Int. J. Communications, Network and System Sciences, **2009, 7, 600-607**
doi:10.4236/ijcns.2009.27067 Published Online October 2009 (http://www.SciRP.org/journal/ijcns/).

Research on the Active DDoS Filtering Algorithm Based on IP Flow

Rui GUO[1], Hao YIN[1], Dongqi WANG[2], Bencheng ZHANG[3]
[1]*Department of Computer Science and Technology, Tsinghua University, Beijing, China*
[2]*The Computing Center, Northeastern University, Shenyang, China*
[3]*Electronic Scouting and Commanding Department, College of Shenyang Artillery, Shenyang, China*
Email: *gr@tsinghua.edu.cn*
Received May 22, 2009; *revised July* 7, 2009; *accepted September* 5, 2009

ABSTRACT

Distributed Denial-of-Service (DDoS) attacks against public web servers are increasingly common. Countering DDoS attacks are becoming ever more challenging with the vast resources and techniques increasingly available to attackers. It is impossible for the victim servers to work on the individual level of on-going traffic flows. In this paper, we establish IP Flow which is used to select proper features for DDoS detection. The IP flow statistics is used to allocate the weights for traffic routing by routers. Our system protects servers from DDoS attacks without strong client authentication or allowing an attacker with partial connectivity information to repeatedly disrupt communications. The new algorithm is thus proposed to get efficiently maximum throughput by the traffic filtering, and its feasibility and validity have been verified in a real network circumstance. The experiment shows that it is with high average detection and with low false alarm and miss alarm. Moreover, it can optimize the network traffic simultaneously with defending against DDoS attacks, thus eliminating efficiently the global burst of traffic arising from normal traffic.

Keywords: DDoS Attack, Genetic Algorithm, IP Flow Statistics

1. Introduction

Denial-of-Service (DoS [1]) attacks use legitimate requests to overload the server, causing it to hang, crash, reboot, or do useless work. The target application, machine, or network spends all of its critical resources on handling the attack traffic and cannot attend to its legitimate clients. Both DoS and DDoS are a huge threat to the operation of Internet sites, but the DDoS [2,3] problem is more complex and harder to solve.

There are two main classes of DDoS attacks: bandwidth depletion and resource depletion. A resource depletion attack is an attack that is designed to tie up the resources of a victim system. This type of attack targets a server or process at the victim making it unable to legitimate requests for service. A bandwidth depletion attack is designed to flood the victim network with unwanted traffic that prevents legitimate traffic from reaching the victim system. And there are three main defense approaches: traceback [4]—with the increase of zombies this approach will be invalidated rapidly; filtrate [5]—because this method requires the participation of the communication company and many routers, the filter must be open at all times, and the approach is too costly; throttle [6]—legitimate data stream will be limited because too many data streams converge at a central point. Thus, based on these three methods, three distinct defense approaches emerge: gateway defense, router defense and computer defense.

This paper aims at discussing a low cost, high performance and easy-to-deploy approach [7] which selects five statistical features from IP flow is proposed on filtering DDoS attacks on routers. We use usual statistical traffic of IP flow to get the percentage of traffic of the upriver routers. Then we use the percentage to assign a weight for the router. When DDoS happens, we observe the traffic quota of several chosen routers. Our goal is to maximize goodput, with the weight that we figure out in normal state. At the same time we can calculate which routers should block the traffic. Then the victim server sends a filtering request to these routers to block all traffic from certain sources to the victim.

We present an implementation of these concepts, along with experimental results from our laboratory testbed. In the rest of this section we give a very brief overview of the filtering mechanism. Section 2 tells the

reason of the IP Flow approach. Section 3 presents the architecture based on routers that can support filtering mechanism. Section 4 gives implementation and performance details. In Section 5, we conclude with a discussion of deployment options, as well as related work.

2. IP Flow Filtering Overview

IP flow is composed of IP packets arriving one after another. As the basic data carrying unit of Internet, IP packet holds the upper layer's information and can be easily caught and handled. In the following part of this section IP flow will be divided into the Micro-Flow and the Macro-Flow and we are going to research how to select effective IP flow based detecting features.

