Flexible Deterministic Packet Marking: An IP Traceback System to Find the Real Source of Attacks

Yang Xiang, *Member*, *IEEE*, Wanlei Zhou, *Member*, *IEEE*, and Minyi Guo, *Senior Member*, *IEEE*

Abstract—Internet Protocol (IP) traceback is the enabling technology to control Internet crime. In this paper, we present a novel and practical IP traceback system called Flexible Deterministic Packet Marking (FDPM) which provides a defense system with the ability to find out the real sources of attacking packets that traverse through the network. While a number of other traceback schemes exist, FDPM provides innovative features to trace the source of IP packets and can obtain better tracing capability than others. In particular, FDPM adopts a *flexible mark length strategy* to make it compatible to different network environments; it also adaptively changes its marking rate according to the load of the participating router by a *flexible flow-based marking scheme*. Evaluations on both simulation and real system implementation demonstrate that FDPM requires a moderately small number of packets to complete the traceback process; add little additional load to routers and can trace a large number of sources in one traceback process with low false positive rates. The built-in overload prevention mechanism makes this system capable of achieving a satisfactory traceback result even when the router is heavily loaded. The motivation of this traceback system is from DDoS defense. It has been used to not only trace DDoS attacking packets but also enhance filtering attacking traffic. It has a wide array of applications for other security systems.

Index Terms—DDoS attacks, IP traceback, performance evaluation, routers, security.

1 Introduction

Nowadays, more and more critical infrastructures are increasingly reliant upon the Internet for operations. Given the widespread use of automated attack tools, attacks against Internet-connected systems are now so commonplace that Internet crime has become a ubiquitous phenomenon. Although a number of countermeasures and legislations against Internet crime have been proposed and developed, Internet crime is still on the rise. One critical reason is that researchers and law enforcement agencies still cannot answer a simple question easily: who or where is the real source of Internet attacks? Unless this question is fully addressed, effective defense systems and legislations against such crime would only be blusterous ornaments because knowing where the DDoS attacking packets come from, where a suspect intruder is located, where a malicious e-mail is originated, or where a terrorism website is hosted is the key to identify, track, report, arrest, and punish criminals.

The dynamic, stateless, and anonymous nature of the Internet makes it extremely difficult to trace the sources of Internet crime, since the attacker can forge the source address field in an Internet Protocol (IP) packet. To find the real source of Internet attacks, we must possess the capability of discovering the origin of IP packets without relying on the source IP address field. This capability is called IP traceback. IP traceback systems provide a means to identify true sources of IP packets without relying on the source IP address field of the packet header, and are the major technique to find the real attack sources [1], [2]. Although currently there have been many publications on IP traceback, some key issues that are essential to make an IP traceback scheme into a really usable traceback system were not solved, such as how many sources can be traced in one traceback process, how large is the false positive rate, how many packets are needed to trace one source, and how to alleviate the load of participating routers.

In this paper, a novel and practical IP traceback system, Flexible Deterministic Packet Marking (FDPM), is presented. FDPM belongs to the packet marking family of IP traceback systems. The novel characteristics of FDPM are in its flexibility: first, it can adjust the length of marking field according to the network protocols deployed (*flexible mark length strategy*); second, it can also adaptively change its marking rate according to the load of the participating router by a *flexible flow-based marking scheme*. These two novel characteristics of FDPM make it more practical than other current traceback systems in terms of compatibility and performance. Both simulation and real system implementation prove that FDPM can be used in real network environments to trace a large number of real sources, with low false positive rates, and with low resource requirement on routers.

The rest of this paper is organized as follows: Section 2 surveys previous work on IP traceback research. In

- *Y. Xiang is with the School of Management and Information Systems and the Centre for Intelligent and Networked Systems, Central Queensland University, 2.06 Building 19, Rockhampton, Queensland 4702, Australia. E-mail: y.xiang@cqu.edu.au.*
- *W. Zhou is with the School of Information Technology, Faculty of Science and Technology, Deakin University, Melbourne, Australia. E-mail: wanlei@deakin.edu.au.*
- *M. Guo is with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China. E-mail: guo-my@cs.sjtu.edu.cn.*

Manuscript received 10 Dec. 2007; revised 1 June 2008; accepted 9 June 2008; published online 21 July 2008.
Recommended for acceptance by M.C. Lin.
For information on obtaining reprints of this article, please send e-mail to: tpds@computer.org, and reference IEEECS Log Number TPDS-2007-12-0462.
Digital Object Identifier no. 10.1109/TPDS.2008.132.

Section 3, the system design of FDPM, including encoding scheme, reconstruction scheme, and flow-based marking scheme, is presented. Section 4 describes the simulation on how FDPM can effectively trace a large number of sources in a single traceback process with limited number of packets required. Section 5 describes the simulation on overload prevention of FDPM with its flow-based marking scheme. Section 6 describes the system implementation of FDPM on a PC-based router. Some practical issues such as maximum forward rate, marked rate, and the number needed to reconstruct the sources are analyzed. Section 7 provides the conclusion of this paper.

2 Previous Work on IP Traceback

2.1 Problem Description

Let A_i, $i \in [0, n]$, be the attackers and V be the victim. The attackers and victim are linked by various routers R_j, $j \in [1, m]$. The main objective of IP traceback problem is to identify the n routers directly connected to A_i. The key issue here is to completely identify the n routers with low false positive rates in a single traceback process (conducted by the same traceback point, e.g., V, for a certain period) because correlating the data in different traceback processes is not only extremely difficult but also meaningless for tracing a time-dependent event. In [3], it was stated that a practical IP traceback system should be able to identify a few hundred (10^2) sources/routers out of 1 million routers. Some traceback schemes not only identify the n routers directly connected to A_i but also find the routes between the n routers to victim V. In this paper, we only deal with the problem of finding these n routers (not the routes). In fact, the packets starting from the same origin and arriving at the same destination still may take different routes because of the dynamic nature of the Internet. Therefore, considering routes may not have direct benefits to identify the real source of attacks.

2.2 Current IP Traceback Schemes

There are some survey papers discussing the tradeoffs of different IP traceback schemes, such as [4], [5], and [6]. Current IP traceback schemes can be classified into five categories: link testing, messaging, logging, packet marking, and hybrid schemes. The main idea of the link testing scheme is to start from the victim to trace the attack to upstream links, and then determine which one carries the attack traffic [7], [8]. It consumes huge amount of resources, introduces additional traffic, and possibly causes denial of service when the number of sources needed to be traced increases. Messaging schemes use routers to send ICMP messages from the participating routers to destinations. For a high volume flow, the victim will eventually receive ICMP packets from all the routers along the path back to the source, revealing its location [9], [10], [11]. The disadvantages of messaging schemes are that the additional ICMP traffic would possibly be filtered by some routers, and huge numbers of packets are required by the victim to identify the sources. Logging schemes include probabilistic sampling and storing transformed information. Logging schemes maintain a database for all the traffic at every router within the domain and to query the database to identify the sources of an IP packet. Hash function or Bloom filter is used to reduce the data stored. The main disadvantage of logging schemes is that they heavily overload the participating routers by requiring them to log information about every packet passing by, although it is claimed that it needs only a single packet to find its origin [12], [13], [14], [15].

Packet marking schemes insert traceback data into an IP packet header to mark the packet on its way through the various routers from the attack source to the destination; then the marks in the packets can be used to deduce the sources of packets or the paths of the traffic [16], [17], [18], [19], [20]. As this method overwrites some rarely used fields in IP header, it does not require modification of the current Internet infrastructure. This property makes it a promising traceback scheme to be part of DDoS defense systems [21]. However, the space in IP header that can be utilized is limited. Thus, the information that one packet can carry is also limited. Therefore, many challenges for this category of traceback schemes are raised. For example, the number of sources that can be traced could be limited, the number of packets required to find one source could be large, and the load of the traceback router could be heavy. In Sections 2.3 and 2.4, we detail current packet marking schemes and analyze their limitations.

Recently, there has been also some research on hybrid schemes [22], [23]. In [22], a hybrid traceback scheme combining logging and packet marking is presented to achieve the small number of packets needed to trace a single source and the small amount of resources to be allocated to the participating routers. Although the hybrid schemes try to overcome the disadvantages of each traceback scheme, the complexity of such combination and the practicability of their implementation still need more research.

2.3 Probabilistic Packet Marking Schemes

Probabilistic Packet Marking (PPM) [16] is one stream of the packet marking methods. The assumption of PPM is that the attacking packets are much more frequent than the normal packets. It marks the packets with path information in a probabilistic manner and enables the victim to reconstruct the attack path by using the marked packets. PPM encodes the information in rarely used 16-bit Fragment ID field in the IP header. To reduce the data that is to be stored in 16 bits, the compressed edge fragment sampling algorithm is used.

Although PPM is simple and can support incremental deployment, it has many shortcomings that can seriously prevent it from being widely used. First, the path reconstruction process requires high computational work, especially when there are many sources. For example, a 25-source path reconstruction will take days, and thousands of false positives could happen [18]. Second, when there are a large number of attack sources, the possible rebuilt path branches are actually useless to the victim because of the high false positives. Therefore, the routers that are far away from the victim have a very low chance of passing their identification to the victim because the information has been lost due to overwriting by the intermediate routers.

