Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2010, Article ID 570940, 15 pages
doi:10.1155/2010/570940

Research Article

An Adaptive Approach for Defending against DDoS Attacks

Muhai Li[1,2] and Ming Li[1]

[1] *School of Information Science & Technology, East China Normal University, no. 500 Dong-Chuan Road, Shanghai 200241, China*
[2] *Department of Computer Science, Zaozhuang University, Shandong 277160, China*

Correspondence should be addressed to Muhai Li, muhaili@126.com

Received 4 February 2010; Accepted 13 March 2010

Academic Editor: Cristian Toma

In various network attacks, the Distributed Denial-of-Service (DDoS) attack is a severe threat. In order to deal with this kind of attack in time, it is necessary to establish a special type of defense system to change strategy dynamically against attacks. In this paper, we introduce an adaptive approach, which is used for defending against DDoS attacks, based on normal traffic analysis. The approach can check DDoS attacks and adaptively adjust its configurations according to the network condition and attack severity. In order to insure the common users to visit the victim server that is being attacked, we provide a nonlinear traffic control formula for the system. Our simulation test indicates that the nonlinear control approach can prevent the malicious attack packets effectively while making legitimate traffic flows arrive at the victim.

1. Introduction

DDoS attacks have been one of the most hazardous threats on the Internet [1–6]. The attacks generate enormous packets by a large number of agents and can easily exhaust the computing and communication resources of a victim within a short period. As a result, it makes a victim deny normal services in the Internet.

Many defense and response mechanisms have been suggested in literature about DDoS attacks. Mirkovic and Reiher [3] presented a comprehensive taxonomy of DDoS attacks and defense mechanisms. Many DDoS detection approaches, such as "IP traceback" [7, 8], "traffic statistic" [8–13], "pushback" [14, 15], "packet filtering" [8, 16–18] and "wavelet analysis" [12, 19–23], "Hurst parameter" [24], and so forth, try to find the identities of real attack sources and defend against attacks. However, these methods can only find out attacks. They cannot drop attack packets adaptively.

The literatures above show that if we expect to prevent DDoS attacks significantly two critical issues must be handled first [6]: (1) accurately identifying the machines participating in forwarding malicious flows and (2) incisively cutting off the malicious flows at those machines.

Hussain et al. [1] presented a framework to classify DDoS attacks into single-source and multi-source attacks. However, these methods cannot be used directly to restrain DDoS attack traffic. In order to detect and filter attack packets at the victim end, Kim et al. [16] provided a general anomaly detection framework. Jin et al. [25] provided a concrete "Hop-Count Filtering" algorithm to filter out spoofed attack packets based on packets' TTL (Time-To-Live) values.

Zou et al. [15] and Lee et al. [26] considered various cost factors, including false positive/negative cost, in the process of developing Intrusion Detection System (IDS). However, they employed a static system design method, which does not take how to dynamically adjust an IDS's configurations into consideration according to the attack condition. In [26], Lee et al. explored the adaptive defense principle, but this principle also gets a problem: how can one compute the probabilities of false positive and false negative under attacks?

There is another problem in [26]. Due to the limit of routers memory, routers can only save the packet information for a short time. Thus, an attack process at the routers must be identified in time. However, when the victim is under severe attacks, it may have no ability to send the warning of attacks. At the same time, the link, which the victim communicates with its upper router, may become so congested that it is not capable of sending attack message to its upper routers in time. Obviously, under this circumstance, the adaptive defense system cannot work normally. In order to solve the problem, we add a special application server (shown in Figure 1), which can not only save and analyze a lot of information the victim sends, but also instruct the upper routers of the victim to control traffic.

Most researches have focused on stationary network operation with fixed configurations. However, in reality, attack detection systems have to face rapidly changing network conditions and various attack intensities [15]. Therefore, apart from finding a good detection algorithm, it is equally or more important to design an "intelligent" defense system that can automatically adjust its detection and filtering parameters to achieve the best performance under every possible attack situation.

We introduce an "adaptive defense principle" based on "normal traffic"—a defense system which can adaptively adjust its configurations according to network conditions and "attack severity." The "normal traffic" refers to the traffic that the victim gets under no attacks. We call such a defense system an "adaptive defense system."

Compared to the traditional nonadaptive defense system, the adaptive defense system can not only find out attacks quickly, but also drop attack traffic and protect a victim or server more vigorously under severe attacks.

