Journal of Theoretical and Applied Information Technology
15th November 2012. Vol. 45 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

# A LIGHTWEIGHT COOPERATIVE DETECTION FRAMEWORK OF DDOS/DOS ATTACKS BASED ON COUNTING BLOOM FILTER

[1, 2]CHUNYAN SHUAI, [1]JIANHUI JIANG, [3]XIN OUYANG

[1]Department of Computer Science and Technology, Tongji University, Shanghai 201804, China

[2] Faculty of electric power engineering, Kunming University of Science and Technology, Kunming 650051, Yunnan, China

[3] Faculty of information engineering and automation, Kunming University of Science and Technology, Kunming 650051, Yunnan , China

## ABSTRACT

Detection and traceback of distributed denial of service (DDoS/DoS) attacks have become a challenge for network security. In this paper, we propose a lightweight cooperative detection framework (CCBFF) based on counting bloom filter to detect and trace DDoS/DoS attack online. The CCBFF contains 2 counting bloom filters CBF1 and CBF2. The CBF1 distinguishes different network connection topology of a router by the "options" field of IP-V4, encodes the existing DDoS/DoS attacks and all connected device's addresses and stored them. By querying the CBF1, the CBF2 recognizes suspicious packets, accumulates them and sends out super alerts to the victim. According to super alerts, the CCBFF at the victim-end recognize DDoS/DoS attacks. The experiment results show that the CCBFF is effective in detection and traceback different DDoS attacks.

**Keywords:** *Counting Bloom Filter, Attack Feature Code, Cooperative Detection and Traceback Framework, DDOS/DOS Attack, Router Information Flag*

## 1. INTRODUCTION

Currently distributed denial of service (DDoS/DoS) attacks have become one of the most significant security threats[1], in 2010 the number of DDoS attacks grew 100%[2]. The DDoS consumed the resources of network or the victim and prevented them from providing normal services. In order to avoid being detected and tracked, most of them used spoofed source IP[3] with different kinds and different rate.

There are numerous approaches to DDoS defense and traceback in general[4], including packet statistics and filter, DoS-Attack-Specific detection， information entropy and anomaly-based detection. For brevity, we provide an overview of the approaches similar to CCBFF, and summarize the rest.

As a simple space-efficient randomized data structure, bloom filters[5](BF) are applied in DDoS detection broadly. Yanxiang[6] used modified counting bloom filters (CBFs)[7] at source-end to detect TCP/SYN pairs of spoofed IP DDoS attacks, in which an IP address is split into 4 segments and mapped into 4 CBFs to reduce the memory usage. But with the spoofed IP increasing, the probability segments belong to different IPs are identified to a same IP will increase dramatically, which increases the false positive. Yang Xiang[8] found the network anomalies by using neural network and classified DDoS packets by a BF-based classifier. Xiao[9] detected early-stage SYN flood attack at source and destination simultaneously, in which a CBF counted and identified half-open SYN flood, but this method must know the victim IP before. The scheme in DanPeng[10] observed IPs of normal traffic and stored them in a BF, by IP querying in the BF to identify attack traffic. Keun[11] built a whitelist using a CBF to record the suspicious source IPs at critical Internet sites(CISs). When the CISs were under DDoS attack, these IPs in the list were given higher priority. Kejie[12] extracted attack features based on temporal-correlation and stored them into CBFs and established a spatial-correlation DDoS detection mechanism at the edge routers.

In this paper, we propose a cooperative detection framework (CCBFF) to detect and trace the DDoS. Firstly, we analyze all possible connection topology of a router, use the "options" field of IP-V4 to distinguish different network connection topology and build the CCBFF framework on a router based on CBFs. Compared with other methods, the CCBFF owns lower CPU and memory costs and is suitable for different rates and types of DDoS attacks.

