linux Click router implementation is used. The maximum forwarding rate can be plotted as the line in input rate and forwarding rate coordinates. Ideally, a router would forward every input packet regardless of input rate, corresponding to the line  $y=x$ . Figure 5 shows the maximum forwarding rate for Click router without any packet marking functions. This figure can be used as the baseline to compare with FDPM's maximum forwarding rate. In our experiments, the maximum forwarding rate is 69,000 packets per second. When input rate exceed this rate, the router will discard received packets due to the bottleneck of the router's CPU. The maximum forwarding rate in our work is different with that in [12] because of the Ethernet card in our configuration does not support polling functions. However, it does not affect the comparison between FDPM and this baseline. Since the performance of FDPM is hardware related, we envision a higher maximum forwarding rate can be obtained if hardware is more advanced.

A series of experiments were carried out to test the maximum forwarding rate of FDPM. Figure 6 shows when  $k=8$ ,  $length\_of\_queue=10$ ,  $max\_pkts=3$ , the curve of maximum forwarding rate of FDPM and the curve when all the packets are marked. From the figure we find the maximum forwarding rate of FDPM is about 15000 packets per second higher than the one when all the packets are marked. This demonstrates FDPM can greatly increase the forwarding rate of a traceback router. Currently most pervious work does not have this capability to prevent router's overload. Additionally, if we compare figure 5 and 6, we find the maximum forwarding rate of FDPM is about 5000 packets per second less than the baseline, which means the router sacrifices about 7% of its forwarding rate performance to fulfill its traceback function, which is a moderate overload level.

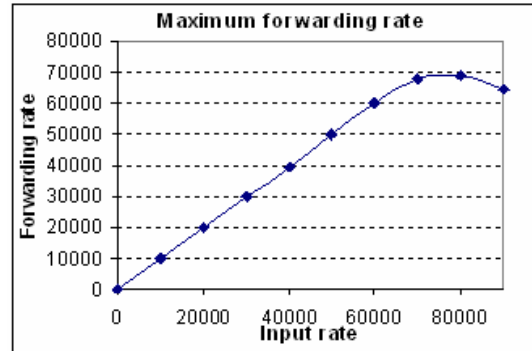


Figure 5. Maximum forwarding rate for Click router

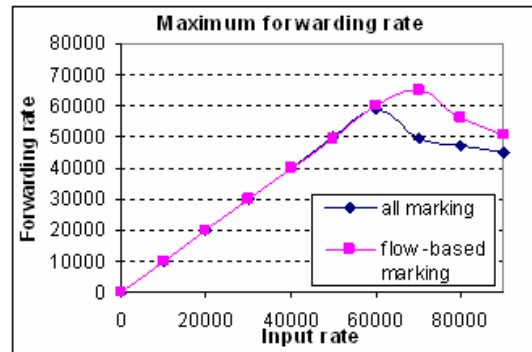


Figure 6. Maximum forwarding rate for FDPM and all marking

Table 1. The relationship between attack packet percentage and maximum forwarding rate

Attack packet percentage	Flow-based marking	All marking
1	65412	58423
0.9	66144	59104
0.8	65252	57451
0.7	64099	56482
0.6	65186	57412
0.5	64230	54132
0.4	63701	55265
0.3	63383	52102
0.2	64163	57412
0.1	67170	56325

Maximum forwarding rate is not sensitive to the attack packet percentage because FDPM can dynamically select most likely DDoS packets to be marked, when the load of router exceed the threshold  $L_{min}$ . Table 1 shows the relationship between the attack packet percentage and the maximum forwarding rate of both FDPM and all marking. Again we can see the maximum forwarding rate of DFPM is much higher than the all marking traceback scheme.

**Table 2. Relationships between marked rate, number of packets needed and percentage of attack packets in simulation**

Attack packet percentage	Marked rate	Number of packet needed
0.9	0.901	21.8
0.8	0.800	24.7
0.7	0.700	24.1
0.6	0.594	24.7
0.5	0.475	24.7
0.4	0.338	29.2
0.3	0.190	37.6
0.2	0.069	71.4
0.1	0.008	288.3

**Table 3. Relationships between marked rate, number of packets needed and percentage of attack packets in Linux implementation**

Attack packet percentage	Marked rate	Number of packet needed
0.9	0.800	20.2
0.8	0.727	25.2
0.7	0.641	25.4
0.6	0.565	25.1
0.5	0.453	24.1
0.4	0.370	22.9
0.3	0.281	26.4
0.2	0.140	56.3
0.1	0.075	74.5

