2011 Fifth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing

Use Double Check Priority Queue to Alleviate Malicious Packet Flows from Distributed DoS attacks

Chu-Hsing Lin[1], Jung-Chun Liu[2], Guan-Han Chen[3], Ying-Hsuan Chen[4],
Chien-Hua Huang[5]

Department of Computer Science and Information Engineering,

Tunghai University, Taichung 407, Taiwan

{chlin[1], jcliu[2], g98350048[3], s963828[4], s963945[5]}@thu.edu.tw

***Abstract*—By using a lot of request packets or garbage packets Distributed Denial of Service attacks occupy network bandwidth and consume performance of the target host. If the attack goal is a commercial website, Distributed Denial of Service attacks will cause transmission delay and more seriously they will deny web services. In this paper, we propose a Double Check Priority Queue structure that effectively mitigates the impact of Distributed Denial of Service attacks so that normal users can still access services.**

Keywords: DDoS attack, network simulator, priority queue, Traffic analysis.

I. INTRODUCTION

The development of the Internet brings much convenience and it has become indispensible of modern life. For instance, the E-commerce, query of living information, and online game have large amount of users. However, new problems take place along with development of the Internet technology. Hackers exploit the vulnerability of browsers and applications to steal data or break down the network system. For example, Distributed Denial of Service (DDoS) attacks will consume the network bandwidth and obstruct normal services[1] and are perilous for commerce websites and government organizations.

To launch DDoS attacks, the attackers useTrojan viruses to control many computers and use them to send a lot of packets to targets that usually are servers or websites. DDoS attacks not only send a large number of request packets so that the server spends lots of resources to process the packets, but also increase network traffic so that normal user's requests are not accepted by the server or responses are not submitted from the server. Thus, they achieve the purpose of denial of service. Due to the fact that the attackers used to control computers to launch attacks, Internet police can hardly track down the address where attacks originate.

For resisting against DDoS attacks[3-7], we use Double Check Priority Queue (DCPQ) structure decrease the affect so that the server can provide normal services when DDoS attacks occur. To determine whether the client is being under attacks or not, we will use two factors: traffic analysis of clients and time interval of received packet.

The remainder of this paper is outlined as follows: Section 2 describes related works; the proposed scheme is presented in Section 3; Section 4 gives configuration of the experiment environment and experimental results; and Finally, Section 5 concludes this article.

II. RELATED WORKS

DDoS attacks use many controlled zombie computers to attack targets to achieve the purpose of damaging host's network services and communication[3-7]. Furthermore, to deny normal users of services from the servers, many network attackers can use a large number of zombie computers to launch bandwidth consumption attacks or resource starvation attacks. Both attacks can effectively paralyze the target. To achieve attack purpose, first the attackers hijack a large number of hosts and install malicious program. Some DDoS attack tools apply multi-layer structure to control thousands of computers. In this way, they generate a great deal of network traffic to paralyze the targeted host.

DDoS attacks can be divided into three layers: the attacker, handler, and agent, each of them plays a different role in the attack, as shown in Fig. 1.

a) Attacker layer: attackers do attacks through the Internet and they can be activities anywhere on the network, even on a notebook computer. The attackers send attack commands to handlers.Handler layer: handlers have been installed with particular programs by the attackers to control multiple agents. The handlers can forward commands from attackers to agents.

c) Agent layer: Agents are also invaded and controlled by the attackers. When attack programs run on the agents, they cause the agents to receive and run the commands from the handlers. The agents are the real attackers in terms of victims.

Lin et al. proposed to use priority queue (PQ)[2] to alleviate malicious packet flows. The scheme has two queues: a high priority queue and a low priority queue. The packets coming from normal sources are put in the high priority queue and obtain services. On the other hand, if

978-0-7695-4372-7/11 $26.00 © 2011 IEEE
DOI 10.1109/IMIS.2011.53

the packets are coming from suspicious sources, they will be put in the low priority queue and wait for server response until the high priority queue is empty. They exploit harmonic mean to analyze throughputs of normal and DDoS nodes. If the value of harmonic mean is over some threshold value, the source will be treated as suspicious.