## 2. TYPES OF DDOS/DOS ATTACKS AND MECHANISM

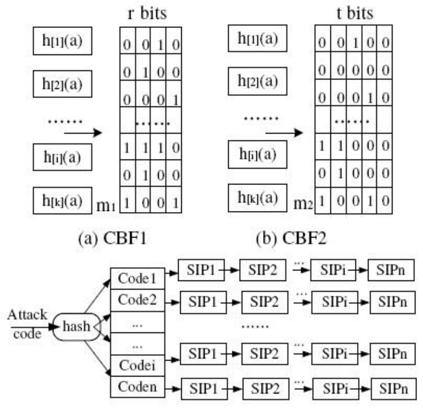
### 2.1 Types Of Ddos/Dos Attacks With Spoofed Source Address

There are different kinds of DDoS/DoS attacks[3], and the victim may be source(e.g smurf) IP or destination(e.g ping) IP in packets. Different DDoS/DoS packets have different signatures including protocols flags, bit flags, addresses, ports, packet length and so on. By encoding the attack features referring to TCP/IP protocol, we differentiate them and form attack feature codes. For example code "080" presents IP protocol, "001" presents ICMP protocol, "006" presents TCP protocol and "00053" presents port 53. Other flags in packets, e.g SYN is encoded to "000010", RST is encoded to "000100" and broadcast address is encoded to "012". These codes are different and exclusive, part of them are shown in Table I.

*TABLE I*
*ATTACK FEATURES AND CODES*

| *DDoS/DoS* | Network layer | Transport layer | features | attack feature codes |
|---|---|---|---|---|
| *SYN Flood* | IP:080 | TCP:006 | SYN=1: 000010 | 080006000010 |
| | | | RST=1: 000100 | 080006000100 |
| *Ping flood* | IP:080 | ICMP:001 | Ping:08 | 08000108 |
| | | | Ping:08,SrcIP=DstIP:11 | 0800010811 |
| *Land* | IP:080 | TCP:006 | SrcIP=DstIP | 08000600001011 |
| *Smurf* | IP:080 | ICMP:001 | Ping:08,DstIP=*: ＊ : ＊ :255:12 | 0800010812 |
| *DNS flood* | IP:080 | UDP:011 | Port=53:0053 | 0800110053 |

### 2.2. Structures Of CBF1, CBF2 And SA-Link



(a) CBF1 (b) CBF2

(c) SA-link

*Figure 1: Structures Of CBF1, CBF2 And SA-Link*

A basic BF for representing a set S= {$a_1$ ,$a_2$, ...$a_n$ } of n elements is described by an array of m bits, initially all are set to 0. A BF uses k independent hash functions $h_i$ to map each item $a_j$ of s to a random number over a range {1,...,m} uniformly, and the bits of $h_i(a_j)$ in bit array are set to 1. After n items in S are stored in BF, by k hash functions $h_i(x)$ mapping, a BF can answer whether x is a member of S or not with a false positive rate (FPR) due to hash collisions, for which it suggests that an element x is in S even though it is not. The reason is that all indexed bits were previously set to 1 by other items. For many applications, FPR may be acceptable as long as it is sufficiently small, to support the deletion operation, CBF was proposed. The specific structures of CBF1, CBF2 and SA-link in the CCBFF can be seen in Figure 1.

1. Counting bloom filter 1 (CBF1). Because the CBF has a constant query delay and can dynamic add/delete items, we use CBF1 to store the IPs of the hosts and devices the router CCBFF-equipped connected, as well as attack feature codes of different DDoS. Each counter of the CBF1 takes up 4 bits[8].

2. Counting bloom filter 2 (CBF2). The CBF2 accumulates the attack packets with attack features or spoofed source IP. Because attack traffics and spoofed IP cannot be forecasted, we initiate the CBF2 according to the bandwidth (details in section 5), and each counter takes up 16 bits which can store $2^{16}$ attack packets at most.

3. Attack source address link of the super alert (SA-link).

• Super alert(SA). When suspicious packets are beyond alert threshold, the CCBFF sends out some super alerts to the victim-end. A super alert contains attack source IP, attack feature code, the initial time of attacks and so on.