### 4.3. Marked rate and number of packets for reconstruction

According to the results of experiments, the relationship between the marked rate and number of packets needed to reconstruct a source obeys a power relationship as the equation  $N = aM^{-b}$ . Where  $N$  means the number of packets needed to reconstruct a source,  $M$  means the marked rate of all packets passing through the router. Coefficients  $a$  and  $b$  can be adjusted according to many factors such as the queue length in flow-based marking ( $length\_of\_queue$ ), maximum threshold to mark the packet ( $max\_npkts$ ), the percentage of attack packets, how many packets are used to carry a source IP address (number of segments  $k$ ). For example, in the experiments, when number of segments  $k=8$ , 10% of packets are attack packets, queue length is 45, the equation can be written as  $N = 6.3871M^{-0.7496}$ .

Table 2 shows in SSFNet simulation when number of segments  $k=8$ ,  $length\_of\_queue=10$ , maximum packet threshold  $max\_npkts=3$ , the relationships between the marked rate, number of packets needed and percentage of attack packets. First, from the figure we can see fewer packets are needed at the

reconstruction end when the attack packets increase because more attacking percentages lead more packets to be marked. Second, the marked rate increases in nearly a direct ratio according to the change of the percentage of attack packets. This proves that the flow-based marking scheme can mark most of the attack packets, which indicates FDPM can effectively mark the most possible attack packets when the marking rate has to be reduced.

Table 3 shows the relationships between the marked rate, number of packets needed and percentage of attack packets in Linux implementation with the same configuration of that in table 2. From the table we can see the same trend in Linux implementation as it is in simulation.

## 5. Related Work

### 5.1. Current traceback mechanisms

According to the survey papers such as [2] [8] current IP traceback mechanisms can be classified into the following categories: link testing, messaging, logging, and packet marking. Link testing methods include input debugging [19] and controlled flooding methods [5]. The main idea of it is to start from the victim to find the attack from upstream links by testing possible routes, and then determine which one carries the attack traffic. Another traceback technique is messaging. Bellovin first proposed an ICMP message to find the source of forged IP packets [3]. Many other improved versions of messaging traceback schemes are proposed later, such as intension-driven ICMP traceback [14]. Logging involves storing the traffic data for analysis. Although to store all the data in the network is impossible, probabilistic sampling or storing transformed information is still feasible. Snoreren [17] proposed a hash-based logging traceback method that can even find the source of a single packet in some situations. However, this method also has excessive processing and storage requirements, which makes it difficult to be widely deployed. Packet marking involves inserting traceback data into the IP packet on its way through the various routers from the attack source to the destination. These marks in the IP packets can be used to reconstruct the path of the malicious traffic. Probabilistic Packet Marking (PPM) [16] is one of the packet marking methods. It lets routers mark the packets with path information in a probabilistic manner and lets the victim reconstruct the attack path by using the marked packets. PPM encodes the information in rarely used identification field within the IP header (used for identifying which packet a fragment belongs

to). To reduce the data to be stored to 16 bits, the compressed edge fragment sampling algorithm was used. PPM requires less traffic volume than ICMP traceback, but encounters computational difficulties as the numbers of attack sources increases. Currently there are also many improved versions of PPM, such as [15] [13]. Another category of packet marking methods, which does not use the probabilistic assumption of PPM and stores the source address in the marking field, is known as the deterministic marking, such as Deterministic Packet Marking (DPM) [4], Flexible Deterministic Packet Marking (FDPM), Deterministic Bit Marking [11] and DPM based on redundant decomposition [10].

To avoid the disadvantages of each traceback scheme, some hybrid schemes are proposed, such as in [1] Al-Duwairi proposed employing packet marking and logging for IP traceback. Their studies show that the proposed schemes offer a drastic reduction in the number of packets required to conduct the traceback process and a reasonable saving in the storage requirement. Yaar et al. proposed a fast Internet traceback scheme in [21], which also aims to reduce the number of packets required to traceback the sources and scale to large distributed attacks with thousands of attackers.

## 5.2 Comparison with other traceback mechanisms

The key difference in our work is on the high effectiveness of the traceback scheme. The major advantages of FDPM are first, it can trace the IP sources with low computation load by its overload prevention mechanism; second, with its low computation load, it achieves high maximum forwarding rates; third, it needs a small number of packets to accomplish the traceback process; and finally, the effectiveness is independent on the attacking distance it needs to trace.

The computation load of FDPM is low, because the algorithms it uses are simple as we can see from previous sections. The marked packets will not increase their size; therefore, no additional bandwidth is consumed. Moreover, with the overload prevention, it can conduct traceback process when the system is loaded heavily. Unfortunately most of the current traceback schemes do not have this overload prevention mechanism.