2.1. The Micro-Flow and Macro Flow

2.1.1. The Micro-Flow

A Micro-Flow is a packet set who is composed of packets belonging to the same time interval of Internet, and all these packets have the same specific characteristics. These same specific characteristics are called keys. A group of commonly used keys are <Protocol, SrcIP, SrcPort, DestIP, DestPort>. Protocol is the protocol used by the upper layer, SrcIP and SrcPort are the source IP address and the source port number separately. DestIP and SrcIP are the destination IP address and the destination port number separately.

The definition of Micro-Flow is helpful in two ways. First, each key group corresponds to one connection from SrcIP to DestIP, so keys can be used to describe DDoS connection. Second, a key group contains much information which can be used by routers and firewalls to operate each packet.

2.1.2. The Macro-Flow

All the packets belonging to one time interval compose a set which is called the Macro-Flow. Macro-Flow is pooled by Micro-Flows.

The definition of Macro-Flow is helpful in two ways too. First, Detecting features can be formed on the base of Macro-Flow. Second, the information contained in the Macro-Flow is the complementarities to keys.

In experiments, we intercept network traffic by time interval i=10s randomly. On one hand, in order to form the Micro-Flow based features, we classify packets by different keys. On the other hand, we abstract the Macro-Flow based features from the whole i directly.

2.2. IP Flow Based Features

2.2.1. Micro-Flow Based Features

1) Average Number of Packets in Per Flow (ANPPF)

Continuously and randomly generated "legitimate" IP are usually used in attack, so the generating speed of Micro-Flow is quickened, and the packet amount in per flow decrease. There are commonly 1~3 packets in per flow [9].

$$ANPPF = \left(\sum_{j=1}^{FlowNum} PacketsNum_j \right) / FlowNum$$

PacketsNumj is the quantity of packets in the jth flow of a time interval. FlowNum is the quantity of packets of the whole interval. Figure 1 shows the experimental com parison of ANPPF between normal traffic and DDoS traffic (110i~180i).The ANPPF of DDoS traffic which is near 1(attacking traffic is the mix of DDoS traffic generated by tfn2k and normal traffic of internet. ANPPF of tfn2k generating traffic is 1) differs from normal ANPPF (ruleless distribution) significantly.

2) Percentage of Correlative Flow (PCF)

During attack, though the victim still has capability to reply to attacking packets' "requests", the replying packets can not get to the zombies, because the attacking IP addresses are faked. If flow x is from SrcIPx=A to DestIPx=B, and flow y is from SrcIPy=B to DestIPy=A, then we call flow x and y is a pair of Correlative Flow.

$$PCF = CFNum / FlowNum$$

CFNum is two times of the pairs of Correlative Flow. PCF represents the "there is going-out but no coming-back" characteristic of DDoS. As is shown in Figure 2, when DDoS happens (110i~180i), PCF is near 0, while the PCF of normal traffic is 0.4~0.6. The difference between them is distinguishable.

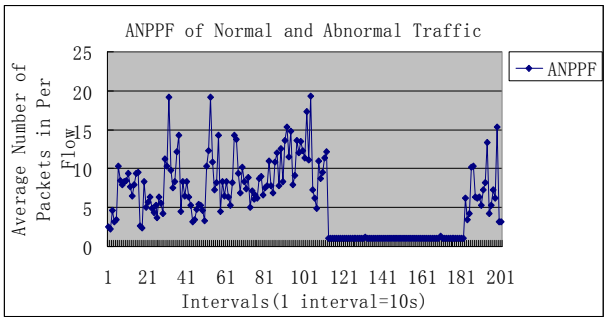

Figure 1. ANPPF of normal and abnormal traffic.

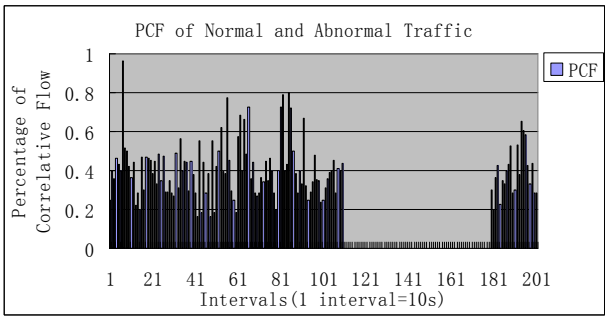

Figure 2. PCF of normal and abnormal traffic.