Many approaches were proposed to overcome the above deficiencies. For example, Song and Perrig [18] proposed an

advanced and authenticated PPM based on the assumption that the victim knows the mapping of the upstream routers. It not only reinforces the capability to trace more sources at one time but also solves the problem of spoofed marking. Another method to reduce the overhead of reconstruction was proposed in [24]. It uses counters to complement the loss of marking information from upstream routers, in order to save computation time and reduce false positives. Adler [25] analyzed the tradeoff between mark bits required in the IP header and the number of packets required to reconstruct the paths.

2.4 Deterministic Packet Marking Schemes

Another stream of packet marking methods, which does not use the above probabilistic assumption and stores the source address in the marking field, is in the category known as the deterministic approaches, such as Deterministic Packet Marking (DPM) [26], [27], our FDPM (the first version of FDPM was published in [28]), and Deterministic Bit Marking [29]. Recently, in [30], the DPM scheme was modified to reduce false positive rates by adding redundant information into the marking fields. Unlike PPM, deterministic approaches only keep the first ingress edge router's information in the marks (but not the whole path). Moreover, they record marks in a deterministic manner (but not a probabilistic manner as in PPM). This category of schemes has many advantages over others, including simple implementation, no additional bandwidth requirement, and less computation overhead. However, enough packets must be collected to reconstruct the attack path (e.g., in the best case, at least two packets are required to trace one IP source with any of the above schemes). Importantly, all previous works neither perform well in terms of, nor have addressed the problems of, the maximum number of sources that the traceback system can trace in a single traceback process, the number of packets needed to trace one source, and the overload prevention on participating routers.

3 Flexible Deterministic Packet Marking Scheme

3.1 System Overview

The FDPM scheme utilizes various bits (called marks) in the IP header. The mark has flexible lengths depending on the network protocols used, which is called flexible mark length strategy. When an IP packet enters the protected network, it is 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 overwritten by intermediate routers when the packet traverses the network. At any point within the network, e.g., the victim host, the source IP addresses can be reconstructed when required.

Processing packets consume resources such as memory and CPU time of a participating router. Therefore, it is possible for a router to be overloaded when there are a large number of arrival packets waiting for FDPM to mark them. A question that has been raised is how much computing power is needed by the marking process of FDPM and is it worth selectively reducing the marking rate? According to the research in [31], the complexity of the forwarding process in a typical router is low (e.g., 2.1 instructions executed per byte of data in a packet) but other processing applications such as data encryption or data compression impose much more complexity (e.g., 10^2 instructions executed per byte of data in a packet). Packet marking requires a router to generate marks including different parts by certain computation methods such as hashing and random number generating. The complexity of packet marking is not measured in this paper; however, it must be more than the forwarding process (as it will be proven in Section 6.3). The flow-based marking scheme is proposed to solve the overload problem. When the load of a router exceeds a threshold, the router will discern the most possible attacking packets from other packets then selectively mark these packets. The aim is to alleviate the load of the router while still maintaining the marking function.

The flexibility of FDPM is twofold. First, it can use flexible mark length according to the network protocols that are used in the network. This characteristic of FDPM gives it much adaptability to current heterogeneous networks. Second, FDPM can adaptively adjust its marking process to obtain a flexible marking rate. This characteristic prevents a traceback router from the overload problems.

The complexity of packet marking schemes can be expressed by the number of packets needed to reconstruct one source. Let b be the number of bits allocated to traceback, and let n_s be the length of the description of the source, e.g., 32 for one source IP address. Because of the deterministic feature of FDPM, it requires only $O(n_s)$ packets to reconstruct one source. However, all the probabilistic schemes require a greater number of packets. For example, an improved PPM [25] requires $O(bn_s^2 2^b(2+\varepsilon)^{4n/2^b})$ packets, for any constant $\varepsilon > 0$, to reconstruct the source with probability greater than $1/2$. Section 4 will give the estimated number of packets needed to reconstruct one source and the experiment results.

3.2 Utilization of IP Header

FDPM is based on IPv4. Possible IPv6 implementation of FDPM will involve adding an extension header in IPv6 packets, which is different with the IPv4 design. The necessity of FDPM IPv6 implementation needs more research because IPv6 has built-in security mechanisms such as authentication headers to provide origin authentication.

Three fields in the IP header are used for marking; they are Type of Service (TOS), Fragment ID, and Reserved Flag. The TOS field is an 8-bit field that provides an indication of the abstract parameters of the quality of service desired. The details of handling TOS and specification of TOS values can be found in [32]. The TOS has been rarely supported by most routers in the past. Some proposed standards such as Differentiated Services in TOS [33], used to indicate particular Quality-of-Service needs from the network, are still under development. Therefore, in FDPM, the TOS field will be used to store the mark if the underlying network protocol does not use the TOS filed.

Fragment ID and Reserved Flag are also exploited. Given that less than 0.25 percent of all Internet traffic are fragments [34], Fragment ID can be safely overloaded without causing serious compatibility problems. Dealing with the fragmentation problems has been discussed in [27]. As shown in Fig. 1, a total of 25 $(8+16+1)$ bits are

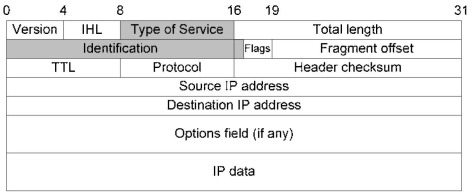


Fig. 1. The IP header fields (darkened) utilized in FDPM.

available for the storage of mark information if the protected network allows overwriting on TOS. When considering the possibility that the TOS field may be unavailable partly or totally, the minimum number of the bits in the IP header is 16 (excluding the 1-bit Reserved Flag). The Reserved Flag is not considered into the marking fields because it is used as the control bit to indicate whether or not the TOS field is being used, which will be discussed later. FDPM can adjust the mark length according to the protocols of the network in which FDPM is deployed. Therefore, even when FDPM is deployed among networks with different protocols, it can still work well because FDPM can differentiate the networks by the control bits.

Because the maximum length of the available mark is 25 bits, more than one packet is needed to carry a 32-bit source IP address. This is the reason why a segment number is needed to reconstruct an IP address into its original order. Each packet holding the mark will be used to reconstruct the source IP address at any point within the network. After all the segments corresponding to the same ingress address have arrived at the reconstruction point, the source IP address of the packets can be reconstructed. In order to keep track of the set of IP packets that are used for reconstruction, the identities showing the packets coming from the same source must be included; therefore, a hash of the ingress address is kept in the mark, known as the digest. This digest always remains the same for an FDPM interface from which the packets enter the network. It provides, on the victim's 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. In essence, this will not introduce false positives, but will result in requiring more packets to reconstruct the sources. In this paper, we have the assumption that no participating router is compromised. Similarly, if a network protocol uses the fields (e.g., TOS) used for marking, the reconstruction scheme will discard those packets, which will also result in requiring more packets to reconstruct the sources.

3.3 Encoding Scheme

Before the FDPM mark can be generated, the length of the mark must be determined based on the network protocols deployed within the network to be protected. According to different situations, the mark length could be 24 bits long at most, 19 bits at middle, and 16 bits at least. Therefore, the flexible length of the marks results in three variations of the

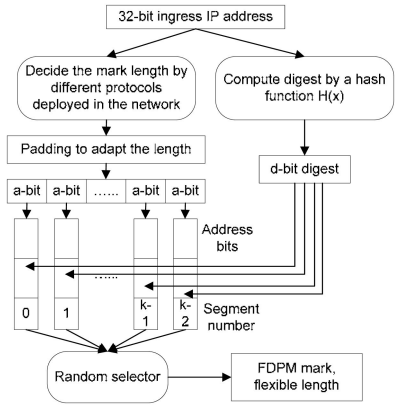


Fig. 2. FDPM encoding scheme.

encoding scheme, which are named as FDPM-24, FDPM-19, and FDPM-16 in the rest of this paper. FDPM encoding scheme is shown in Fig. 2. The ingress IP address is divided into k segments and stored into k IP packets. The padding is used to divide the source IP address evenly into k parts. For example, if $k = 6$, the source address is padded with 4 bits of 0, making it 36 bits long, then each segment will be 6 bits long.

The segment number is used to arrange the address bits into a correct order. The address digest enables the reconstruction process to recognize that the packets being analyzed are from the same source. Without this part, the reconstruction process cannot identify packets coming from different sources, thus will not be able to trace multiple IP packets.

The encoding algorithm is shown in Fig. 3. In FDPM, before the encoding process begins, the length of the mark must be calculated. If the TOS field in the IP packet is not used by the protected network, the 1-bit Reserved Flag in the header is set to 0, and the length of mark is set to 24. Under

```
1.   Marking process at router R, edge interface A, in network N
2.   Set the bit array Digest and Mark to 0
3.   if N does not utilize TOS
4.     Reserved_Flag:=0
5.     7th and 8th bit of TOS:=0
6.     Length_of_Mark:=24
7.   else
8.     Reserved_Flag :=1
9.     if N utilizes Differentiated Services Field or
10.    N supports Precedence and Priority
11.      7th and 8th bit of TOS:=1
12.      Length_of_Mark:=16
13.    else if N supports Precedence but not Priority
14.      7th bit of TOS:=1
15.      8th bit of TOS:=0
16.      Length_of_Mark:=19
17.    else if N support Priority but not Precedence
18.      7th bit of TOS:=0
19.      8th bit of TOS:=1
20.      Length_of_Mark:=19
21.  Decide the lengths of each part in the mark
22.  Digest:=Hash(A)
23.  for i=0 to k-1
24.    Mark[i].Digest:=Digest
25.    Mark[i].Segment_number:=i
26.    Mark[i].Address_bit:=A[i]
27.  for each incoming packet p passing the encoding router
28.    j:=random integer from 0 to k-1
29.    write Mark[j] into p.Mark
```

Fig. 3. Algorithm of FDPM encoding scheme.