• SA-link. The SA-link is links used to store the attack source IP. After receiving super alerts from different attack sources, the CCBFF at the victim-end gets the attack feature code, maps the feature code to SA-link by hash function, and stores the attack source IPs into SA-link. The CCBFF at the victim-end will generate the attack source address links with same attack feature code. The SA-link is just as $Code_i \to SIP_1 \to SIP_2 \to ... \to SIP_n$, in which the $Code_i$ is an attack feature code, and $SIP_1, ..., SIP_i$ are different attack source IPs.

## 2. 3. Detection Mechanism And Process

In this section, we analyze the CCBFF's detection mechanism and make the following reasonable assumptions:

- Routers CCBFFs-equipped are security and topology-aware, by routing information exchange, who know all devices connected including connected ports, hosts, switches, and local area networks (Lan) topology, until to the next router;
- Routers CCBFFs-equipped can dynamically update information with the topology changing.

To a router, there are several actual connection models just as router R4 in the Figure.3. The left of Figure.3 shows the whole topology, attackers, the victim and traffic flows. The right of Figure.3 shows the specific connected structures of R4.

1. Hosts connect to a router directly, for example IP4-1-1 and IP4-1-2 connect to R4's by physical ports rp4-2 and rp4-6 directly.
2. Routers connect each other directly, such as R4 and R1.
3. A Lan connects to a router by hubs or switches, for example Lan9 connects to the port rp4-3 of router R4 by Hub H2.
4. A Lan connect to a router by network address translation (NAT), for example Lan7 connects to R4 by NAT through rp4-7.
5. There is mixed connected model, for example hosts in Lan8, Lan10 and Lan4 connect to R4's rp4-1 by switch S4, and there is a Lan6 connecting to S4 through router R5.