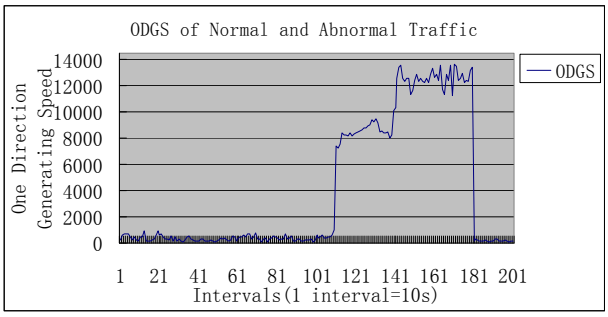

Figure 3. ODGS of normal and abnormal traffic.

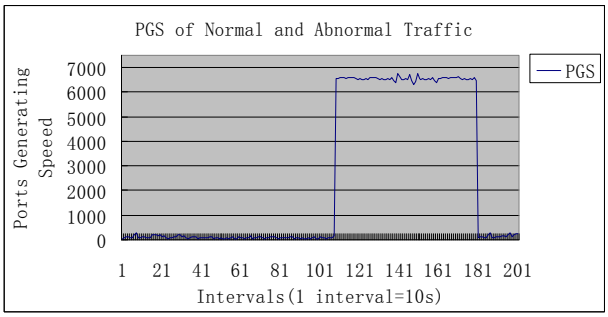

Figure 4. PGS of normal and abnormal traffic.

3) One Direction Generating Speed (ODGS)

Flow generating speed quickens when attack happens or busy time comes. In order to distinguish these two kinds of situations, ODGS is proposed.

$$ODGS = (FlowNum - CFNum) / interval$$

ODGS reflects the sudden increase of traffic when DDoS happens, and it also reflects the "there is going-out but no coming-back" characteristic of DDoS. Figure 4 gives the experimental comparison of ODGS between normal traffic (110i~180i) and abnormal traffic. ODGS' order of magnitude in normal traffic (102) is much smaller than that in the abnormal traffic (104).

4) Ports Generating Speed (PGS)

$$PGS = PortsNum / interval$$

PortsNum is the number of distinct port in one time interval. Some researchers select the size of port [2] as a detecting feature, while we find that many newly emerged services and applications (such as famous p2p application BT) use port number bigger than 1024, so approach of [2] is not suitable anymore. Through deeper investigation, we realize that attackers continuously and randomly generate port too, so PGS is proposed. As is shown in Figure 4, the PGS of normal traffic is not bigger than 200, while PGS of attacking traffic (110~180i) is over thousands.

2.2.2. The Macro-Flow Based Feature

PAP (Percentage of Abnormal Packets)

In order to increase the efficiency of attacking, attacking packets' content parts are usually unfilled or only filled with very few useless bytes (such as famous attacking tools tfn2k, trinoo). This kind of procedure results in the increase of abnormal small packets (for example, some TCP packets are only a little bigger than 40bytes, and UDP packets are only a little bigger than 28bytes). PAP presents this characteristic of DDoS attack by counting the percentage of abnormal packets in the one i(a Macro-Flow). Figure 5 is the comparison of PAP of normal traffic and abnormal traffic. As we can see, there is a significant change of PAP from near 0 to more than 0.9 when DDoS happens (110i~180i).

Defending against DDoS attacks often involves detection and response. There are a number of statistical approaches for detection of DDoS attacks, including the use of IP addresses and TTL [11] values and TCP SYN/FIN packets for detecting SYN flood attacks. Also entropy and Chi-Square statistics are used to differentiate between attack and normal packets. The D-WARD approach [8] uses, in addition to network and transport header statistics, application layer [10] knowledge to implement the filter policy. But all these method require the participation of many routers, the filter must be open at all times, so the approach is too costly.