III. Proposed Scheme

In this section, we describe our propose DCPQ scheme.

A. Concept of priority queue

Based on traffic of received packets and the interval of arrival time, we determine the user is normal or malicious and the level of priority for coming packets. In this way, the request packets from normal users will be sent to the high priority queue and get services immediately; and packets from malicious users will be sent to the low priority and are processed after the high priority queue is empty. If the low priority queue is full, the coming low priority packets will be dropped. To drop packets, we adopt DropTail queue management so that if the queue exceeds its maximum capacity, coming packets cannot enter the queue and are dropped.

B. Structure of DCPQ Scheme

In this paper, we propose to use a Double Check Priority Queue (DCPQ). First, we analyze traffic to determine whether the source is secure or not. Next we check the interval of arrival time of packets. If packets pass the double check, they are considered as normal and are sent to the high priority queue and get normal services from the server. By this mechanism, normal clients can acquire very stable and smooth services.

The main advantages of DCPQ over the original PQ scheme[2] is that by including the packet traffic into queue analysis, possibility of wrong decisions is reduced and time of decision is also reduced via the double check scheme. Because many suspicious packets can be found in the process of traffic analysis, the second check based on time interval is bypassed most of time.

Fig. 2 shows the flow chart of the DCPQ scheme. When receiving packets, the server will confirm if it has ever received packets from the same source. If the answer is no, it will establish a new data structure including the source address, arrival time, traffic, etc. and saves them in the database. Then it sends packets to the high priority queue. If the answer is yes and the recorded source is suspicious, it directly puts these coming packets in the low priority queue and updates traffic record. If the answer is yes and the recorded source is normal, then it starts harmonic mean analysis that computing average of packets arrival time and used to compare with a threshold value. If the value of the harmonic mean is greater than the threshold, coming packets will be put in the low priority queue; otherwise, they will be put in the high priority queue.

Our scheme will use the following elements and functions:

a) *Database:*It records information of each source.

b) Source analysis function: It decides if the source is normal or suspicious.

c) *Analysis of packet interval function:It decides if coming packets are normal or suspicious.*

d) *Traffic control unit:According to activities of the high priority queue it dynamically allocates resources to packets in the low priority queue.*

e) *Management of queues:*The normal users will be placed in the high priority queue, while the suspicious users will be placed in the low priority queue. The server processes requests of users according to their priorities.

The database records information for each source including the following items:

- Source address
- The arrival time of coming packets arriving at the server
- The previous packet arrival rate
- The average time of coming packets
- Traffic record
- The security level

Base on the DCPQ scheme, we use traffic analysis to do the first check, and if needed, we compute the harmonic mean of coming packets to do the second check in order to prevent DDoS attacks.

C. Traffic Analysis Mechanism

We assume that the attacker sends a large number of packets to paralyze the server. To defend the attack, we design a way that the server will response to clients low traffic first. In this way, the normal users will get stable and smooth services.

We record the traffic of transmission from each client, and store these data in the database. At a preset time interval we analyze the data to determine if packets are coming from suspicious sources. It would consume too much resource if the preset time interval is too short, on the other hand, it would slow down the reaction if the preset time interval is too long.

For traffic analysis, we use the following formula:

$$T = C\frac{b}{n} \qquad (1)$$

where T: the threshold used in traffic analysis

b: the server bandwidth

n: the maximum number of current users

C: an adjustable constant

If we let X_i be traffic from source i, then we have a total traffic: $X_1 + X_2 + ... + X_n \le b$, where $X_1 \le X_2 \le ... \le X_n$.