Let $y(t)$, $n(t)$, and $N(t)$ be total traffic, normal traffic, and statistic traffic at time t, respectively. The attack traffic, which is generated by attackers, is denoted by $a(t)$.

Then $y(t)$ can be abstractly expressed by

$$y(t) = n(t) + a(t). \tag{1.1}$$

Obviously, if a server is not under attacks, $a(t)$ equal to zero and $y(t)$ equal to $n(t)$. If a victim is under attacks, because of a large number of attack traffic from attackers, the value

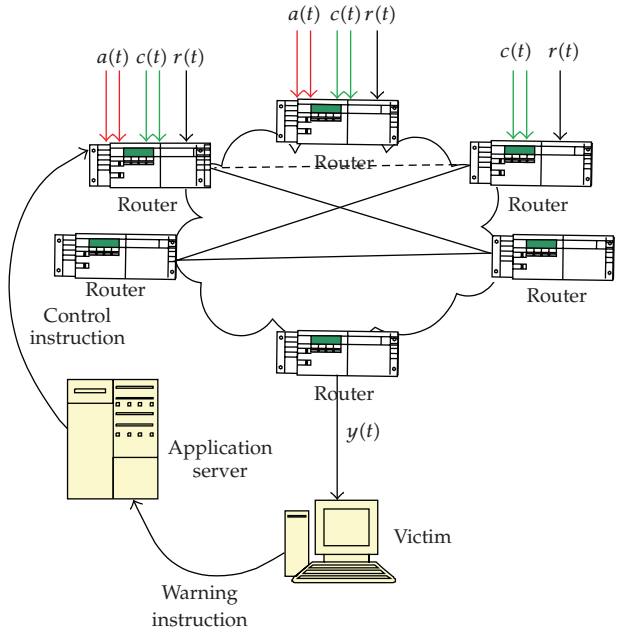

Figure 1: A network topology of adaptive defense system.

of $a(t)$ will rapidly increase to a high level. Therefore, if we can capture the value of $a(t)$ during detection, it should be very easy to indentify attacks. Unfortunately, we have no way to get the value of $a(t)$ directly. However, $y(t)$ can be captured with sniffer software [27]. According to (1.1), if we can obtain the value of $n(t)$, then the aforementioned problem can be solved easily. Note that $n(t)$ is yet unknown during detection. Studies in [11, 12] show that $n(t)$ under attacks can be substituted by statistic traffic $N(t)$ under no attacks.

In order to make a victim possess the ability to provide normal service under attacks, we divide normal traffic into two parts. One part is named common traffic, denoted by $c(t)$, which is created by common users. The other is generated by random users and we denote it as $r(t)$. Thus, $n(t)$ can be represented by

$$n(t) = c(t) + r(t). \tag{1.2}$$

Because the start time of attack and attack traffic under attacks are unknown exactly, one may encounter difficulties in finding a method to recognize attack traffic from total network traffic. In order to remove the attack traffic, we propose an adaptive drop-packet approach:

$$y(t) = u(t)^{j(t)}c(t) + v(t)^{k(t)}[a(t) + r(t)], \tag{1.3}$$

where $j(t)$ and $k(t)$ are functions of t. In general, they are integers related with t. The functions of $u(t)$ and $v(t)$, which range from 0 to 1, are real and related with $y(t)$.

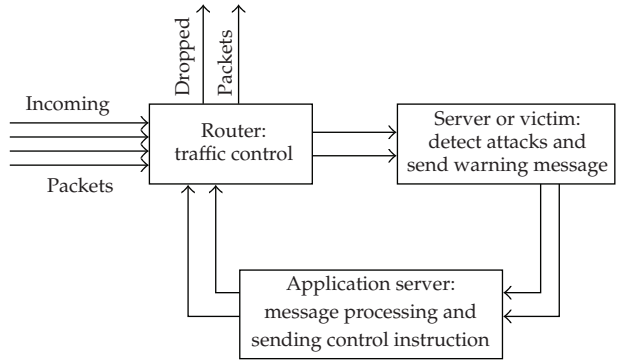

Figure 2: The working procedure of adaptive defense system.

We present concrete adaptive defense system for defending against DDoS attacks. The system does not depend on attack types, so it can be used widely in the network security. The working procedure of the system is as shown in Figure 2.