3. The Design of Statistical Analysis Filtering System

From the Micro-Flow and Macro Flow, we can get the statistical result: $\sum_{j=1}^{n} S_{kj}$ are all connections form source IPk, $\sum_{i=1}^{n} S_{ik}$ are all connections which are routed to destination IPk. It is easy to create probability statistics of access records. Generally, DoS attacks launched by a large number of hosts which host never accessed the victim network before. Meaning during a DDoS attack most of the hosts to the victim are fresh new, which is so different to flash crowd [7]. So we can use history IP database by putting these IP of high frequency in a pool. Common algorithm is not efficient enough to catch up with the line rate of high speed at reasonable memory consumption.

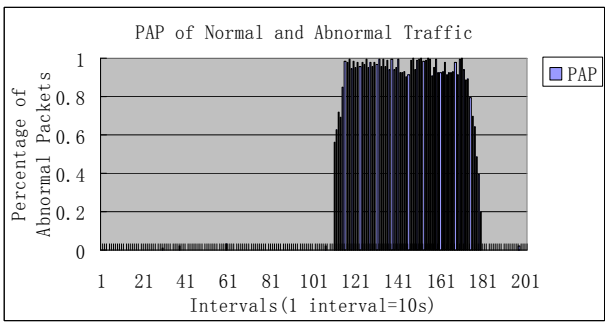

Figure 5. PAP of normal and abnormal traffic.

To address this limitation, one can use the Bloom filter. It reduces space/time complexity by allowing small degree of inaccuracy in membership representation Bloom Filter is chosen to generate the IP address white list. If a host exists in this IP Bloom Filter the router will route the packet to destination, if not it will pass though filter.

The conventional algorithm requires a memory of 1 G bits while our Bloom filter array requires a memory of only 50M bits, at the cost of losing 1% accuracy in membership representation.

A Bloom filter for representing a set S={x1,x2,…,xn} of n elements is described by an array of m bits, initially all set to 0. It uses k independent hash function h1, …, hk with range {1,…,m}. Here we have an assumption that hash functions are perfectly random, which means the hash functions map each item in the universe to a random number uniform over the range {1,…,m}. For each element $x \in S$, the bits hi(x) are set to 1 for $1 \le i \le k$. Alocation can be set to 1 multiple times, but only the first change has an effect. For the membership query if $y \in S$, we check if $\forall i$, hi(y)=1. If $\exists$ hi(i)≠1, then $y \notin S$. If $\forall i$, hi(y)=1 is true, we can assume $y \in S$ with a false positive rate as

$$p_{err} = (1-p_0)^k = \left(1-\left(1-\frac{1}{m}\right)^{kn}\right)^k \approx \left(1-e^{-\frac{kn}{m}}\right)^k$$

The construction of BF is shown in Figure 1. Initially, each bit in the element BF is set to 0 and the pointer list is set to null. Then each history Flow {S,A} *Si*, $i \in [1, n]$ is hashed by function *Hj*, $j \in [1, k]$ with corresponding hit bit in BF being set to 1. A new node of link list is created with the sum field being filled by the sum of previous value and the last 16 bits of the index value of the filter that are being set to 1. The *Si*, $i \in [1, n]$ are hashed *k* times. If the bit has already been set to 1, a new node of link list array is appended to the list. This design does not affect much accuracy because in all the experiments the false positive rates are the same (Figure 6).

As shown in Figure 7 our DDoS defense system has an Offline Training System (OTS) and an Online Filtering System (OFS) and is deployed between the source end and the victim end. From OTS we create whitelist and map the list in BF. The GA-Filter modules are deployed at the edge routers that are close to the attack. During DDoS attacks, if a flow matchs this bloom filter, it will be transmitted by routers, if not it will be filtered by GA-filter. The filtering routers can afford to selectively block traffic to the victim server. In that case legitimate traffic passing from that router is also unnecessarily filteredtogether with the attack traffic. We would like to filter out all attackers and allow all good traffic to reach the server. Unfortunately, in a DDoS attack, it's hard to differentiate attack traffic and legitimate traffic. In this paper, we aim at designing a defense system that contains