The adjustable constant C is set to be between 1.5 and 10. When the bandwidth of the server and the total flow rate are close, C is about 5. The greater the difference in traffic and bandwidth, the faster C is changed. When the server has excess bandwidth, C is increased, as a result, the threshold becomes loose; when the server is very busy, i.e. it has heavy traffic, C is decreased and the threshold becomes strict. The constant C is adjusted as the following two formulas:

- The server with excess bandwidth :

$$C = 5\sqrt{\left(1 - \frac{total\ traffic}{b}\right)} + 5 \qquad (2)$$

- The server is busy:

$$C = 3.5 \times \frac{b}{total\ traffic} + 1.5 \qquad (3)$$

After the threshold value being computed, traffic from all sources is compared with the threshold value: if it is greater than the threshold value, the source is marked as a suspicious source; otherwise, if it is less than the threshold value, the source is marked as a normal source.

We dynamically adjust the C value in two ranges of time: a short time interval of every 10 to 20 seconds or a long time interval of every 2 to 6 seconds. If the computed C value compared with the previous one is more than 10%, the total traffic is abruptly changed so that we compute the next C value using the short time interval. If the computed the next C value is less than 10%, the total traffic is steady , and in order to save resources we compute the next C value using the long time interval. When using the short time interval to adjust C, we observe the total traffic every second, and if more than 50% change in traffic is observed, we set to compute the new C value immediately.

D. Harmonic Mean Mechanism

In order to distinguish suspicious packets from normal packets, we use the harmonic mean (VHM)[2] to do traffic statistics by using the following equations:

$$H_{t12}(t) = \frac{2}{\frac{1}{t_1} + \frac{1}{t_2}} \qquad (4)$$

$$H_{t23}(t) = \frac{2}{\frac{1}{t_2} + \frac{1}{t_3}} \qquad (5)$$

$$H_{avg\_diff} = H_{t23}(t) - H_{t12}(t) \qquad (6)$$

By equation (4), we compute the harmonic mean of traffic with packets arriving at time t_1 and time t_2. Similarly, by equation (5), we compute the next harmonic mean of traffic with packets arriving at time t_2 and time t_3. By equation (6), we obtain the difference of these two harmonic means.

The difference of harmonic means is used to observe the rate of the arriving packets. When the arriving rate of packets is higher than threshold value, the system is likely under attack. And these suspicious packets will be put in the low priority queue.

IV. Experimental Setting and Results

Fig. 3 shows our experimental network topology. The number of attack nodes and normal clients are not fixed. The packets from normal users and attackers are delivered to the server through a router. The priority queue unit is built on the path between the router and the server.

We performed four experiments to prove the DCPQ scheme is useful to defend DDoS attacks. Moreover, we will compare the results of DCPQ scheme with the PQ scheme to prove that the DCPQ scheme is more effective than the PQ scheme.In our experiments, we used the Poisson distribution to model the network traffic.

A. Experiment 1

In Experiment 1 we aimed to find a suitable value for the threshold. We used the Network Simulator (NS-2) as the simulation tool. The experiment parameters are listed in Table 1. We set ten normal user nodes, each with a bandwidth of 100 Kbps, and ten attack nodes, each with bandwidth of 1000 Kbps. All of the normal user nodes send packets at 0 second of the experiment. The total experimental time is fifty seconds. The maximum size of the PQ is 10. The bandwidth of the path between the router and the server is 1 Mbps.

Table I Setup of Experiment 1

	Normal User	***Attacker***
Number	10	10
Start Time	0 second	0 second
End Time	50 second	50 second
Packets Rate	100 Kbps	1000 Kbps

As shown in Fig. 4, when the threshold value is 0.1, the throughput of normal users is low. As the threshold value is decreased from 0.1 to 0.02, throughput of normal nodes packets increase because packets from more DDoS nodes are identified and put into the low priority queue. We find that the optimal threshold value for this

experiment setting is 0.02 with largest throughput of normal nodes packet, 4205 Kb. When threshold value is lower than 0.009, most malicious packets are treated as normal packets so that the priority queue loses most defensive capability, as a result, throughput of the normal nodes is significantly reduced.