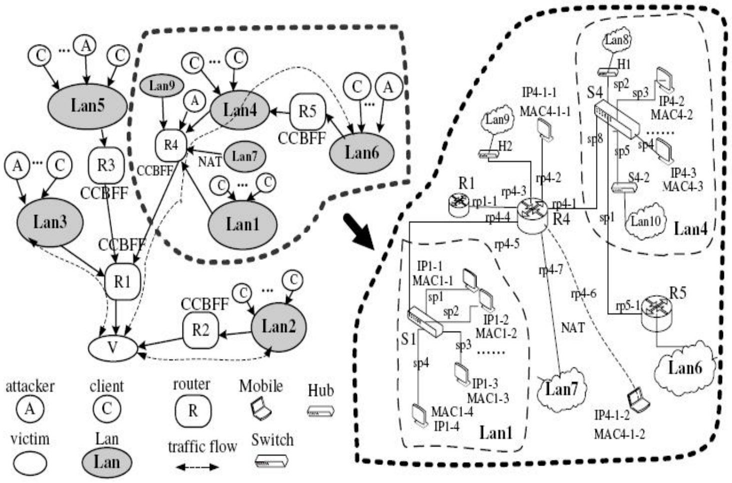


*Figure 2: Network Example With Ddos/Dos Attacks*

By monitor physical ports, exception of the fifth topology, the CCBFF of R4 can make sure where the packets come from. To mark a packet in fifth topology, we introduce router information flags (RIFs) and write it into the "option" field of IP-V4 packet, which usually is NULL except of some routing information and can extend to 20 bytes. In order to avoid being faked, the RIFs contains two flags: packet identification (PID) and the latest router exchange time stamp (RETS). The PID identifies the packet's order, takes up 16 bits, initially is set to a random number and increases by 1 circularly with packet coming. It is difficult to estimate the PID and RETS for the attacker and this can prevent being faked. If the forward port connect to a router directly, the CCBFF will set the RIF of a packet to NULL; if not, the CCBFF will write the RIF of outgoing port into the options field. The CCBFF on R4 monitors all the flows through it, when receiving a packet, it records physical ingress port and connected topology, parses source/destination IPs, options and other features of packet and generates attack codes. If the options field is NULL or other flag, the CCBFF queries the CBF1 to identify spoofed source IP and attack codes. If the options flag is RIFs, the CCBFF compares PID and RETS of the packet with the PID and RETS of ingress port. If the packet is spoofed, the CCBFF will store it's source and destination IPs in CBF2 and forward it. When the accumulation of CBF2 counters exceeds the threshold $T_1$, the CCBFF will send out a super alert to the victim. When received the super alerts from R5, R4, R3 and Lan3, the CCBFF of R1 (the victim) accumulates them in it's SA-link, and If the number of super alerts exceeds the warning threshold $T_2$, the CCBFF of R1 will send the recognition alerts to sources to warn that some DDoS/DoS attacks have happened. Beginning we cannot identify the victim IP, when the abnormal packet appears for the first time, both source and destination IPs in the packet are stored in the CBF2. For subsequent packets, if source IP changes, some or all of $h_i(src\ ip)$ in CBF2 are 0, while all of $h_i(dst\ ip)$ in CBF2 are not 0. The counters of $h_i(dst\ ip)$ corresponding in CBF2 will increase 1. On the contrary, if destination IP changes, the counters of $h_i(src\ ip)$ corresponding will increase 1.

By querying the CBF2, we obtain the appearance times of an address. Because of hash collisions, $k$ counters's value may be inconformity to the same address. Let $x_1$, $x_2$ be source and destination address in the same packet, and $M_i(x_1)$ is the value of $i$th counter compared with uncertain spoofed IP, the victim IP is certain and appearance times are far more than spoofed IP's. So with attacks going on, the victim address corresponds to the maximum counter, that is

Max(Min(M_i(x_1)), Min(M_i(x_2))) (1)

With the attack going on, the spoofed IPs whose appearance is restricted in the CBF2's will bring higher FPR, so we will reset the CBF2's counter whose value is far less than the $T_1$ periodically.

## 3. PARAMETERS, FPR AND FNR

### 3.1 Parameters

When $n$ elements in the set $S$ have been mapped to $m$ counters of the CBF by $k$ different independent hash functions, the probability a counter still being 0 is $p' = (1-\frac{1}{m})^{kn}$ . Because $\lim_{m\to\infty}(1-\frac{1}{m})^{-m} = e$ , $p' = (1-\frac{1}{m})^{kn} \approx e^{-kn/m}$. The probability a counter still being 1 is $1-p'$, and $(1-p')^k$ is the probability after $k$ hashes mapping. Mitzenmacher[12] has proved that the false positive rate(FPR) or false positive probability(FPP) is :

$$\begin{aligned} FPR &= (1-p')^k \\ &= (1-(1-\frac{1}{m})^{kn})^k \\ &\approx (1-e^{-kn/m})^k \\ &= e^{\ln(1-e^{-kn/m})^k} \end{aligned} \tag{2}$$

Let $g = \ln(1-e^{-kn/m})^k$ , and $p = e^{-kn/m}$ , then $\ln(p) = \ln(e^{-kn/m}) = -kn/m$ , and $k = -m/n\ln(p)$ therefore

$$\begin{aligned} g &= \ln(1-e^{-kn/m})^k \\ &= k\ln(1-e^{-kn/m}) \\ &= -m/n \times \ln(p)\ln(1-p) \end{aligned} \tag{3}$$

When p = 1/2, g and FPR get the minimum values that is $FPR_{min} = (0.6185)^{m/n}$. Equivalently, the value of $k$ is

$$\begin{aligned} e^{-kn/m} &= 1/2 \\ k &= \ln 2 \times (m/n) \end{aligned} \tag{4}$$

As we know if an item is a member of $s$, the corresponding counter of CBF will not be 0, so the

$$FNR=0 \tag{5}$$

### 3.2 FPR And False Negative Rate(FNR)

The CCBFF identifies packets according to their features and addresses, so detection accuracy depends on the CBF1, also the FPR and FNR of CCBFF are equal to those of the CBF1, respectively.

**1. $FNR_{CBF1}$** An abnormal packet is recognized as normal one, which includes: (1)The spoofed IP hits but attack feature code does't hit, the probability is $FPR_{IP}$; (2)Both The spoofed IP and attack feature code hit, the probability is $FPR_{IP} \times FNR_{code}$.