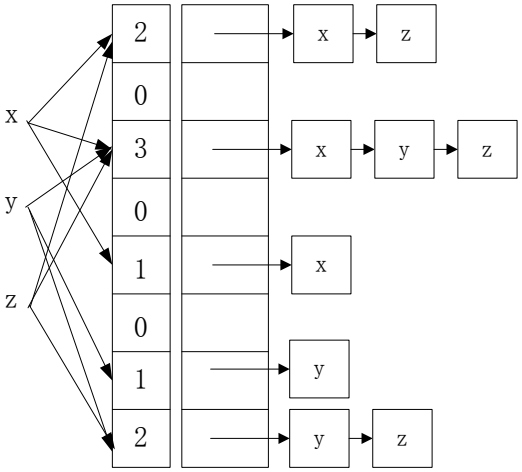

Figure 6. The construction of bloom filter.

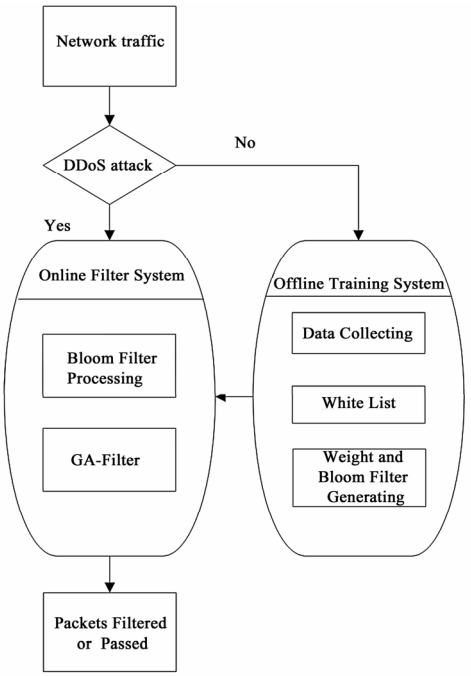

Figure 7. The Filter architecture.

DDoS flooding attacks in high-speed networks. The objectives are to

1) Maximize friendly traffic throughput while reducing attack traffic as much as possible

2) Minimize the disturbance of the defense system on delay performance of friendly traffic

3) Achieve high compatibility to the original systems.

A router-based defense strategy: These routers are inserted in some important point of the network. We envision these routers that are deployed in the network to collaboratively perform the desired countermeasure functions, including detection of DDoS flooding attacks

and access control of network traffic.

3.1. Combinatorial Optimization of Filtering Problem

The filtering problem is a combinatorial optimization of the traffic to victim server, which seeks for maximum legitimate traffic from all good or bad traffic. We assume that there are n distinct routers involved and the traffic in total transmit to victim server is $w_i \sum$, generally, each router j (j = 1, …, n) transmit traffic to victim server has assigned a profit Pi (i= 1, …, n) and the maximum throughput is C. When a router route stream i (i = 1, …, m) to victim server , we define Xi=1; if stream i (i =1, …, m) isn't routed to server, we define Xi =0. So the stream in total is $\sum_{i=1}^{n} w_i x_i$, but the good traffic is $\sum_{i=1}^{n} p_i x_i$,

The problem is to identify a subset of all traffic that leads to the highest possible total good traffic and does not exceed maximum throughput C. Formally, our filtering model can be stated as follows:

Maximize $\sum_{i=1}^{n} p_i x_i$

subject to $\sum_{i=1}^{n} w_i x_i \le c$

with $p_i \ge 0$, $w_i \ge 0$, $C \ge 0$

3.2. Genetic Algorithms for Filtering Bad Traffic

Genetic algorithms are stochastic iterative algorithms for search and optimization that find their origin and inspiration in the Darwinian theory of biological evolution. GA abstract and mimic some of the traits of the ongoing struggle in evolution in order to do a better job in problems that require adaptation, search and optimization. Since we are in fact dealing with artificial systems, we should also feel free to employ whatever device works well for a given class of problems, even if it has no direct biological origin. Genetic Algorithms are computer algorithms that search for good solutions to a problem from among a large number of possible solutions. Genetic Algorithms of our filtering can be stated as follows:

3.2.1. Initial Population

The algorithm begins by creating an initial population which contains M individuals; a mutation probability; a crossover probability; the length of every chromosome N, and the maximum generations. Randomly generate a population of N chromosomes. We randomly generated traffic to victim server and the percentage of bad traffic. Initial transmitting throughput by routers is more than the maximum throughput C which the server can handle.