The rest of this paper is organized as follows. Section 2 introduces the concept of adaptive controller. Section 3 contains the design of our proposed adaptive control traffic algorithm. In Section 4, we give a simulation test. Section 5 concludes with some final remarks and suggestions for future work.

2. Adaptive Control

Intuitively, an adaptive controller is such a controller that can modify its behavior in response to changes in the dynamics of the process and the character of the disturbances [28]. Since ordinary feedback also attempts to reduce the effects of disturbances and difference between feedback control and plant uncertainty, the problem of the adaptive control immediately arises.

Most network systems are very complex and unintelligible; it is neither possible nor economical to make a thorough investigation of the causes of the process variation. Adaptive controllers can be a good alternative in such cases. In other situations, some of the dynamics may be well understood, but other parts are unknown. For example, the network system, which is composed of many devices such as computers, routers, servers and so forth, does not change in some period, but the traffic that is created by these computers does change continuously at the same time. In such cases, it is of great importance to use the prior knowledge and estimate and adapt to the unknown part of the process.

In general, there are four types of adaptive systems: gain scheduling, model-reference adaptive control, self-tuning regulators, and dual control [28]. The block diagram of an adaptive system with gain scheduling is shown in Figure 3.

The adaptive system will become nonlinear because of the parameter adjustment mechanism. Since general nonlinear systems are difficult to deal with, we give a very special system that belongs to a special class of nonlinear systems. The system has two loops: one is a normal feedback with the process and the controller. The other is the controller-parameter adjustment loop based on the operating conditions.

The model of gain scheduling can satisfy our requirement to control traffic because the scheme is originally used to measure the gain of traffic at the victim. The system will change

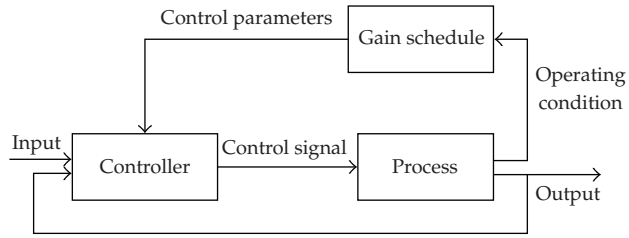

Figure 3: A diagram of adaptive system with gain schedule.

the control parameters of a router based on the gain in an application server. The controller, namely, router, compensates for changes in the process gain. In Figure 3, "Input" represents the incoming traffic of routers; that is to say, $c(t) + r(t) + a(t)$. "Output" denotes the receiving traffic of a victim, $y(t)$. Obviously, the system is adaptive.

The system we proposed to control traffic is different from the common adaptive system because we consider that the traffic of a victim can be controlled in normal state. This is to say, we must ensure that a victim can maintain the normal service for its common users whether attacks exist or not. In order to achieve the destination, we classify the victim's users into two types based on the times of visiting victim in some period. One type is for common users who usually visit the site, and the other is the type of noncommon ones.

The $c(t)$ in "Input" is right the traffic of common users, and the $r(t)$ and $a(t)$ present the noncommon traffic. Because we want to protect a victim from attacks, when the routers control traffic, we will try our best to hold the traffic of common users, namely, $c(t)$. At the same time, we cut the other traffic off as much as possible.

When a victim is under attacks, the attacker often controls thousands of zombies to send rubbish packets to a victim. The attack traffic of a victim may increase quickly. In general, the attack traffic increases in geometric progression at the beginning of attacks. So we use exponent coefficient $u(t)^{j(t)}$ and $v(t)^{k(t)}$ to limit the increase of attack traffic, where $j(t)$ and $k(t)$ usually equal to t. $u(t)$ and $v(t)$, which range between 0 and 1, are functions related with $y(t)$. Therefore, we get an adaptive control system described with formula (1.1).

Obviously, when the values of $u(t)$ and $v(t)$ are 1, the traffic is not controlled. In fact, the packet-dropping rate of common users can be computed by $1-u(t)^{j(t)}$ at time t. One of the other users is $1 - v(t)^{k(t)}$. In order to make the victim provide normal service under attacks, the $u(t)^{j(t)}$ and $v(t)^{k(t)}$ in (1.1) will meet the following relation: $u(t)^{j(t)} \gg v(t)^{k(t)}$, where t is a time variable.