Because IPs and attack feature codes use the same CBF1, $FPR_{IP} = FPR_{code}$ and $FNR_{IP} = FNR_{code} = 0$, Therefore the $FNR$ of CBF1 is:

$$\begin{aligned} FNR_{CBF1} &= FPR_{IP} + FPR_{IP} \times FNR_{code} \\ &= FPR_{IP} \\ &= (1-e^{-kn/m})^k \end{aligned} \tag{6}$$

**3.** ***FPR*_CBF1_.** A normal packet is recognized as abnormal one, which includes: (1) The attack feature code hits, the probability is $FNR_{code}$ .(2)Normal IP address doesn't hits, the probability is $FPR_{IP}$ . (3)Normal IP address doesn't hit and attack feature code hits, the probability is $FPR_{IP} \times FNR_{code}$.

Because $FNR_{IP} = FNR_{code} = 0$ , the *FPR* of CBF1 is:

$$\begin{aligned} FNR_{CBF1} &= FPR_{IP} + FPR_{IP} \times FNR_{code} \\ &= FPR_{IP} \\ &= (1 - e^{-kn/m})^{k} \end{aligned} \tag{7}$$

### 3.3. Complexity

When the FPR is no greater than $\varepsilon$ , and the number of hash functions is optimal, to express the set *S* with n elements, the size of CBF array must be[12]

$$m \ge n \frac{\log_2(1/\varepsilon)}{\ln 2} = n \log_2 e \times \log_2(1/\varepsilon) \tag{8}$$

1. Space complexity. As the network topology is relatively fixed and the types of DDoS/DoS attacks are limited, the dominant memory usage of CCBFF focuses on the CBF2 compared with CBF1. Assume that *T* is the attack traffic, packet length is *l* bytes, and both source and destination IP are spoofed. Let every counter of the CBF2 takes up *r* bits, the size of CBF2 array is:

$$\begin{aligned} m &\ge 2n \log_2 e \times \log_2(1/\varepsilon) \\ &\ge 2 \log_2 e \times \log_2(1/\varepsilon) * (Tr/(8l)) \\ &\ge \frac{Tr}{4l} \log_2(e/\varepsilon) \end{aligned} \tag{9}$$

2. Time complexity. When querying, the CCBFF needs check IP and attack feature code in CBF1 and packet in CBF2 by *k* hash functions which are constant, so the querying time complexity is *O*(1).

## 4. EXPERIMENTS

In order to evaluate the efficacy of the CCBFF, we make experiments on the testbed, which includes 48 Vlans and 204 hosts. Each CCBFF host contains Intel 3.0Ghz CPU, 2G RAM, a 100Mbps network interface card(NIC) and a 1Gbps NIC. The CCBFF gets the streams from the router by port Mirroring or netflow and sends back the control information by NIC. During the experiment, attack packets e.g TCP flood, ICMP flood, SYN flood and UDP flood, are sent from attackers in different VLans at different speed.

*Table II: The Performance Of Ccbff Compared With Other Methods*

| | Dataset 1 | | | | | | Dataset 2 | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| AP (x104) | 3 | 6 | 9 | 12 | 15 | 18 | 10 | 20 | 30 | 40 | 50 | 60 | 120 | 150 |
| CA | 30 | 60 | 90 | 120 | 150 | 180 | 100 | 200 | 300 | 400 | 500 | 600 | 1200 | 1500 |
| SA | 32 | 62 | 92 | 122 | 153 | 184 | 101 | 201 | 303 | 401 | 504 | 597 | 1205 | 1507 |
| FPR% | 6.66 | 3.33 | 2.22 | 1.66 | 2.00 | 1.11 | 1.00 | 0.50 | 1.00 | 0.25 | 0.80 | 0.50 | 0.42 | 0.46 |
| DR% | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 99.5 | 100 | 100 |
| FNR% | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.50 | 0 | 0 |
| Notice | AP-attack packets | | | | CA-alerts attacker send | | | | SA- alerts target received | | | | | |
| | FPR-false positive rate | | | DR-detection rate | | | | | FNR-false negative rate | | | | | |