3.2.2. Encoding of the Chromosomes

Encoding of the problem in a binary string, the length is n, X_i =1, meaning the traffic passes through to the server, X_i =0, meaning the router drops the traffic. Such as X={0，1，0，1，0，0，1} expressing that traffic is passing through router 2, 4, 7. Namely, router 2, 4, 7 will transmit traffic to victim server. We randomly select bits of a chromosome and set it to 0 or 1. For each of the chromosome, test whether the constraint is satisfied. If so, accept it to be a number of the population. If not, drop it and randomly create a new chromosome. The x-vector describes which of the routers that are chosen in each solution, for example, the vector 01001011 means that router NO. 2, 5, 7 and 8 are chosen to route data to server.

3.2.3. Fitness Function

Given a chromosome that represents which router filtrate the traffic, the corresponding fitness function is defined as follow: fitness function $f(X) = \sum_{i=1}^{n} X_i P_i$ subject to $\sum_{i=1}^{n} X_i W_i \le C$.

At first, we define stream *i* passes through a router to victim server , we set X_i =1; if stream *i* (*i* = 1; : : : ;m) drop, we set X_i =0. Considering about n routers, the throughput is $\sum_{i=1}^{n} W_i X_i$ in total, but the goodput is $\sum_{i=1}^{n} P_i X_i$, so how to optimize variable X_i (*i*=1,2,…,n)and maximize goodput. So this problem is subject to two formulas: at $\sum_{i=1}^{n} X_i W_i \le C$ maximize $\sum_{i=1}^{n} P_i X_i$ X_i=1 or 0 (i=1,2,…,n). after analyzing the problem, for the fitness function, it can be stated as follows: $f(x) = \sum_{i=1}^{n} P_i X_i$, X_i=1 or 0 (i=1,2,…,n).

3.2.4. Selection Functions

We choose chromosomes based on probability, and appoint the individual to be the first generation. In the implementation of the program, we tried roulette-wheel methods: the fitness value of each individual is f_i, the probability of *i* is chosen shown as follow: P_{Si} = $f_i / \sum_{i=1}^{n} f_i$; For the initial population, first we figure out the fitness value of each chromosome, and then we calculate selection probability. After the comparison, the chromosome with low chosen probability is eliminated and the high chosen probability chromosome will be copied. This copied chromosome takes the place of the eliminated chromosome. Then the selection of popula-

tion is over.

3.2.5. Crossover

We use single point crossover. The crossover point is determined randomly by generating a random number between 0 and 1. We perform crossover with a certain probability. If crossover probability is 100% then a whole new generation is made by crossover. If it is 0% then whole new generation is made by exact copies of chromosomes from old population. We decided upon crossover rate of P_c. This means that P_c of the new generation will be formed with crossover and 1-P_c will be copied to the new generation.

3.2.6. Mutation

Mutation is made to prevent GA from falling into a local extreme. We perform mutation on each bit position of the chromosome with 0.1 % probability.

4. Performance Evaluation and Comparison

For evaluating our system, we use Bell lab's [11,12] data. Bell lab's data is stored as pure text, and each row of the text is a packet composed of SIP, DIP, SPort, DPort, packet length and ACK (TCP packet) et. The attack launched in our own simulation is constant rate attack, so we choose the constant rate UDP attack data of Bell lab's as the attack samples. (Table 1).

We suppose to have a server that has a capacity of C bandwidth and several routers transmit traffic with different ratio. We want the greatest total benefit without overloading the constraint of the bandwidth. We use a data structure, called cell, with two fields (goodput and traffic) to represent every router (Table 2). Then we use an array of type cell to store all routers in it.