In order to discuss the efficiency of the adaptive control system, we often simplify formula (1.1) with the following ways. Let $u(t)^{j(t)}$ equal $b * v(t)^{k(t)}$, where b can equal $2, 3, 4$, and so on. Let $j(t)$ and $k(t)$ equal t. Since $v(t)$ is a const that relates with $y(t)$, the greater the value of $y(t)$ exceeds the normal value itself, the less the values of $v(t)$. We often set the values of $v(t)$ equal to 0.9 at first. Then formula (1.1) can be simplified as follows:

$$y(t) = [b * c(t) + r(t) + a(t)]v^t. \tag{2.1}$$

Since $c(t)$, $r(t)$ and $a(t)$ are bounded on time space, there exists a value T, which meets the relation

$$b * c(t) + r(t) + a(t) \leq T \quad (\forall t) \tag{2.2}$$

we want to control traffic between *vma* and *vmi*. Let the average value of them, namely, $(vma+vmi)/2$, be satisfying traffic that is controlled by the controller, where *vma* and *vmi* are defined in Section 3.1. Let (2.2) substitute into (2.1), we have

$$\frac{(vma + vmi)}{2} \leq Tv^t. \tag{2.3}$$

So, after $\log_v^{(vma+vmi)/2*T}$ unit times, the traffic can be in normal state. In actual application, the victim's devices decide the values of *vma* and *vmi*. T is an experience value. It is worthwhile to note that, when a victim is under attacks, the value of $a(t)$ is far greater than the sum of $b * c(t)$ and $r(t)$. Therefore, we can set the value of v according to the experience value of attack intensity. The adaptive algorithm of control traffic can be found in Section 3.

3. Adaptive Approach for Traffic Control

In order to make a victim send attacked message to its upper routers in time, we propose an adaptive defense system with an application server.

Firstly, we find out attack events by the traffic analysis of the victim, if attacks exist for the victim. After that, the victim will send a warning or attack message to application server at once.

Secondly, on one hand, when the application server receives the attack-warning message from the victim, the timer of it will start automatically. Once the value of the timer goes over the time threshold, or the application server receives attack message, the application server sends control instruction to routers immediately. On the other hand, when the application server receives recovery-warning message, it will send a control instruction for recovering traffic control.

Thirdly, the upper routers of the victim will set their parameters to control traffic after they receive warning instruction.

3.1. Traffic Analysis and Warning Message

In [11, 12], we propose a method for detecting attacks, and we should employ this method in this paper.

In control traffic algorithm, we use formula (1.1) to control traffic of network system. On one hand, when the values of $u(t)$ and $v(t)$ are less than 1 and attack traffic is in a stable situation, the traffic $y(t)$ will decrease very quickly (shown in Figure 7 about 970 ms). Therefore, we give a minimum traffic, denoted by *vmi*. When the traffic $y(t)$ is less than *vmi*, the system will send to application server a recovery instruction. On the other hand, When the values of $u(t)$ and $v(t)$ equal 1 and attacks exist, the traffic $y(t)$ increases rapidly. In order to make the victim provide the normal service, we define a maximum value, denoted by *wma*, which the victim can endure. When the total traffic of the victim is greater than *wma*,