## 6. Conclusion

The effectiveness of FDPM traceback scheme was discussed in terms of marking efficiency, maximum forwarding rate, and number of packets for reconstruction in this paper. FDPM shows high marking efficiency when it selectively marks the IP packets while the router is under high load for DDoS defense. FDPM also shows a high maximum forwarding rate compared with the baseline of Linux router implementation. This flexibility enables it a practical and effective traceback in a real DDoS defense scenario.

## 7. References

- [1] B. Al-Duwairi, and M. Govindarasu, "Novel Hybrid Schemes Employing Packet Marking and Logging for IP Traceback", *IEEE Transactions on Parallel and Distributed Systems*, Vol.17, No.5, 2006, pp.403-418.
- [2] H. Aljifri, "IP Traceback: A New Denial-of-Service Deterrent?", *IEEE Security & Privacy*, Vol.1, No.3, 2003, pp.24-31.
- [3] S. M. Bellovin, "ICMP Traceback Messages", Internet Draft, Network Working Group, 2000.
- [4] A. Belenky, and N. Ansari, "IP Traceback With Deterministic Packet Marking", *IEEE Communications Letters*, Vol.7, No.4, 2003, pp.162-164.
- [5] H. Burch, and B. Cheswick, "Tracing Anonymous Packets to Their Approximate Source", *Proc. of the 14th Systems Administration Conference (LISA 2000)*.
- [6] R. C. Chen, W. Shi, and W. Zhou, "Simulation of Distributed Denial of Service Attacks", technical report, TR C4/09, School of Information Technology, Deakin University, Australia, 2004.
- [7] S. Floyd and V. Jacobson, "Random Early Detection Gateways for Congestion Avoidance", *IEEE/ACM Transactions on Networking*, Vol.1, No.4, 1993, pp.397-413.
- [8] Z. Gao and N. Ansari, "Tracing Cyber Attacks from the Practical Perspective", *IEEE Communications*, Vol.43, No.5, 2005, pp.123-131.
- [9] L. Garber, "Denial-of-Service Attacks Rip the Internet", *IEEE Computer*, Vol.33, No.4, 2000, pp.12-17.
- [10] G. Jin and J. Yang, "Deterministic Packet Marking Based on Redundant Decomposition for IP Traceback", *IEEE Communications Letters*, Vol.10, No.3, 2006, pp.204-206.
- [11] Y. Kim, J.-Y. Jo, F. L. Merat, "Defeating Distributed Denial-of-Service Attack with Deterministic Bit Marking", *IEEE GLOBECOM 2003*, pp.1363-1367.
- [12] E. Kohler, R. Morris, B. Chen, J. Jannotti, and M. F. Kaashoek, "The Click Modular Router", *ACM Transactions on Computer Systems*, Vol.18, No.3, 2000, pp.263-297.
- [13] M. Ma, "Tabu Marking Scheme for IP Traceback", *19th IEEE IPDPS*, 2005, pp.292b.
- [14] A. Mankin, D. Massey, C.-L. Wu, S. F. Wu and L. Zhang, "On Design and Evaluation of Intention-Driven ICMP Traceback", *Proc. of Computer Communications and Networks*, 2001, pp.159-165.
- [15] K. Park and H. Lee, "On the Effectiveness of Probabilistic Packet Marking for IP Traceback under Denial of Service Attack", *IEEE INFOCOM 2001*, pp.338-347.



- [16] S. Savage, D. Wetherall, A. Karlin and T. Anderson, "Network Support for IP Traceback", *ACM/IEEE Transactions on Networking*, Vol.9, No.3, 2001, pp.226-237.
- [17] A. C. Snoeren, C. Partridge, L. A. Sanchez, C. E. Jones, F. Tchakountio, B. Schwartz, S. T. Kent, and W. T. Strayer, "Single-Packet IP Traceback", *IEEE/ACM Transactions on Networking*, Vol.10, No.6, 2002, pp.721-734.
- [18] Scalable Simulation Framework, <http://www.ssfnet.org>.
- [19] R. Stone, "CenterTrack: An IP Overlay Network for Tracking DoS Floods", 9th Usenix Security Symposium, 2000, pp.199-212.
- [20] Y. Xiang, W. Zhou, and J. Rough, "Trace IP Packets by Flexible Deterministic Packet Marking (FDPM)", *IEEE IPOM 2004*, pp.246-252.
- [21] A. Yaar, A. Perrig, and D. Song, "FIT: Fast Internet Traceback", *IEEE INFOCOM 2005*, Vol.2, pp.1395-1406..