In our experiments, we measured filtering characteristics by the rate of false and rate of missed [13]. In Table 3 shows the sensitivity and accuracy of the Bloom Filter. The ROC curves in Figure 8 and Figure 9 show the sensitivity and accuracy of the neural network. A ROC curve is a plot with the false positive rate on the X axis and the true positive rate on the Y axis. The area below the curve reflects the sensitivity of the neural network. As we can see, the curve is close to both the Y axis and the point (0, 1) which means that we obtained low false positives and the classification capability is good.

Micro-Flow and Macro-Flow detection based detecting features that we described in Section 2 is used to allocate the weights for traffic routing. As indicated in Table 3, that our IP flow based filtering method achieves pretty high accuracy and precision. It's low cost, high performance and easy-to-deploy. It optimizes the web flow; enhance the network efficiency by precluding and dismissing the overall current abruptness of ordinary flow.

Table 1. DDoS traffic.

Country	DDoS Type I		DDoS Type II	
	% of Good Traffic	% of bad Traffic	% of Good Traffic	% of bad Traffic
USA	36.27	43.9	36.2	45.9
Korea	5.8	11.5	0	12
China	18.35	10.3	24.1	0
Taiwan	2.46	6.1	2.4	16.7
Canda	3.64	5.4	3.6	5.4
UK	6.74	5.2	6.7	5.3
Germany	8.4	5.1	8.4	5.2
Australia	2.5	4.3	2.5	1.1
Japan	13.91	4.2	14.2	0
Netherlands	1.93	4.1	1.9	8.4

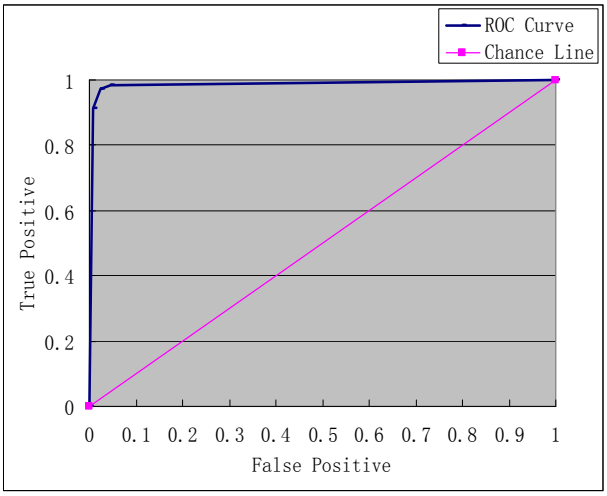

Figure 8. Our own data ROC curve.

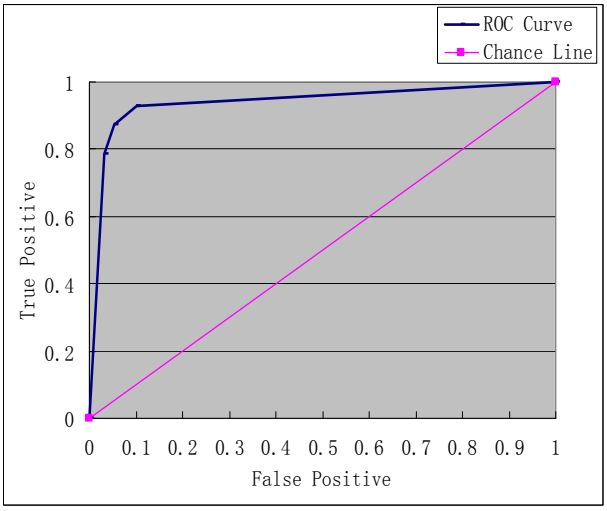

Figure 9. Bell ROC curve.

Table 2. Router information array.

Router 0		Router1		Router2		Router3		Router4	
36.27	80.17	5.8	17.3	18.35	28.65	2.46	8.56	3.64	9.04
Router 5		Router6		Router 7		Router 8		Router9	
6.74	11.94	8.4	13.5	2.5	6.8	13.91	18.11	1.93	6.03

Table 3. The detection results.