*Table III: Dr, Fpr And Fnr Under High-Rate Attacks*

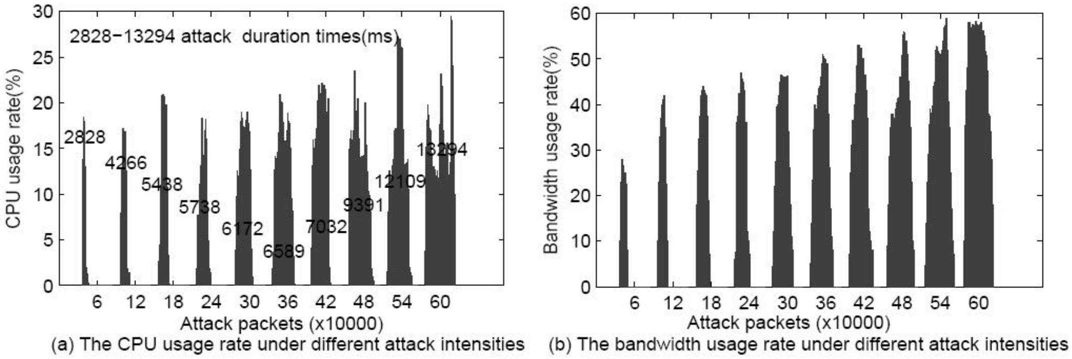
| | | High-rate attack | | | Low-rate attack | | |
|---|---|---|---|---|---|---|---|
| | | DR | DPR | FNR | DR | DPR | FNR |
| DCD | TCP | 99% | <1% | | 68%-98% | <1% | |
| | UDP | 91% | 23% | | 80%-90% | 10%-30% | |
| CUSUM | TCP | 69% | 13.0% | | 52% | 12.9% | |
| Machine learning | TCP | 97.26% | 11.46% | | 65.63% | 12.28% | |
| CCBFF | TCP | 99.5% | 6.66% | 0.5% | 100% | 10.6% | 0 |
| | UDP | 99.5% | 8.23% | 0.6% | 100% | 11% | 0 |
| | Memory usage | | Maximum CPU usage | | Maximum attack speed(Packets/second) | | |
| D_WARD | <7MB | | | | >12000 | | |
| Machine learning | 3.28MB | | | | | | |
| CCBFF | 8.524MB | | 30% | | 20000-200000 | | |

According to the attack rate[13], the detection threshold $T_1$ and $T_2$ are set to 500 and 20 respectively. In Table II, with the packets increasing from 30000 to 150000, the DR of CCBFF can reach 100%. Under low-rate attacks, since the impact of background traffics, the probability normal packets are recognized as abnormal ones will increase, and decrease with the attack intensity increasing. In Table III, we select several typical mechanisms to compare, including machine learning [14], CUSUM[15], D_WARD[16], and DCD[17]. The DR of the CCBFF is higher than DCD, CUSUM and machine learning under any attack intensity. Under high-rate TCP Flood, the FPR of the CCBFF is 6.66% little higher than DCD's, but lower than DCD's under UDP flood. The memory usage of the CCBFF is just higher than D_WARD and machine learning, but the average performance of CCBFF is better than others, higher DR and lower FPR and less memory cost under different attack speed.

In Figure.3, we remove the background traffics, and use different kinds of attack packets to test the CCBFF. When attack packets increase from 60000 to 600000 gradually, the detection time increases from 2828ms to 13294ms, bandwidth usage reaches 60% and memory usage keeps at 8.524MB. The maximum CPU usage is less than 30%, comparing with query time O(1), parsing packet dominates CPU consumption.

In Figure.4 (a) and (b), we increase 30000 attack packets every time, and increase 100000 packets every time in (c) and (d). With attack traffics increasing, the bandwidth and CPU usages grow linearly.