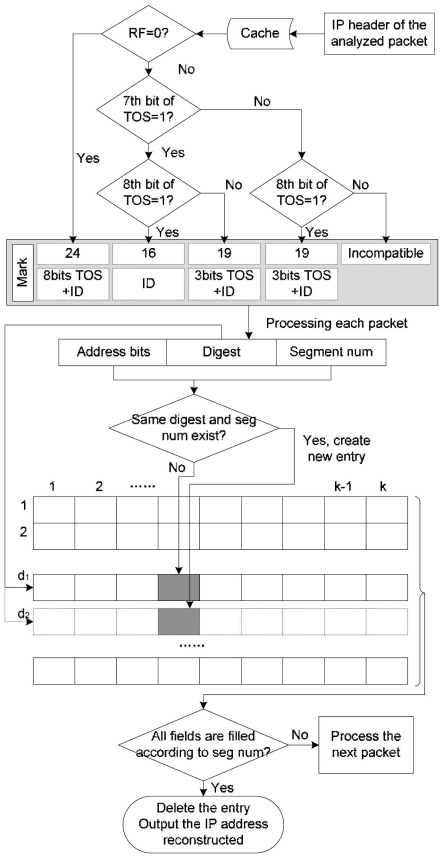


Fig. 4. FDPM reconstruction scheme.

other situations, the length of mark will be 19 or 16, with relevant bit(s) in TOS marked. If the network supports TOS Precedence but not TOS Priority, fourth to sixth bits of TOS are utilized for marking; and if the network supports TOS Priority but not TOS Precedence, first to third bits of TOS are utilized for marking.

3.4 Reconstruction Scheme

The reconstruction process includes two steps: mark recognition and address recovery. When each packet arrives at the point that requires reconstruction, it is first put into a cache because, in some cases, the reconstruction processing speed is slower than the arrival speed of the incoming packets. The cache can also output the packets to another processing unit, by this design the reconstruction methods can be applied in a parallel mode (e.g., if the router has multicore architecture [35], [36]). This will be left as our future work.

The mark recognition step is the reverse process of the encoding process. By reading the control fields in the mark, the length of the mark and which fields in the IP header store the mark can be recognized. If the RF is 0, the mark length is 24 (both TOS and ID are deployed). If the RF is 1, according to different protocols of TOS used, the mark length is 16 or 19.

The second step, address recovery, analyzes the mark and stores it in a recovery table. It is a linked-list table; the number of rows is a variable, and the number of columns in the table is k, representing the number of segments used to carry the source address in the packets. Here, the segment

```
1.   Reconstruction at victim V, in network N
2.   for each coming packet p passing the reconstruction point
3.     mark recognition (length and fields)
4.     if all fields in one entry are filled
5.       output the source IP
6.       delete the entry
7.     else
8.       if same digest and segment number exist
9.         create new entry
10.        fill the address bits into entry
11.      else
12.        fill the address bits into entry
```

Fig. 5. Algorithm of FDPM reconstruction scheme.

number is used to correlate the data into the correct order. The row of the table means the entry; usually each digest owns one entry (source IP address). However, different source IP addresses may have the same digest because the digest is a hash of the source IP address, and is shorter than an IP address. In this case, hash collision is unavoidable. When the hash collision occurs, more than one entry may be created in order to keep as much information as possible. The advantage of this design is that it can reconstruct all possible sources but the disadvantage is it also brings possible irrelevant information. Compared with DPM in [27], our reconstruction process is compatible with different protocols and will not lose any sources even when hash collision occurs. More details about the benefits of this design can be found in Sections 4.2 and 4.4.

Fig. 4 shows the reconstruction scheme. When all fields in one entry are filled according to the segment number, this source IP address is reconstructed and the entry in the recovery table is then deleted. To simplify the description, we present the algorithm of FDPM reconstruction scheme as shown in Fig. 5.

3.5 Flow-Based Marking Scheme

The possibility of the overload problem always exists because the resources of a router are always limited. If the router is overloaded, the marking scheme can be totally ineffective. All packet marking traceback schemes consume the computing power and storage capacity of routers as they need to overwrite many bits in the IP header. Therefore, overload prevention is important to all packet marking traceback schemes. There are many methods to lighten the burden of a router. One is to increase the computing capability of the router, for example, by using multicore-based architecture [36]. Another is to apply an adaptive algorithm to reduce the load of processing of packets when the load of the router exceeds a threshold, which is our novel approach, flexible flow-based marking scheme.

The idea of flow-based marking is to selectively mark the packets according to the flow information when the router is under a high load. Therefore, it can reduce the packet marking load of a router but still maintain the marking and traceback function. Because the main application of FDPM in our research is DDoS defense, the flow-based marking mainly deals with the packets in DDoS attack scenarios. For other applications, this overload prevention mechanism can be modified accordingly to target most possible attacking packets.

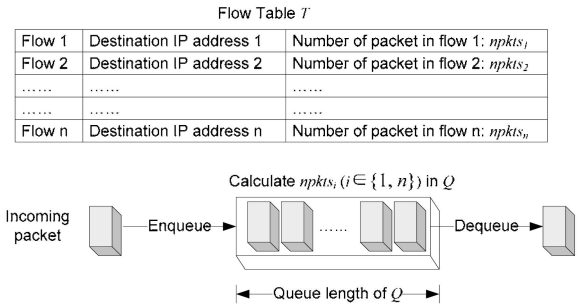


Fig. 6. Dynamic flow table T and FIFO queue Q in FDPM flow-based marking scheme.

The goal of flow-based marking is to mark the most possible DDoS attacking packets (from the same sources but not necessarily with same source IP addresses and to the same destination), then let the reconstruction process in the victim end reconstruct the source by using a minimum number of packets. Ideally, the flow-based marking scheme should be able to keep a separate state for every flow that the router needs to forward, regardless of whether the flow contains large or small number of packets. In our flow-based marking scheme, we aim at reducing complexity and increasing efficiency. It does not keep the state for each flow, but simply uses a single first-in, first-out (FIFO) queue which can be shared by all flows. The advantage of this is that it can be easily implemented in current router architecture, with little impact on the router's packet processing capability. This process is similar to some congestion control schemes such as the Random Early Detection (RED) [37], which isolates the flows that have an unfair share of bandwidth and drops the packets in those flows. The flow-based marking scheme needs to isolate and mark the flows that occupy more bandwidth containing most possible DDoS attacking packets. It can mark packets with a certain probability from each flow, in proportion to the amount of bandwidth the flow uses.

The simple data structures include a dynamic flow table T and a FIFO queue Q, as shown in Fig. 6. Each record in T stands for a flow. Here, the flow means the group of packets that have defined specific subsets of identifiers and are in the Q at a certain time. In DDoS scenarios, attacking packets are classified into different flows according to the destination IP address in the IP header because the aggregation effect is the major feature in DDoS attack traffic. The flow records in T are the destination IP addresses and the number of packets from this flow in the queue Q, denoted as *npkts*. The algorithm of flow-based marking is shown in Fig. 7.

There are two load thresholds L_{max} and L_{min} for the traceback router. L_{max} is the threshold that controls the whole packet marking process, which means the router will not mark any packet if its load exceeds this value. Congestion control mechanisms can be turned on in order to guarantee a best effort service [38] for the router. The load threshold L_{min} means that if the load exceeds this value, the router can still work, but it must reduce its marking load. If the load stays below L_{min}, then the router will just mark all the incoming packets because the router can process all packets without having performance penalty. These two thresholds should be set according to

```
1.   if (load of router R > threshold L_max)
2.     do not mark any packets
3.     turn on congestion control mechanisms
4.   else if (load of router R > threshold L_min)
5.     turn on flow-based marking at R, edge interface A, in network N
6.     for each incoming packet p
7.       check npkts with same destination address of p from T
8.       if (npkts == 0, means no such flow in T)
9.         add a new entry in T, set its npkts = 1
10.      else
11.        npkts ++
12.      insert packet p into Q
13.      calculate marking probability p_a
14.      with probability p_a mark the packet (encoding procedure)
15.      if Q is full
16.        dequeue
17.  else
18.    mark all the packets at R, edge interface A, in network N
```

Fig. 7. Algorithm of FDPM flow-based marking scheme.

real situations in routers. For example, they can be decided by the CPU usage of the router, or the input rate of the router, depending on what is the essential measurement of the router's load. In this paper, input rate is chosen to determine these two load thresholds. How to obtain the best load thresholds is left as a question for future research.

When flow-based marking is turned on, the probability of marking an incoming packet from a particular flow is roughly proportional to the flow's share of bandwidth through the router. We define this probability p_a as

$$p_a = \frac{npkts - \min(npkts_i, i \in \{1,n\})}{\max(npkts_i, i \in \{1,n\}) - \min(npkts_i, i \in \{1,n\})} \times \frac{L_{\max} - L}{L_{\max} - L_{\min}}, \quad (1)$$

where *npkts* is the number of packets in the flow containing current incoming packet, L is the current load of the router. This definition has $P_a \subset [0,1]$. When the current load of the router L reaches L_{max}, P_a becomes 0, which means no marking is performed.