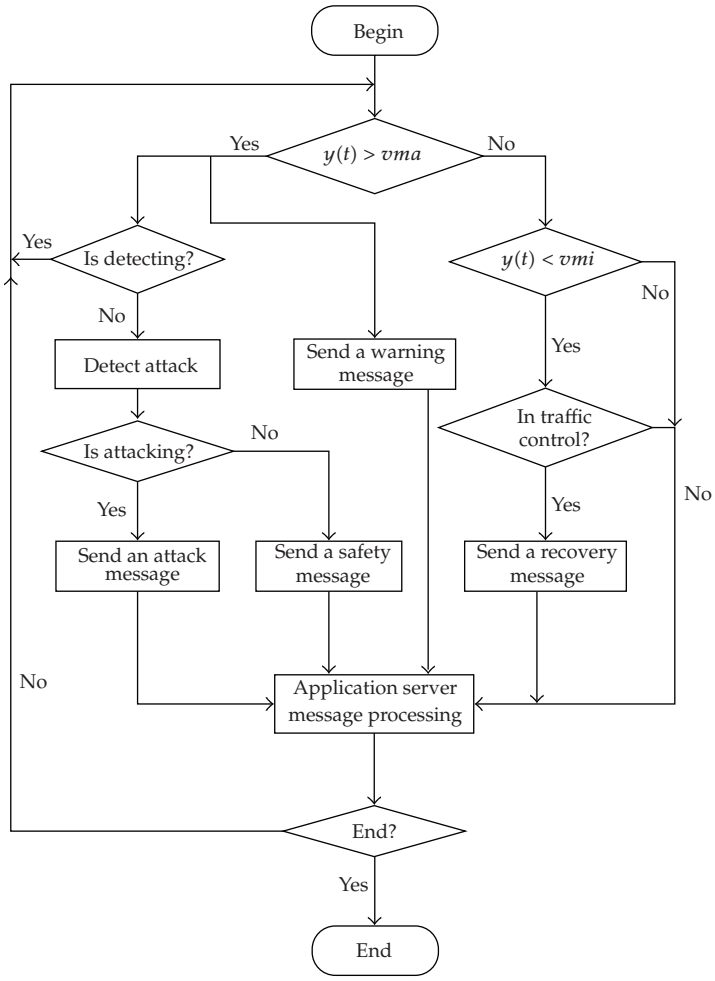

Figure 4: The flow chart of attack detection and message processing in the victim.

our adaptive defense system will start attack detection and send an attack-warning message automatically. If attacks are found after detection, the victim will also give an attack message to application server at once.

The algorithm for detecting attack and sending warning message is as shown in Figure 4.

3.2. Application Server and Message Processing

Due to the limit of router memory and functions, it is very difficult to restore and process a lot of information of packets. The application server can receive and process the information. In our system, the application server, which is specially used to do the task, is not affected by network congestion.

There is a timer in the application server. With the timer, it is convenient to send routers control information when the victim is under severe attacks and has no ability to inform the

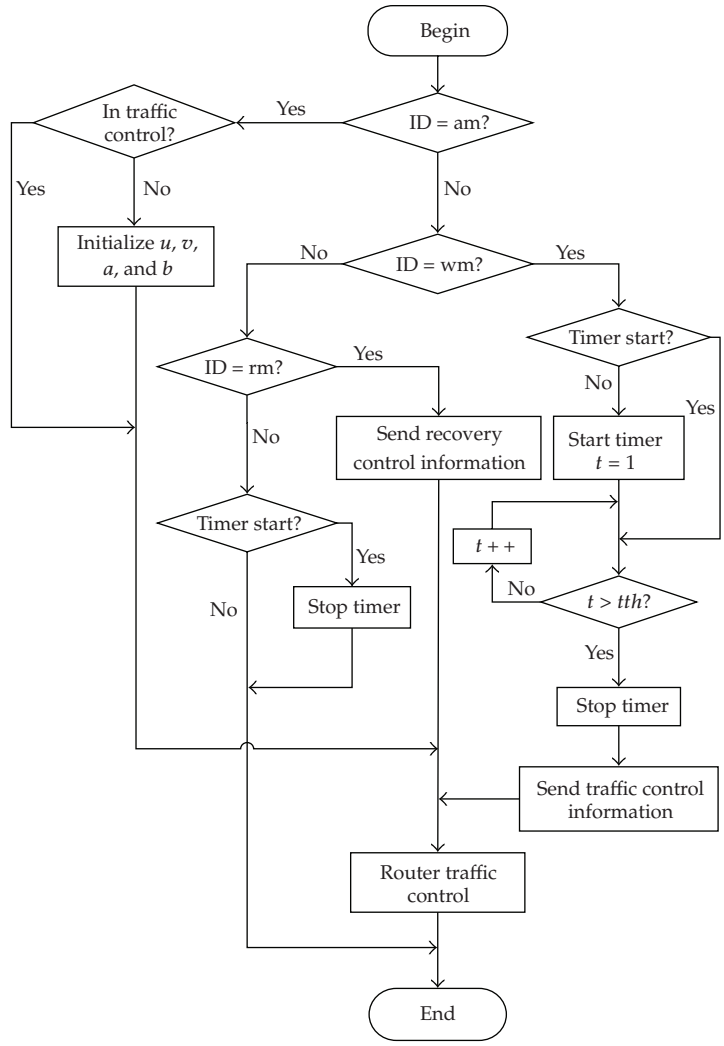

Figure 5: The application server processes messages from the victim.

application server. Once the server receives an attack-warning message, its timer will start. Then it periodically tests whether the value of the timer is greater than a time threshold, denoted by *tth*, which is a const. If the result is true, the application server will send traffic control information to its upper routers.