In Figure 5, setting $\varepsilon = 0.001$, $l = 128$ and $r = 16$, the memory usages and attack traffic can be detected by CCBFF grow with the size of CBF2 linearly and the maximum memory usage is 8.524MB.



(a) The CPU usage rate under different attack intensities (b) The bandwidth usage rate under different attack intensities

*Figure 3: Usage Rates Of CPU And Bandwidth, Attack Duration Times Under Different Attack Intensities*

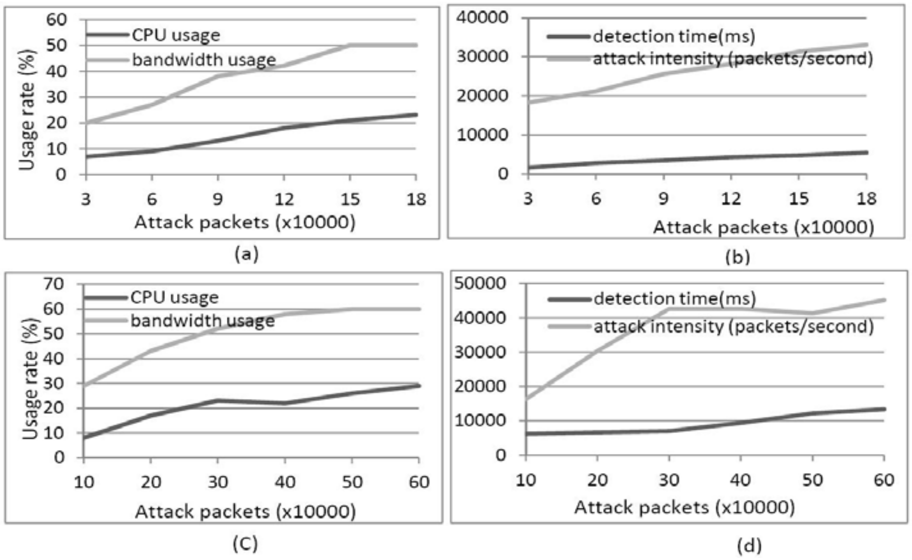


*Figure. 4: Detection Time, Bandwidth And CPU Usages Under Different Attack Intensities(Unit Is X104 )*

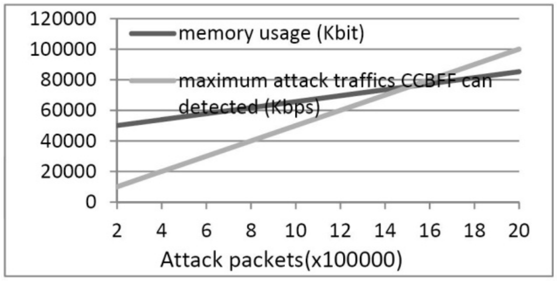


*Figure. 5: Memory Usage And Maximum Attack Traffics Can Be Detected By CCBFF Under Different Size Of CBF2 Array*

## 5. CONCLUSION AND FUTURE WORK

In this paper, we analyze the features of different kinds of DDoS/DoS attacks on TCP and IP layer, and encode them into attack feature codes. Then we analyze all possible connection topology of a router, use the ”options” field of IP-V4 to distinguish different network connection topology, and propose a lightweight cooperative detection framework -CCBFF based on counting bloom filter to detect the DDoS/DoS attack at early stage. The CCBFF includes CBF1, CBF2 and a source address link of super alert(SA-link). Because of topology-aware, CCBFFs on routers can delete/add hosts stored in CBF1 dynamically. All CCBFFs in the network are equivalent and cooperative, and can detect out DDoS/DoS attacks outgoing and incoming at any speed, event new DDoS attacks with spoofed IP.

Compared with other methods the CCBFF is more applicable to high-speed network, has higher detection rate and lower false positive rate. In the future, we will make the CCBFF be suitable for more kinds of DDoS/DoS attacks.