Recall that the flow-based marking scheme aims at isolating and marking the flows containing the most possible DDoS attacking packets, the above design cannot differentiate between flash crowds [39] and DDoS traffic because all the *npkts* values are *current* values, which cannot reflect the accumulating effect of DDoS attacks. In some cases, normal burst flows will also have a large probability of being marked. In DDoS filtering research, the CUSUM algorithm [40] has been widely used to detect the accumulating effect of DDoS attacks. In order to smooth the short-term fluctuations, we apply a low-pass filter with exponentially weighted moving averages (EWMA), which is a fast and practical approach. CUSUM and related algorithms are not used because, here, the detection rate is not the major concern but keeping low complexity is. Therefore, when calculating the marking probability P_a, we use the EWMA *npkts* which is defined as

$$\overline{npkts}_k = \alpha \overline{npkts}_{k-1} + (1-\alpha) npkts_k, \quad (2)$$

where α is the filter constant, which dictates the degree of filtering, e.g., how strong the filtering action will be. By using this low-pass filter, the historical effect of *npkts* can be considered. In our experiments, this filter constant is

TABLE 1
Relationship between the Parameters in FDPM and DPM

k		2	4	8	16	32
s		1	2	3	4	5
a		16	8	4	2	1
FDPM-16	d	0	6	9	10	10
	N_{max}	1	64	512	1024	1024
DPM	d	0	7	10	11	11
	N_{max}	1	128	1024	2048	2048
FDPM-19	d	2	9	12	13	13
	N_{max}	4	512	4096	8192	8192
FDPM-24	d	7	14	17	18	18
	N_{max}	128	16384	131072	262144	262144

set to 0.95. This value controls how many historical values of *npkts* are used. As deciding this value depends on many factors such as the attack characteristics and the degree of tolerance of the defense system, we leave the precise estimation of this value as our future work.

4 Simulation: Trace Large-Scale Sources

4.1 Evaluation Measurement: Maximum Number of Sources

One goal of this research is to find the maximum number of sources that FDPM can trace in a single traceback process. This is a very important evaluation measurement when the traceback system is used to trace large-scale sources. Theoretically, the more marking bits, the more sources FDPM can trace. Because the maximum effective mark length of FDPM is 24, FDPM offers a stronger capability of tracing multiple attacker sources than other traceback schemes. The relationship between the number of packets that are needed to carry one IP address k, the bit of fragment s, the address bits a, the digest bits d, the maximum number of attacker source that can be traced N_{max} under different situations of FDPM, which is affected by the digest bits d, and the same relationship of the parameters in the DPM [27], are shown in Table 1.

From this table, we can see under the ideal situation (it is assumed there is no collision in hash functions, which is the cause of false positives and will be discussed in the next section), the maximum number of sources that can be traced in by FDPM is 262,144. It is 128 times the maximum number of the attacker sources that DPM can trace (2,048). Fig. 8 shows a comparison of the maximum number of sources that can be traced under different encoding schemes by FDPM and by DPM. From the figure and the table, we can see the maximum number that FDPM can trace increases according to the increase of the segment number k. The digest bits d also have to increase if k increases in order to differentiate different sources.

4.2 False Positive Analysis

False positives of FDPM come from collision in hash functions, a situation that occurs when two distinct inputs into a hash function produce identical outputs. If more than one edge router marks the IP packet with the same digest bits, then, in the victim end, the reconstruction will mix the marks from different routers and generate incorrect source IP addresses. The possibility of false positive always exists because the digest bits (at most 18 bits with FDPM-24 encoding scheme) are less than a complete IP address length, which is 32 bits.

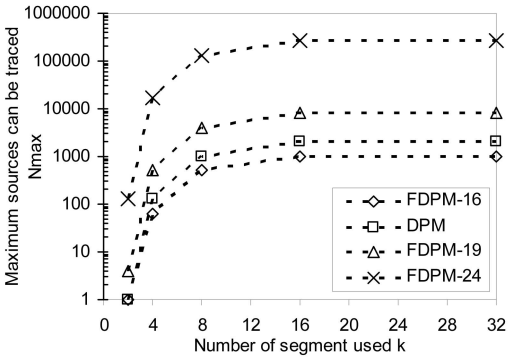


Fig. 8. Maximum number of sources that can be traced, if no hash collision exists.

Let Z be the set of all integers. The domain of an IP address can be written as the set $U = \{x | x \in Z \wedge 0 \leq x < 2^{32}\}$. The domain of a digest can be written as the set $W = \{x | x \in Z \wedge 0 \leq x < 2^d\}$, $d \in [0, 18]$. We have $W \subset U$. If we have hash function f to hash the IP address number in U into the digest in W, then according to [41], the mathematical expectation of different digests from different IP addresses can be written as

$$E[F] = 2^d - 2^d\left(1 - \frac{1}{2^d}\right)^N, \tag{3}$$

where d is the digest bits, and N is the number of different IP addresses. When collision in hash functions occurs, there will be N_d number of IP addresses resulting in the same digest. Then, the expected number of different values in segment bits can be written as

$$E[S] = 2^a - 2^a\left(1 - \frac{1}{2^a}\right)^{N_d}, \tag{4}$$

where a is the address bits. Then, the expected number of permutations that result in a given digest can be written as

$$E[P] = \frac{(E[S])^k}{2^d} = \frac{\left(2^a - 2^a\left(1 - \frac{1}{2^a}\right)^{N_d}\right)^k}{2^d}, \tag{5}$$

where k is the segment number. Recall that in the address recovery step of IP address reconstruction in Section 3.4, entries with the same digest are created to keep as many as possible valid IP addresses. This strategy further reduces false positives. The number of false positives is the total possible permutations, less the number of valid reconstructed IP addresses. The number of valid reconstructed IP addresses has two parts, the ones recovered by different digests, and the ones kept by multiple entries with same digests by the aforementioned strategy. Therefore, the false positive rate can be written as

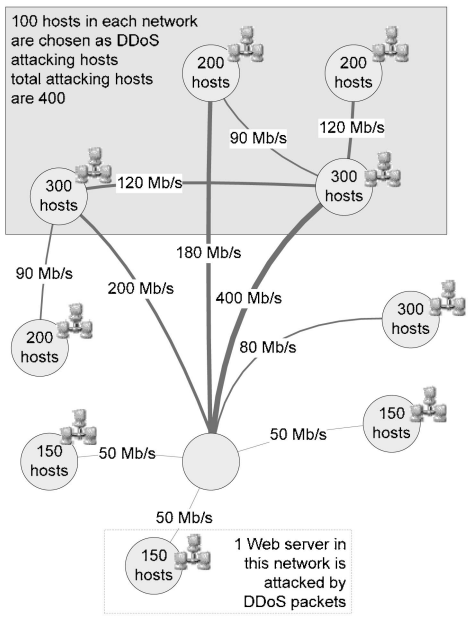


Fig. 9. Network topology in simulation.

$$\eta = \frac{(N - E[F])E[P] - \left[\frac{(N-E[F])N_d}{2^d} + \frac{(E[S])^k}{2^{a+d}}\right]}{N} = \frac{\left[N - 2^d + 2^d\left(1 - \frac{1}{2^d}\right)^N\right]\left[2^a - 2^a\left(1 - \frac{1}{2^a}\right)^{N_d}\right]^k}{2^d N} - \frac{\left[N - 2^d + 2^d\left(1 - \frac{1}{2^d}\right)^N\right]N_d}{2^d N} - \frac{\left[2^a - 2^a\left(1 - \frac{1}{2^a}\right)^{N_d}\right]^k}{2^{a+d}N}. \quad (6)$$

4.3 Simulation Environment

An SSFNet simulator [42] is created to simulate the whole process of FDPM and gather experimental data for analysis. SSFNet is a collection of Java components used for modeling and simulation of IPs and networks at or above the IP level of detail. Our previous work [43] on simulation of DDoS tools, TFN2K and Trinoo, is used to carry out the experiment. An experimental network topology is set up according to a real network as it is shown in Fig. 9. The simulated FDPM system is installed on all the routers in the network.

Three new Java packages are embedded into the SSFNet simulator, which are the Encoding subsystem, the Reconstruction subsystem, and the Flow-based Marking subsystem. The simulation of the Flow-based Marking subsystem will be presented in the later sections. In the Encoding subsystem, the hash function must be chosen carefully because hash collision is one of the main factors affecting the traceback performance in terms of the maximum number of sources that can be traced. Given that all processes in FDPM must be done through the hash function, the function must fulfill two requirements: it must be fast and it must have a strong ability to evenly distribute hashed values throughout the space. The latter requirement minimizes collisions and prevents data items with similar values from being hashed to just one part of the hash table. Three general-purpose hash functions, the MD5 hash function [44], the PJW hash function [45], and the BKDR hash function [46], are selected to test the effectiveness of hashing in FDPM. We chose these functions because they can be implemented in any programming language and are fast with good distribution capability.

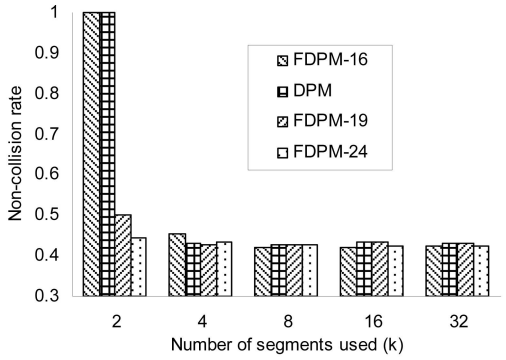


Fig. 10. Noncollision rate for different number of segments used.