In addition, there are many states to be judged in the server. For example, during traffic control, if the server receives an attack-warning message again, it will send a message to routers so that the routers can do traffic control further. The detailed processing procedure can be seen from Figure 5.

In Figure 5, we define a variable id, which can receive some identifiers from the victim, such as "wm", "am", "rm", "sm", and so forth. Let "wm", "am", "rm", and "sm" be "warning message", "attack message", "safety message" and "recovery message", respectively.

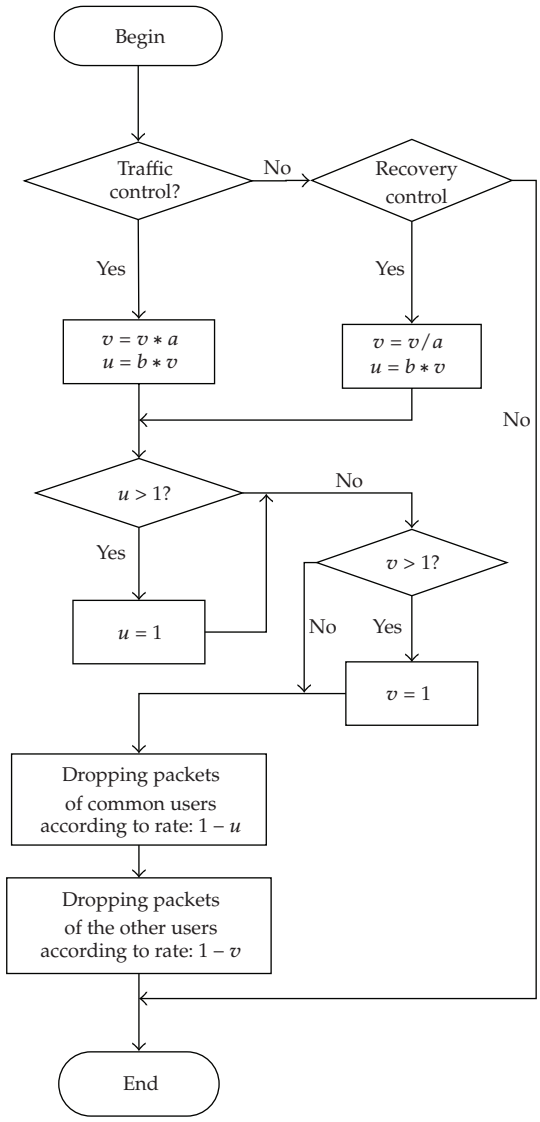

Figure 6: The traffic control in a router.

3.3. Router Traffic Control

As for fighting against DDoS attacks, it is very important to control traffic by router.

An adaptive defense DDoS attacks algorithm faces two major challenges in identifying attack flows: source IP spoofing and multiple distributed zombies' utilization. There are two extreme cases of IP spoofing. On one hand, all sources of IP addresses are illegal or unreachable. On the other hand, all the attack packets carry legitimate source IP addresses. Note that "legitimate" IP address is forged, and it does not represent the true IP addresses of the computer that sends the attack packets. Therefore, we cannot regard all "legitimate" IP address as safe ones. In our algorithm, a router, which is viewed as controller, maybe receives two types of information. One is used for traffic control. The other for recovery control.

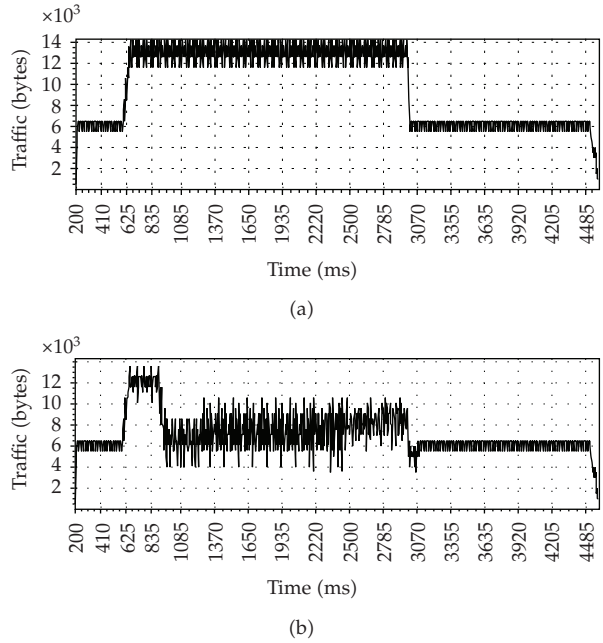

Figure 7: (a) The traffic chart without adaptive control in a victim. (b) The traffic chart under adaptive control in a victim.