Fig. 5 shows throughput of DDoS and normal nodes by using the DCPQ scheme. Comparing results of the DCPQ scheme with the PQ scheme, we find the queue performance improves 5% when the threshold is set to the optimal value, 0.02. Furthermore, at each threshold value, throughput of the normal nodes is very good and steady without any sharp plunge.

B. *Experiment 2*

In Experiment 2 we compared queue performance of the DCPQ with the DropTail queue. The experiment parameters are listed in Table 2. The threshold value is set as 0.02.

Fig. 6 shows throughput of normal nodes using DropTail queue scheme. We find that under DDoS attacks, which attack at 20 second of the simulation time, throughput of each normal node degrades seriously and packets from normal users can be hardly received by the server.

Table II Setup of Experiment 2

	Normal User	*Attacker*
Number	10	10
Start Time	0 second	20 second
End Time	50 second	50 second
Packets Rate	100 Kbps	1000 Kbps

Fig. 7 shows throughput of normal nodes using the DCPQ scheme. We find that under DDoS attacks, which attack at 20 second of the simulation time, throughput of each normal node does not degrade much. From Fig. 8 and Fig. 9, we conclude that DCPQ scheme is effective in defending DDoS attacks, but DropTail queue scheme is not.

C. *Experiment 3*

In Experiment 3 we compared queue performance of DCPQ, PQ, and DropTail queue schemes and checked if they were able to maintain normal service when facing a great amount of DDoS attacks.

The experiment parameters are listed in Table 3. Note that the number of attackers is varied from 0 to 500. Other parameters are the same as setup of Experiment 1.

Fig. 8 shows throughput of DDoS and normal nodes using DCPQ, PQ, and DropTail queue scheme under various numbers of DDoS attack nodes. The curve of throughput of normal nodes using DropTail queue scheme drop sharply and immediately as the number of attacks increases from 1. If the number of attacker is 20, throughput of normal nodes using DropTail queue scheme value is almost 0 Kb. When the number of attackers increases from 0 to 500, the throughput of normal nodes using PQ scheme is slightly reduced, while the throughput of normal nodes using DCPQ scheme is almost fixed. We conclude that among these three schemes, DCPQ is most effective against large amount of DDoS attacks.

Table III Setup of Experiment 3

	Normal User	*Attacker*
Number	10	0-500
Start Time	0 second	0 second
End Time	50 second	50 second
Packets Rate	100 Kbps	1000 Kbps

D. *Experiment 4*

In Experiment 4 we compared performance of DCPQ, PQ, and DropTail queue schemed and checked if they were able to maintain service when facing attacks with different packets rates.

The experiment parameters are listed in Table 4. Note that the packets rate of attackers is varied from 0 to 900 Kbps. Other parameters are the same as setup of Experiment 1.

Fig. 9 shows throughput of normal nodes using DCPQ, PQ, and DropTail schemes against attacks with various DDoS packets rates from 0 to 900 Kbps. As shown in the figure, throughput of normal nodes using DropTail queue scheme decreases continuously as the DDoS packets rate increases. The throughput of normal nodes using PQ scheme decreases when the DDoS packets rate increases from 0 to 300 but the throughput begins to increase when DDoS packets rate is 400 Kbps, that is, when the PQ scheme starts to recognize malicious packets from DDoS nodes and put them in the low priority queue. The DCPQ scheme gives best performance. Throughput of normal nodes using this scheme dips a little when the packets rate of DDoS nodes decreases from 0 to 100 Kbps, but when the DDoS packets rate is over 100 Kbps, throughput of normal node increases and always maintains a highest value among these three queue schemes.