Bloom Filter	Rate of false Alarm(%)	Rate of missed Alarm(%)	Average detection Latency of attack (s)
1	0	7.6	12.9
2	1.1	4.5	11.2
3	1.3	2.3	10.3
4	2.6	0	9.8
5	2.9	0	7.6
6	3.6	0	7.5
7	4.9	0	7.1
8	5.9	0	6.9
9	8.6	0	6.5
10	12.8	0	6.3

5. Conclusions

The defense mechanism of DDoS attacks, particularly the multi-based, multi-approached and diversified flow method of offensive artifice, simulating the competition of legal users, inhabits a keystone and difficulty in the internet security arena. In this paper we present five effective detecting features base on the characteristics of IP flow: PAP, ANPPF, PCF, ODGS and PGS. These five features can exploit the abnormalities during DDoS attack. Byproducts of features generation are helpful for filtering. We prove the capabilities of these five features through experimental comparison between their normal values and values in attack.

Our mechanism is characteristically distinct from current methods:

1) Utilizes few resources and does not require participation from all ISP routers. In general, only requires several routers.

2) It's low-cost, high-performance and easy-to-deploy. It allows for simple and convenient updating of the Filter Algorithm.

3) Optimizes the IP flow; enhances the server's efficiency by precluding and dismissing the overall current abruptness of ordinary flow.

All in all, allocating the server and bandwidth resources to both the validation and service components with more efficiency, and applying the algorithm more accurate to filter flooding DDoS are seeking to be done in this sector of internet security.

6. Acknowledgements

This work was supported in part by the National Natural Science Foundation of China under Grant No. 60673184 and Grand No. 60873254, in part by the National 863 program of China under Grant No.2007AA01Z400, in part by the National Basic Research Program of China under Grant No. 2008CB317101, and Tsinghua-ChinaCache CDN Program.

7. References

[1] J. Mirkovic, S. Dietrich, D. Dittrich, and P. Reiher, "Internet denial of service: Attack and defense mechanisms," Prentice Hall PTR, 2004.

[2] V. A. Siris and F. Papagalou, "Application of anomaly detection algorithms for detecting SYN flooding attacks In: Regency H, ed," Global Telecommunications Conf. (GLOBECOM'04). Dallas: IEEE, pp. 2050–2054, 2004.

[3] W. Li, L. F. Wu, and G. Y. Hu, "Design and implementation of distributed intrusion detection system NetNumen," Journal of Software, pp. 1723–1728, 2002.

[4] M. Sung and J. Xu, "IP traceback-based intelligent packet filtering: A novel technique for defending against Internet DDoS attacks," IEEE Trans. on Parallel and Distributed Systems, pp. 861–872, 2003.

[5] A. Chandra and P. Shenoy, "Effectiveness of dynamic resource allocation for handling Internet," University of Massachussets, 2003.

[6] F. Liang and D. Yau, "Using adaptive router throttles against distributed denial-of-service attacks," Journal of Software, pp. 1120–1127, 2002.

[7] A. B. Kulkarni, S. F. Bush, and S. C. Evans, "Detecting distributed denial-of-service attacks using kolmogorov complexity metrics," General Electric Research and Development Center, December 2001.

[8] J. Mirkovic, "D-WARD: Source-end defense against

distributed denial-of-service attacks," PhD thesis, University of California, Los Angeles, pp. 310–321, August 2003.

[9] C. Jin, H. Wang, and K. G. Shin, "Hop-count filtering: An effective defense against spoofed DDoS traffic," Proceedings of the 10th ACM Conference on Computer and Communication Security, ACM Press, pp. 30–41, October, 2003.

[10] Y. Chen, K. Hwang, and Y. K. Kwok, "Filtering of shrew DDoS attacks in frequency domain," lcn, pp. 786–793, The IEEE Conference on Local Computer Networks 30th Anniversary (LCN'05), Jan. 2005.

[11] C. Sangpachatanaruk, S. M. Khattab, T. Znati, R. Melhem, and D. Mosse', "A simulation study of the proactive server roaming for mitigating denial of service attacks," Proceedings of the 36th Annual Simulation Symposium (ANSS'03), pp. 1430–1441, March 2003.

[12] Bell Labs. Bell Labs Internet Traffic Research. http://stat.bell-labs.com/InternetTraffic/index.html.

[13] ICSI Center for Internet Research Traffic Generators for Internet Traffic. http://www.icir.org/models/trafficgenerators.html.