4.4 Collision in Hash Functions

We define noncollision rate λ as the percentage of the nonrepeated hashed values in the total hashed values. Fig. 10 shows the average noncollision rate of the hashed digest in the traceback experiments. When the number of segments k (how many packets are used to carry one 32-bit IP source address) is 2, noncollision rate λ is close to 0.5 for FDPM-19 and FDPM-24, and is 1 for FDPM-16 and DPM. When k increases, the noncollision rates λ are stable between 0.42 and 0.45 for all the schemes. Therefore, N_d, the number of IP addresses resulting in the same digest, which equals to $1/(1-\lambda)$, is about 1.75 in our experiments.

According to the above test on collision in hash functions, we can further obtain the average maximum number of sources that FDPM can actually trace. Let us fix the false positive rate η in (6), Section 4.2, as 0.1 percent. The average maximum numbers of sources that can be traced under different situations in the experiments are shown in Fig. 11. Compared with the theoretical analysis in Section 4.1, the actual maximum numbers that FDPM can trace in the experiments are not as large as the theoretical values in the ideal situation. However, in FDPM-24, more than 110,000 (e.g., 10^5) different sources in one traceback process can still be traced; and in FDPM-19, about 500 different sources in one traceback process can still be traced. This is an important feature of FDPM of being a practical traceback system because, to our knowledge, no

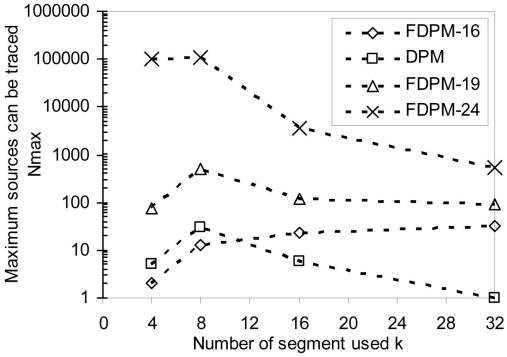


Fig. 11. Maximum number of sources that can be traced in simulation if false positive rate $\eta = 0.1$ percent.

existing system can trace such a large number of sources in a single traceback process. If we allow a larger false positive rate, e.g., $\eta = 1$ percent, the maximum number of sources that can be achieved increases. In this case, FDPM-19 can trace 1,495 sources when $k = 4$. However, the maximum number of sources that DPM can trace is 49 when $k = 4$.

From Fig. 11, we can also derive the optimal segment number k to achieve the maximum number of sources that can be traced. For FDPM-16, the optimal segment number $k_{opt} = 32$; for DPM, $k_{opt} = 8$; for FDPM-19, $k_{opt} = 8$; and for FDPM-24, $k_{opt} = 8$, when false positive rate η is limited to 0.1 percent. We find that in order to be a practical traceback system, FDPM-19 and FDPM-24 are preferred because both FDPM-16 and DPM can only trace 10^2 order of sources, while FDPM-19 can trace 10^3 order of sources and FDPM-24 can trace 10^5 order of sources. In [27], the theoretical estimate of the maximum number of sources that can be traced is 108 when $k = 8$, $a = 4$, $d = 10$, $N_d = 2$. If we use the N_d that is obtained from the above experiment, e.g., $N_d = 1.75$, the maximum number that can be traced by DPM becomes even smaller, e.g., 30, which is far below the requirement of being a practical traceback system.

Collision in hash functions plays an important role in improving the maximum number of sources that can be traced. However, unfortunately under most circumstances, we find that tuning hash functions can be difficult because it requires considerable empirical testing, and it largely depends on what data set is used. Unless the hash table is set up in a preset manner (the possible hash value is subjectively chosen beforehand and cannot fit for the general network environments), the noncollision rate is difficult to improve.

5 Simulation: Overload Prevention

5.1 Evaluation Measurements: Marked Rate and Number of Packets Needed to Trace One Source

The overload prevention mechanism is important to all packet marking traceback schemes. FDPM can adjust the marking rate according to the current load of a router, while still maintaining a good marking function, because it can isolate the most possible attacking flows and then mark them. In Section 2.5, the algorithm of the flow-based marking scheme has been presented. In the following sections, we will discuss the effectiveness of this flexible marking scheme. As presented in Section 4.3, a Java package called Flow-based Marking subsystem in the SSFNet simulator is used to conduct the experiments.

The evaluation measurements of rating the effectiveness of the flow-based marking scheme are the marked rate β, and the number of packets needed to trace one source N_N. Marked rate β is the measurement of marking efficiency, which also reflects the router load imposed by FDPM. A lower value of marked rate β means the participating router will cost fewer resources for traceback. The number of packets needed to trace one source N_N can be used to measure the effectiveness of the traceback power. The less number of packets needed to trace one source, the better chance the defense system can react to the attack. Let us consider the two-packet traceback scenario (the best case scenario for FDPM). In this scenario, the defense system can identify the attack source by just two packets carrying one

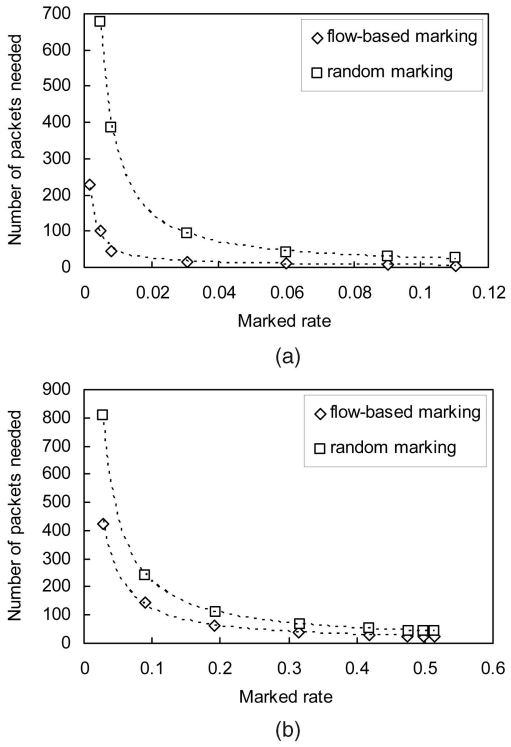


Fig. 12. The relationship between the number of packets needed to trace one source N_N and the marked rate β for the flow-based marking scheme and the random marking scheme in simulation. (a) $k = 2$, $\gamma = 0.1$. (b) $k = 8$, $\gamma = 0.5$.

32-bit IP source address, without waiting for more incoming packets which potentially have unforeseen arrival times. This measurement is very important for rating traceback systems. Theoretically, if all incoming packets are marked, and there is no hash collision problem, then the expected number of packets needed to trace one source can be a Coupon Collector problem [41] decided by the number of segment used k, as

$$E[N_N] = k\left(\frac{1}{k} + \frac{1}{k-1} + \cdots + 1\right). \quad (7)$$

5.2 Flow-Based Marking versus Random Marking

When the load of a router exceeds a certain threshold, 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 packets are not selectively marked, each packet receives the same probability to be marked), the victim which possesses reconstruction will use more packets to reconstruct the sources than the flow-based marking scheme.

Fig. 12a shows the number of packets needed to trace one source N_N and the marked rate β of all the packets passing through the router in the flow-based marking scheme and random marking scheme. The condition is that the router uses two packets to carry a source IP address $(k = 2)$ and the percentage of attacking packets $\gamma = 0.1$. The expected number of packets needed to trace one source $E[N_N]$ is 2 when $k = 2$ according to (7). Fig. 12b shows the marking efficiency in the flow-based marking scheme and the random marking scheme when the router uses eight

packets to carry a source IP address ($k = 8$) and the percentage of attacking packets $\gamma = 0.5$. The expected number of packets needed to trace one source $E[N_N]$ is 22 when $k = 8$.

From Fig. 12, we find that when the router has to reduce its load of packet marking, the flow-based marking scheme performs much better than the random marking scheme in terms of the number of packets needed to trace one source N_N and the marked rate β. For example, in Fig. 12a, when the marked rate is 0.11, the flow-based marking scheme needs six packets to reconstruct one source IP address ($N_N = 6$), while the random marking scheme has $N_N = 26$. Therefore, we can see that the flow-based marking scheme requires much less number of packets to reconstruct the source IP addresses than random marking at different marked rate. If we look at a fixed number of packets needed to reconstruct one source IP address, for example, in Fig. 12b, $N_N = 100$, the flow-based marking scheme needs 13 percent of incoming packets to be marked ($\beta = 0.13$), and the random marking scheme has $\beta = 0.22$. Therefore, we can see if the same numbers of packets are used to reconstruct sources, in order to achieve the same accuracy of traceback, the flow-based marking scheme requires the router to contribute much less marking resources than the random marking scheme.

The random marking scheme cannot control which packet needs to be marked because its selection is random. Therefore, both attacking packets and normal packets have the same possibility to be marked. On the other hand, by using flow-based marking scheme, the attacking packets have more chances to be marked. Thus, in the reconstruction end, less packets are needed to reconstruct the source.