V. CONCLUSION

From the experimental results, the DCPQ scheme is found to be very effective against DDoS attacks. Its performance is proven to be better than the PQ and DropTail schemes. Since it is not possible to completely prevent networks from attacks, we can only minimize damage caused by DDoS attacks. The DCPQ scheme can be used to efficiently alleviate effect due to various numbers of DDoS attacks with various packet rates and maintain quality of service for normal users.

Table IV Setup of Experiment 4

	Normal User	*Attacker*
Number	10	10
Start Time	0 second	0 second
End Time	50 second	50 second
Packets Rate	100 Kbps	0~900 Kbps

ACKNOWLEDGEMENT

This work was supposed in part by Taiwan National Science Council under grants NSC 99-2221-E-029-034-MY3.

REFERENCES

[1] Aleksandar Kuzmanovic, and Edward W. Knightly, "Low-rate TCP-targeted denial of service attacks: the shrew vs. the mice and elephants," Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications, 2003, pp.75-86.

[2] Chu-Hsing Lin, Jung-Chun Liu, Fuu-Cheng Jiang, and Chien-Ting Kuo, "An Effective Priority Queue-Based Scheme to Alleviate Malicious Packet Flows from Distributed DoS Attacks," International Conference on Intelligent Information Hiding and Multimedia Signal Processing, IIHMSP '08, August 15-17, 2008, pp.1371-1374.

[3] Kihong Park, and Heejo Lee, "On the Effectiveness of Route-Based Packet Filtering for Distributed DoS Attack Prevention in Power-Law Internets," Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications, August 27-31, 2001, pp.15-26.

[4] Y. Chen, Y. -K. Kwok, and K. Hwang, "MAFIC: Adaptive Packet Dropping for Cutting Malicious Flows to Push Back DDoS Attacks," Proceedings of 25th IEEE Int' Conf. Distributed Computing Systems Workshops 2005, June 2005.

[5] K. Park and H. Lee, "On the Effectiveness of Probabilistic Packet Marking for IP Traceback under Denial of Service Attacks," Proceedings of IEEE INFOCOM 2001, Mar. 2001.

[6] Zhaoyang Qu, Chunfeng Huang, and Ningning Liu, "A Novel Two-Step Traceback Scheme for DDoS Attacks," Second International Symposium on Intelligent Information Technology Application, December 20-22, 2008, pp.879-883.

[7] Alex C. Snoeren, "Hash-based IP traceback," Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications, August 27-31, 2001, pp.3-14 .

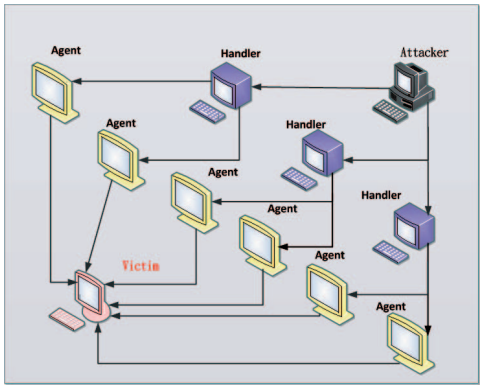

Fig. 1 Layers of DDoS attacks.

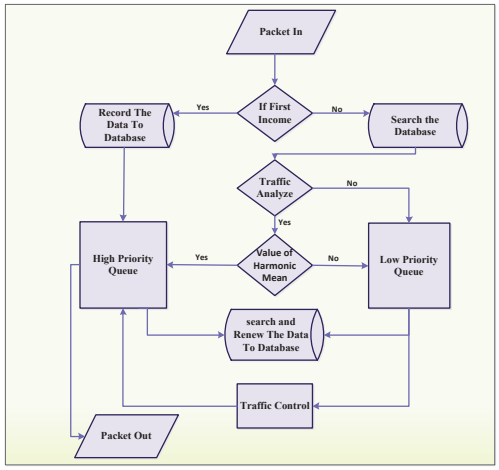

Fig. 2 Flow chart of DCPQ scheme.