## ACKNOWLEDGEMENTS

This work was supported by the National High Technology Research Foundation of China under Grant No.2007AA01Z142, and by the Key Science-Technology Project of Science and Technology Commission of Shanghai, China under Grant No. 04DZ15032, 06DZ15003.

## REFRENCES:

[1] X. Liu, X. Yang, Y. Lu, “To filter or to authorize network-layer DoS defense against multimillion-node botnets”, Proceedings of the ACM SIGCOMM, ACM Conference on Data Communication, October, 2008, pp.195-206.

[2] Worldwide Infrastructure Security Re-port. Vol.V, Arbor Networks, 2010, http:// www.arbornetworks.com/report.

[3]D. Moore, G. Voelker, and S. Savage, “Inferring internet denial of service activity”, ACM Transactions on Computer Systems, Vol. 24, No. 2, 2006, pp. 115-139.

[4] T. Peng, C. Leckie, K. Ramamohanarao, “Survey of network-based defense mechanisms countering the DoS and DDoS problems, ACM Computing Surveys”, Vol.39, No.1, 2007, pp.1-42.

[5] H. Burton, “Space/Time Trade-Offs in Hash Coding with Allowable Errors”, Comm. ACM, Vol. 13, No. 7, 1970, pp. 422-426.

[6] Yanxiang He, Wei Chen, Bin Xiao, Wenling Peng, “An Efficient and Practical Defense Method Against DDoS Attack at the Source-End”, Proceedings 11th International Conference on Parallel and Distributed Systems, Vol. 2, July 22-22, 2005 , pp.265-269.

[7] L. Fan, P. Cao, J. Almeida, and A. Broder, Summary Cache, “A Scalable Wide Area Web Cache Sharing Protocol”, IEEE/ACM Trans. Networking, Vol. 8, No. 3, 2000. pp. 281-293.

[8] Yang Xiang and Wanlei Zhou, “Classifying DDoS packets in high-speed networks”, International Journal of Computer Science and Network Security, Vol. 6, No. 2B, 2006, pp.107-115.

[9] Xiao B, Chen W, He Y, “A novel approach to detecting DDoS/DoS attacks at an early stage”, Journal on Supercomputing, Vol. 26, 2006, pp.235-248.

[10] Dan Peng, Guiran chang, Rui Guo and Yanjuntan, “Research on DDoS Filtering Algorithm based on Bloom Filter WhiteList”, International Conference on Multimedia and Information Technology, Dec 30-31, 2008, pp.291-297.

[11]Myung Keun Yoon, Kookmin, Using Whitelisting to Mitigate DDoS Attacks on Critical Internet Sites, IEEE Communications Magazine, Vol. 48, No.7 ,2010, pp,110-116

[12] M. Mitzenmacher, Compressed Bloom Filters. IEEE/ACM Transactions on Networking , Vol.10, No.5, 2002, 604-612.

[13] Yang Xiang, Ke Li, Wanlei Zhou. Low-Rate DDoS Attacks Detection and Traceback by Using New Information Metrics, IEEE transactions on information forensics and security, Vol.6, No.2, 2011, pp.426-438.

[14] Kejie Lu, Dapeng Wu, Jieyan Fan, Sinisa Todorovic, Antonio Nucci, Robust and efficient detection of DDoS attacks for large-scale internet, Elsevier Computer Networks, Vol.51, No.3, 2007, pp.5036-5056

[15] H.Wang, D. Zhang, K.G. Shin, Detecting SYN flooding attacks, IEEE Infocom Institute of Electrical and Electronics Engineers Inc, June 23-27, 2002, pp. 1530-1539.

[16] Jelena Mirkovic, Peter Reiher, D-WARD: A Source-End Defense against Flooding Denial-of-Service Attacks, IEEE Transactions on Dependable and Secure Computing, Vol.1, No.2, 2005, pp. 216-232.

[17] Yu Chen, Kai Hwang, Wei-Shinn Ku, Member. Collaborative Detection of DDoS Attacks over Multiple Network Domains, IEEE Transactions on parallel and distributed systems, Vol.18, No.12, 2007, pp.1649-16628.