From Fig. 12, we also find that it is not necessary to mark every packet to achieve the minimum number of packets needed to reconstruct one source IP address. In Fig. 12a, only 11 percent of packets need to be marked to achieve minimum $N_N = 6$, which is close to the optimal value $E[N_N] = 2$. In Fig. 12b, only 51.5 percent of packets need to be marked to achieve minimum $N_N = 24$, which is very close to the optimal value $E[N_N] = 22$. In real cases, N_N is affected by other parameters such as marked rate β, percentage of attacking packets γ, and hash noncollision rate λ. The generic relationship between N_N and $E[N_N]$ can be written as

$$N_N = a(\beta, \gamma, \lambda)E[N_N], \tag{8}$$

where a is a function of β, γ, and λ. The details of expression of this function will be our future work.

5.3 Percentage of Attacking Packets

Fig. 13 shows the relationships between the marked rate β, the number of packets needed to trace one source N_N and the percentage of attacking packets γ ($k = 2$ for Fig. 13a and $k = 8$ for Fig. 13b). First, from the figures, we can see that as a higher percentage of attacking packets lead more packets to be marked, less packets are needed at the reconstruction end when the percentage of attacking packets increases. For example, when the percentage of attacking packets γ increases from 0.1 to 0.9, the number of packets needed to trace one source N_N drops from 32 to 6 when $k = 2$; and with the same measurements N_N drops from 289 to 22 when $k = 8$. Second, and more importantly, the marked rate β increases in an almost direct ratio according to the change of the percentage of attacking packets γ in both figures. This proves that the flow-based marking scheme can mark most of the attacking packets, which means FDPM can effectively mark the most possible attacking packets when marking load has to be reduced.

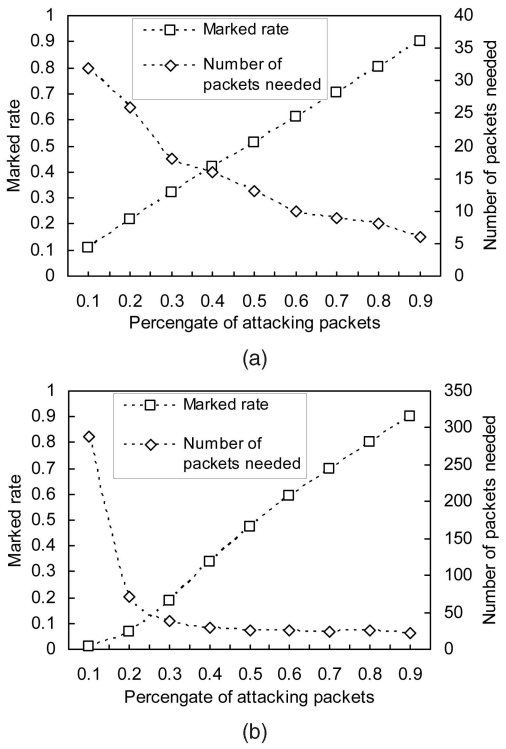


Fig. 13. Relationships between the marked rate β, the number of packets needed to trace one source N_N, and the percentage of attacking packets γ in simulation. (a) $k = 2$. (b) $k = 8$.

6 Real System Implementation

6.1 Evaluation Measurements: Number of Packets Needed to Trace One Source and Maximum Forwarding Rate

Currently, most existing works on IP traceback are based on simulation or theoretical analysis. Few traceback schemes have been implemented and tested by real system implementation. It is very difficult to test the real performance of a traceback scheme if only simulation is conducted. The motivation of real system implementation of FDPM is that we want to know how well it can perform under real environments. The main evaluation measurements we used are the marked rate β, the number of packets needed to trace one source N_N, and the maximum forwarding rate θ_{max}. Maximum forwarding rate is the rate at which an FDPM-enabled router can forward 64-byte packets over a range of input rates. It is difficult to be measured in simulation, but it can be measured in real system implementation. The maximum forwarding rate can be plotted as the line in input rate and forwarding rate coordinates. Ideally, if a router has unlimited computing power and storage, and if the interfaces' bandwidth is unlimited, it would forward every input packet regardless of input rate, corresponding to the line $y = x$.

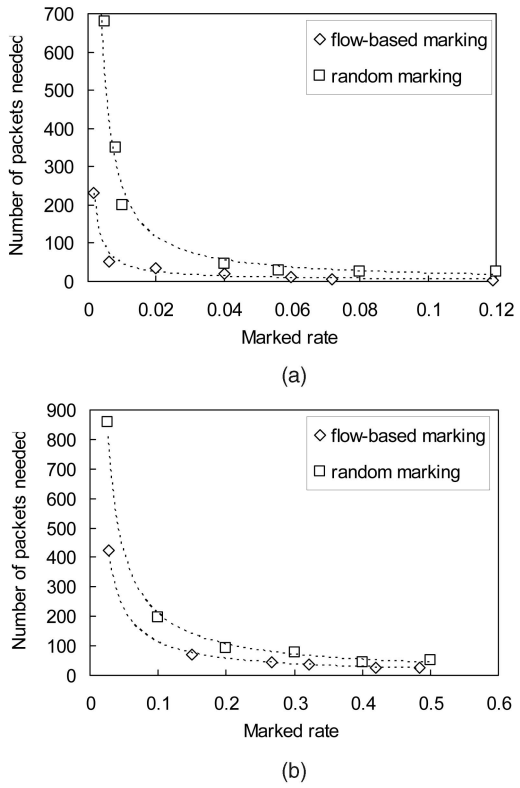


Fig. 14. The relationship between the number of packets needed to trace one source N_N and the marked rate β for the flow-based marking scheme and the random marking scheme in real system implementation. (a) $k = 2$, $\gamma = 0.1$. (b) $k = 8$, $\gamma = 0.5$.

We used the Click modular router [47] to implement our FDPM on PC-based router (Intel Pentium 4 Processors 2 GHz, DRAM 1 Gbyte, double D-Link network 100-Mbps adapters). Click router is a software architecture running on PCs for building flexible and configurable routers, which is assembled from packet processing modules called *elements*. The FDPM Encoding element, Reconstruction element, Flow-based Marking control element, and other associated measuring elements were added to this architecture. Turning on added elements reduces the forwarding capability of the router. The tradeoffs of packet marking schemes will be discussed in Section 6.3.

6.2 Number of Packets for Reconstruction

Fig. 14 shows the relationship between the number of packets needed to trace one source N_N and the marked rate β for flow-based marking scheme and random marking scheme in Click router implementation. The condition of Fig. 14a is that the router uses two packets to carry a source IP address $(k = 2)$ and the percentage of attacking packets $\gamma = 0.1$. The condition of Fig. 14b is that the router uses eight packets to carry a source IP address $(k = 8)$ and the percentage of attacking packets $\gamma = 0.5$. From the comparison between Figs. 12 and 14, we can see that the simulation and real system 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 on routers.

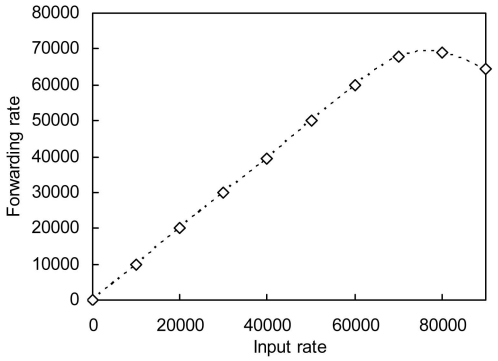


Fig. 15. Maximum forwarding rate of the Click router.

Fig. 14 also shows that, in real cases, we do not have to mark all the packets to make the traceback function work. For example, in Fig. 14a, if 10 percent of the packets are marked $(\beta = 0.1)$, on average only as few as four packets are needed to reconstruct one source by FDPM; if 1 percent of the packets are marked $(\beta = 0.01)$, on average only as few as 40 packets are needed. In Fig. 14b, if 50 percent of the packets are marked $(\beta = 0.5)$, on average only as few as 27 packets are needed to reconstruct one source by FDPM; if 1 percent of the packets are marked $(\beta = 0.01)$, on average only 105 packets are needed. This is strong evidence that FDPM can relieve the participating router from its packet marking load. Random marking requires many more packets to reconstruct one source IP address in real system implementation experiments, which matches our findings in simulation.

6.3 Maximum Forwarding Rate

This section evaluates FDPM-enabled router's performance of forwarding IP packets under different conditions. Fig. 15 shows the maximum forwarding rate θ_{max} for the raw Click router without any packet marking function. This figure can be used as the baseline to compare with FDPM-enabled router's maximum forwarding rate. In our experiments, the maximum forwarding rate θ_{max} of the Click router is 69,000 packets per second. When the input rate exceeds this rate, the router will discard received packets due to the bottleneck of the router's computing power. The maximum forwarding rate in our work is lower than that in [47] because the network adapters in our configuration do 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 θ_{max} of an FDPM-enabled router. Fig. 16 shows when $k = 8$, the curve of maximum forwarding rate θ_{max} of an FDPM-enabled router and the curve when all the packets are marked, which is defined as the all marking scheme. From the figure, we find that the maximum forwarding rate of FDPM is about 15,000 packets per second higher than the case where all the packets are marked. This demonstrates that FDPM's flow-based marking scheme can greatly increase the forwarding rate of a traceback router. Additionally, if we compare Figs. 15

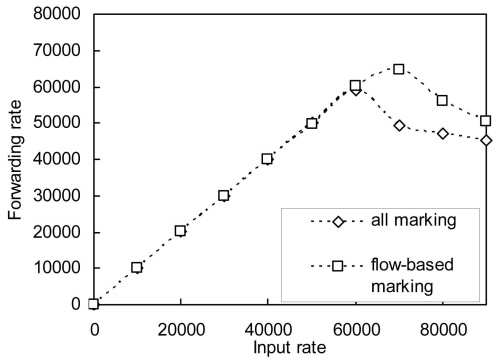


Fig. 16. Maximum forwarding rate of FDPM and all marking schemes.

and 16, we can see that the maximum forwarding rate θ_{max} of an FDPM-enabled router is only about 5,000 packets per second less than the baseline, which means the router sacrifices about 7 percent of its forwarding rate performance to fulfil its traceback function, which is at a moderate level. The result proves that in order to be a practical packet marking traceback system, the marked rate must be flexible; otherwise, the performance of the traceback router will be significantly affected in terms of maximum forwarding rate θ_{max}. For example, if all incoming packets have to be marked (like other current traceback systems), the maximum forwarding rate θ_{max} of the router will reduce 29 percent, which is not acceptable for a normally functioning router.