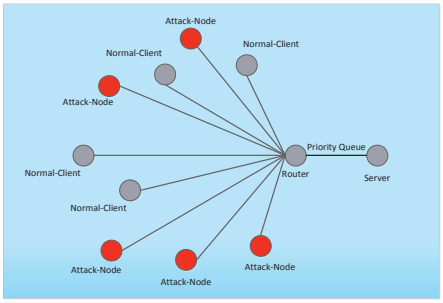

Fig. 3 Network topology.

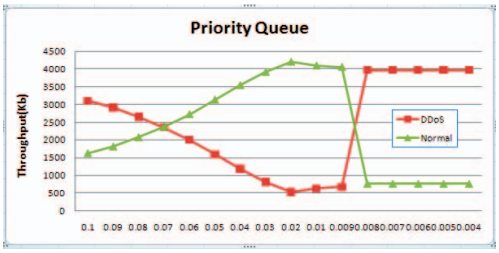

Fig. 4 Throughput of DDoS and normal nodes of PQ scheme with various thresholds.

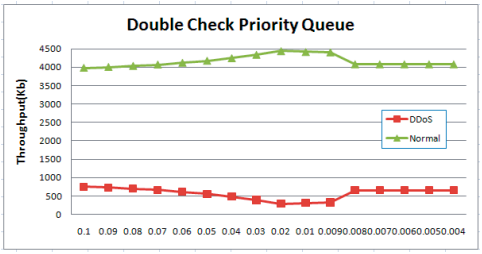

Fig. 5 Throughput of DDoS and normal nodes of DCPQ with various thresholds

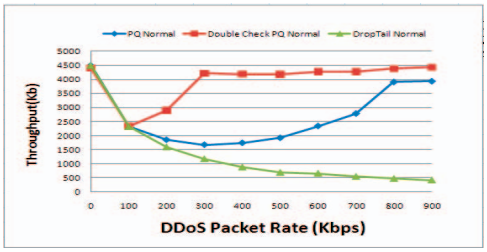

Fig. 9 Throughput of normal nodes using DCPQ, PQ, and DropTail schemes against attacks with various packets rates.

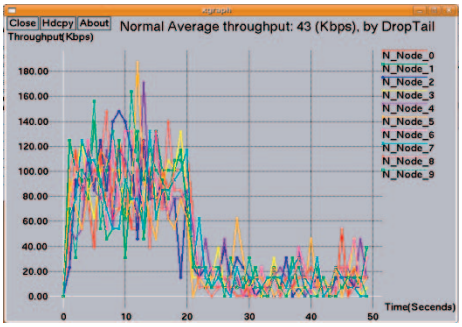

Fig. 6 Throughput of normal nodes using the DropTail queue.

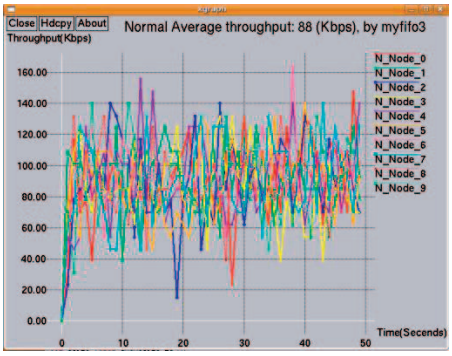

Fig. 7 Throughput of normal nodes using the DCPQ scheme.

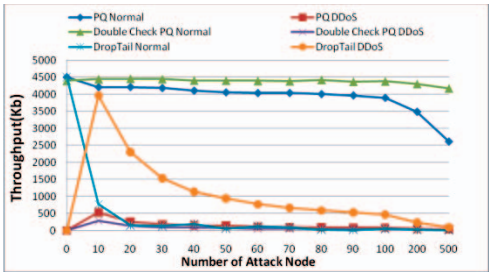

Fig. 8 Through of DDoS and normal nodes using DCPQ, PQ, and DropTail queue scheme under various attacks.