Table 2 shows the relationship between the percentage of attacking packets γ and the maximum forwarding rate θ_{max} of both an FDPM-enabled router and all marking scheme. From the table, we find that the maximum forwarding rate θ_{max} is not sensitive to the change of γ because FDPM's flow-based marking scheme can dynamically select most likely DDoS packets to be marked, when the load of router exceeds the threshold L_{min}. Again, we can see that the maximum forwarding rate θ_{max} of FDPM is much higher than the all marking scheme.

TABLE 2
Relationship between the Percentage of Attacking Packets and the Maximum Forwarding Rate

Percentage of attacking packets γ	θ_{max} of flow-based marking scheme	θ_{max} of all marking scheme
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

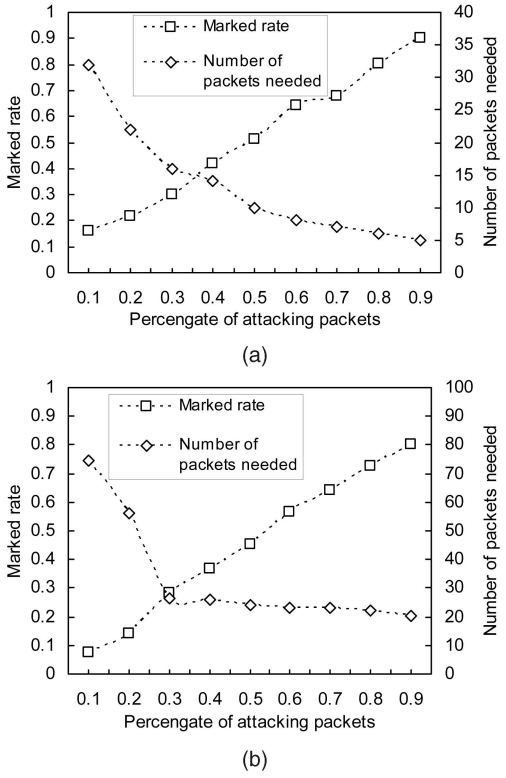


Fig. 17. Relationships between the marked rate β, the number of packets needed to trace one source N_N, and the percentage of attacking packets γ in real system implementation. (a) $k = 2$. (b) $k = 8$.

6.4 Percentage of Attacking Packets

Fig. 17 shows in real system implementation the relationships between the marked rate β, the number of packets needed to trace one source N_N, and the percentage of attacking packets γ ($k = 2$ for Fig. 17a and $k = 8$ for Fig. 17b). The comparison between Figs. 13 and 17 shows that the trend in Click router implementation is the same as in SSFNet simulation. The performance of FDPM in real system implementation is slightly better than in simulation. For example, when the percentage of attacking packets γ increases from 0.1 to 0.9, the number of packets needed to trace one source N_N drops from 32 to 5 when $k = 2$; and with the same measurements N_N drops from 75 to 22 when $k = 8$. The experimental values of N_N are very close to the expected values $E[N_N]$. Again, the real system implementation experiments demonstrate that FDPM can effectively mark the most possible attacking packets when marking load has to be reduced.

7 Conclusion

FDPM is suitable for not only tracing sources of DDoS attacks but also DDoS detection. The main characteristic of DDoS is to use multiple attacking sources to attack a single victim (the aggregation characteristic). Therefore, at any point in the network, if there is a sudden surge in the number of packets with the same destination address and with the same group of digest marks, it can be a sign of a DDoS attack. More details can be found in [48].

In FDPM, the marks in packets do not increase their size; therefore, no additional bandwidth is consumed. Moreover,

with the overload prevention capability, FDPM can maintain the traceback process when the router is heavily loaded, whereas most current traceback schemes do not have this overload prevention capability. Compared with other schemes, FDPM only needs 10^2 packets to trace up to 10^5 sources, so the sources/packets ratio is the highest. FDPM requires little computing power and adaptively keeps the load of routers in a low degree. Where compatibility is concerned, FDPM does not need to know the network topology, and it can be implemented gradually because it has the control bits to differentiate different network protocols used.

An effective traceback system is essential to control Internet crime. While some research has been done, to the best of our knowledge, none of the previous work has fully solved problems such as the maximum number of sources that a traceback system can trace in one traceback process, and the possible overload problem of participating router. We are among the first to examine overload prevention in traceback systems. Compared with other IP traceback schemes, FDPM provides more flexible features to trace IP packets than other packet marking schemes, and can obtain better tracing capacity. To summarize this paper, we list our major contributions here:

1. A novel and practical packet marking traceback system, incorporating a flexible mark length strategy and flexible flow-based marking scheme, is proposed.
2. Simulation and real system implementation show FDPM produces better performance than any other current traceback scheme in terms of false positive rates, the number of packets needed to reconstruct one source, the maximum number of sources that can be traced in one traceback process, and the maximum forwarding rate of traceback-enabled routers.

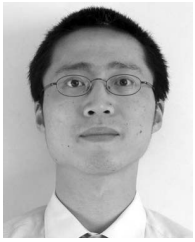
ACKNOWLEDGMENTS

This work is partially supported by the ARC Discovery grant (Project number DP0773264). This work is supported by the National High-Tech Research and Development Plan of China (863 Plan) under Grant Nos. 2008AA01Z106 and 2006AA01Z202, the National Natural Science Foundation of China under Grant Nos. 60811130528, 60725208, and 60533040, and Shanghai Pujiang Plan No. 07pj14049.

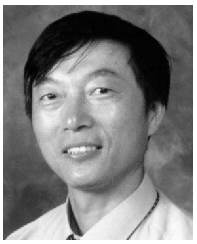
REFERENCES

[1] H. Farhat, "Protecting TCP Services from Denial of Service Attacks," *Proc. ACM SIGCOMM Workshop Large-Scale Attack Defense (LSAD '06),* pp. 155-160, 2006.
[2] H. Wang, C. Jin, and K.G. Shin, "Defense against Spoofed IP Traffic Using Hop-Count Filtering," *IEEE/ACM Trans. Networking,* vol. 15, no. 1, pp. 40-53, 2007.
[3] M.T. Goodrich, "Efficient Packet Marking for Large-Scale IP Traceback," *Proc. Ninth ACM Conf. Computer and Comm. Security (CCS '02),* pp. 117-126, 2002.
[4] H. Aljifri, "IP Traceback: A New Denial-of-Service Deterrent," *IEEE Security and Privacy,* vol. 1, no. 3, pp. 24-31, 2003.
[5] A. Belenky and N. Ansari, "On IP Traceback," *IEEE Comm.,* vol. 41, no. 7, pp. 142-153, 2003.
[6] Z. Gao and N. Ansari, "Tracing Cyber Attacks from the Practical Perspective," *IEEE Comm.,* vol. 43, no. 5, pp. 123-131, 2005.
[7] H. Burch and B. Cheswick, "Tracing Anonymous Packets to Their Approximate Source," *Proc. 14th Systems Administration Conf. (LISA '00),* pp. 319-327, 2000.
[8] R. Stone, "CenterTrack: An IP Overlay Network for Tracking DoS Floods," *Proc. Ninth USENIX Security Symp. (Security '00),* pp. 199-212, 2000.
[9] S.M. Bellovin, *ICMP Traceback Messages—Internet Draft,* Network Working Group, 2000.
[10] A. Mankin et al., "On Design and Evaluation of Intention-Driven ICMP Traceback," *Proc. 10th Int'l Conf. Computer Comm. and Networks (ICCCN '01),* pp. 159-165, 2001.
[11] C. Jin, H. Wang, and K.G. Shin, "Hop-Count Filtering: An Effective Defense against Spoofed DDoS Traffic," *Proc. 10th ACM Conf. Computer and Comm. Security (CCS '03),* pp. 30-41, 2003.
[12] N.G. Duffield and M. Grossglauser, "Trajectory Sampling for Direct Traffic Observation," *Proc. ACM SIGCOMM '00,* pp. 271-282, 2000.
[13] A.C. Snoeren et al., "Single-Packet IP Traceback," *IEEE/ACM Trans. Networking,* vol. 10, no. 6, pp. 721-734, 2002.
[14] T. Baba and S. Matsuda, "Tracing Network Attacks to Their Sources," *IEEE Internet Computing,* vol. 6, no. 3, pp. 20-26, 2002.
[15] J. Li et al., "Large-Scale IP Traceback in High-Speed Internet: Practical Techniques and Theoretical Foundation," *Proc. IEEE Symp. Security and Privacy (S&P '04),* pp. 115-129, 2004.
[16] S. Savage et al., "Network Support for IP Traceback," *ACM/IEEE Trans. Networking,* vol. 9, no. 3, pp. 226-237, 2001.
[17] K. Park and H. Lee, "On the Effectiveness of Route-Based Packet Filtering for Distributed DoS Attack Prevention in Power-Law Internet," *Proc. ACM SIGCOMM '01,* pp. 15-26, 2001.
[18] D.X. Song and A. Perrig, "Advanced and Authenticated Marking Schemes for IP Traceback," *Proc. IEEE INFOCOM '01,* pp. 878-886, 2001.
[19] M. Waldvogel, "GOSSIB versus IP Traceback Rumors," *Proc. 18th Ann. Computer Security Applications Conf. (ACSAC '02),* pp. 5-13, 2002.
[20] A. Yaar, A. Perrig, and D. Song, "Pi: A Path Identification Mechanism to Defend against DDoS Attacks," *Proc. IEEE Symp. Security and Privacy (S&P '03),* pp. 93-107, 2003.
[21] T. Peng, C. Leckie, and K. Ramamohanarao, "Survey of Network-Based Defense Mechanisms Countering the DoS and DDoS Problems," *ACM Computing Surveys,* vol. 39, no. 1, pp. 1-42, 2007.
[22] B. Al-Duwairi and M. Govindarasu, "Novel Hybrid Schemes Employing Packet Marking and Logging for IP Traceback," *IEEE Trans. Parallel and Distributed Systems,* vol. 17, no. 5, pp. 403-418, May 2006.
[23] C. Gong and K. Sarac, "IP Traceback Based on Packet Marking and Logging," *Proc. IEEE Int'l Conf. Comm. (ICC),* 2005.
[24] Y.K. Tseng, H.H. Chen, and W.S. Hsieh, "Probabilistic Packet Marking with Non-Preemptive Compensation," *IEEE Comm. Letters,* vol. 8, no. 6, pp. 359-361, 2004.
[25] M. Adler, "Trade-Offs in Probabilistic Packet Marking for IP Traceback," *J. ACM,* vol. 52, no. 2, pp. 217-244, 2005.
[26] A. Belenky and N. Ansari, "P Traceback with Deterministic Packet Marking," *IEEE Comm. Letters,* vol. 7, no. 4, pp. 162-164, 2003.
[27] A. Belenky and N. Ansari, "On Deterministic Packet Marking," *Computer Networks,* vol. 51, no. 10, pp. 2677-2700, 2007.
[28] Y. Xiang, W. Zhou, and J. Rough, "Trace IP Packets by Flexible Deterministic Packet Marking (FDPM)," *Proc. IEEE Int'l Workshop IP Operations and Management (IPOM '04),* pp. 246-252, 2004.
[29] Y. Kim, J.Y. Jo, and F.L. Merat, "Defeating Distributed Denial-of-Service Attack with Deterministic Bit Marking," *Proc. IEEE Global Telecomm. Conf. (GLOBECOM '03),* pp. 1363-1367, 2003.
[30] G. Jin and J. Yang, "Deterministic Packet Marking Based on Redundant Decomposition for IP Traceback," *IEEE Comm. Letters,* vol. 10, no. 3, pp. 204-206, 2006.
[31] T. Wolf and J.S. Turner, "Design Issues for High-Performance Active Routers," *IEEE J. Selected Areas in Comm.,* vol. 19, no. 3, pp. 404-409, 2001.
[32] *Type of Service in the Internet Protocol Suite,* RFC1349, Network Working Group, 1992.
[33] *Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers,* RFC2474, Network Working Group, 1998.
[34] I. Stoica and H. Zhang, "Providing Guaranteed Services without Per Flow Management," *Proc. ACM SIGCOMM '99,* pp. 81-94, 1999.
[35] R. Ennals, R. Sharp, and A. Mycroft, "Task Partitioning for Multi-Core Network Processors," *Lecture Notes in Computer Science,* pp. 76-90, Springer, 2005.
[36] W. Zhou, "Using Multi-Core to Support Security-Sensitive Applications," *Proc. IFIP Int'l Conf. Network and Parallel Computing (NPC '07),* http://www.deakin.edu.au/~wanlei/papers/MultiCoreSec Wanlei0709.pdf, 2007.

[37] S. Floyd and V. Jacobson, "Random Early Detection Gateways for Congestion Avoidance," *IEEE/ACM Trans. Networking,* vol. 1, no. 4, pp. 397-413, 1993.
[38] P. Gevros et al., "Congestion Control Mechanisms and the Best Effort Service Model," *IEEE Network,* vol. 15, no. 3, pp. 16-26, 2001.
[39] J. Jung, B. Krishnamurthy, and M. Rabinovich, "Flash Crowds and Denial of Service Attacks: Characterization and Implications for CDNs and Web Sites," *Proc. 11th Int'l World Wide Web Conf. (WWW '02),* pp. 252-262, 2002.
[40] H. Wang, D. Zhang, and K.G. Shin, "Change-Point Monitoring for the Detection of DoS Attacks," *IEEE Trans. Dependable and Secure Computing,* vol. 1, no. 4, pp. 193-208, Oct.-Dec. 2004.
[41] W. Feller, *An Introduction to Probability Theory and Its Applications.* John Wiley & Sons, 1968.
[42] *SSFNet, Scalable Simulation Framework,* http://www.ssfnet.org, 2005.
[43] R.C. Chen, W. Shi, and W. Zhou, *Simulation of Distributed Denial of Service Attacks,* TR C04/09, technical report, School of Information Technology, Deakin Univ., 2004.
[44] R. Rivest, *RFC 1321—The MD5 Message-Digest Algorithm,* Network Working Group, 1992.
[45] A. Binstock and J. Rex, *Practical Algorithms for Programmers.* Addison-Wesley, 1995.
[46] B.W. Kernighan and D.M. Ritchie, *The C Programming Language,* second ed. Prentice Hall, 1988.
[47] E. Kohler et al., "The Click Modular Router," *ACM Trans. Computer Systems,* vol. 18, no. 3, pp. 263-297, 2000.
[48] Y. Xiang and W. Zhou, "Mark-Aided Distributed Filtering by Using Neural Network for DDoS Defense," *Proc. IEEE Global Telecomm. Conf. (GLOBECOM),* 2005.



Yang Xiang received the PhD degree in computer science from Deakin University, Melbourne, in 2007. He is currently with the School of Management and Information Systems, Faculty of Business and Informatics, Central Queensland University, Rockhampton, Queensland, Australia. His research interests include network and system security, and wireless systems. He has served as guest coeditor for many journals such as the *Journal of Network and Computer Applications* and *Concurrency and Computation: Practice and Experience*. He has served as PC chair for many conferences such as *14th IEEE ICPADS*, and *11th IEEE HPCC*. He is on the editorial board of the *Journal of Network and Computer Applications*. He is a member of the IEEE and the IEEE Computer Society.



Wanlei Zhou received the PhD degree from the Australian National University, Canberra, Australia, in 1991 and the DSc degree from Deakin University, Victoria, Australia, in 2002. He is currently the chair professor of Information Technology and the associate dean (International), Faculty of Science and Technology, Deakin University, Melbourne. His research interests include distributed and parallel systems, network security, mobile computing, bioinformatics, and e-learning. He has published more than 170 papers in refereed international journals and refereed international conference proceedings. Since 1997, he has been involved in more than 50 international conferences as the general chair, a steering chair, a PC chair, a session chair, a publication chair, and a PC member. He is a member of the IEEE and the IEEE Computer Society.

Minyi Guo received the PhD degree in computer science from University of Tsukuba, Japan. Before 2000, Dr. Guo had been a research scientist of NEC Corp., Japan, and a professor in the School of Computer Science and Engineering, The University of Aizu, Japan. Currently, Dr. Guo is a distinguished chair professor of the Department of Computer Science and Engineering, Shanghai Jiao Tong University, China and an adjunct professor at the University of Aizu. He is also a guest professor at Nanjing University, Huazhong University of Science and Technology, and Central South University, China. Dr. Guo has published more than 160 research papers in international journals and conferences. Dr. Guo has served as general chair, program committee, or organizing committee chair for many international conferences. He is the founder of the International Conference on Parallel and Distributed Processing and Applications (ISPA) and the International Conference on Embedded and Ubiquitous Computing (EUC). He is the editor-in-chief of the *Journal of Embedded Systems*. He is also on the editorial board of the *Journal of Pervasive Computing and Communications*, the *International Journal of High Performance Computing and Networking*, the *Journal of Embedded Computing*, the *Journal of Parallel and Distributed Scientific and Engineering Computing*, and the *International Journal of Computer and Applications*. Professor Guo received the National Science Fund of China (NSFC) for Distinguished Young Scholars in 2007, and is also the PI of the NSFC Key Project "Theoretical and Technical key points of Pervasive Computing." Dr. Guo's research interests include parallel and distributed processing, parallelizing compilers, pervasive computing, embedded systems software optimization, and software engineering. He is a senior member of the IEEE and the IEEE Computer Society, and a member of the ACM, IPSJ, CCF, and IEICE.