Figure 6 shows the procedure of traffic control by routers. On one hand, when a router receives the instruction of traffic control, the $u(t)$ and $v(t)$ can be computed iteratively according to the formula

$$v = a * v, \qquad u = b * v. \tag{3.1}$$

On the other hand, if a router receives instruction for recovery traffic, we do not set the values of $u(t)$ and $v(t)$ at once; instead we use the formula as follows:

$$v = \frac{v}{a}, \qquad u = b * v. \tag{3.2}$$

The reason we use formula (3.2) to recover traffic is that the traffic is very large at the beginning of recovery. The method can help to increases the values of $u(t)$ and $v(t)$ bit by bit. It can ensure that a victim works in normal state.

In formulas (3.1) and (3.2), "a" represents a factor used for traffic control and "b" is used for adjusting the control rate of common traffic, in order to let a victim provide more services for common users as best as possible.

4. Simulation Test

We have simulated the adaptive algorithm with ns-2 simulator [29]. There are 10 common users and 4 zombies linking to the victim with a router. The ten common users send packets

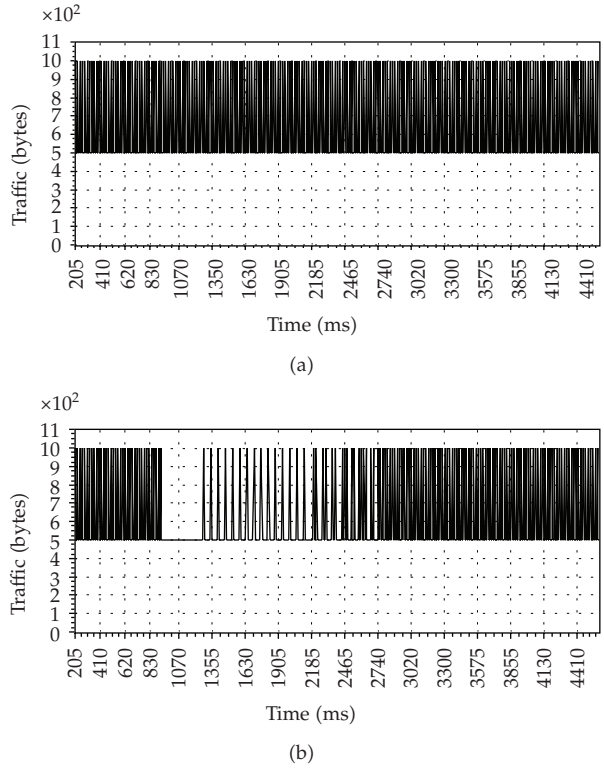

Figure 8: (a) The traffic chart of a common user without adaptive control. (b) The traffic chart of a common user under adaptive control.

with 500 bytes and 3 ms time delay. The other users, regarded as attackers, launch their packets with the random size which ranges from 1000 to 1100 bytes and 2 ms time delay. Both the start time and end time that the common users visit the victim are random values. The start time ranges from 200 to 210 ms, and the end time ranges from 4500 to 4600 ms.

The start time of attacks ranges from 600 to 650 ms, and the end time of them is at 3000 ms.

In detecting attacks, we set the *vma* and *vmi* equal to 8500 bytes and 5000 bytes, respectively. When the traffic of the victim is greater than *vma* or less than *vmi*, our adaptive algorithm will be started to detect the state of the traffic automatically. The detection time lasts for 300 ms. The initial values of a, b, u, and v are 0.9, 2, 1, and 1, respectively.

The adaptive algorithm can be seen from Figures 4, 5, and 6.

Figures 7(a) and 7(b) show the changes of traffic under adaptive control. In Figure 7(b), the total traffic is almost controlled under 8500 bytes between 965 ms and 2985 ms. We can see that the data, of which the time ranges from 585 ms to 965 ms, have a few changes because our system detects attacks at that time. In order to prevent the traffic rapidly increasing, our algorithm defaults to cut 10% traffic off during the first detecting attacks.

In addition, during detecting attack, our control algorithm will maintain the rate of traffic control before detection attack. This is the reason why the data at about 2985 ms have evident changes singularly under no attacks. Because when the system is detecting attacks, the traffic at the time is under control.

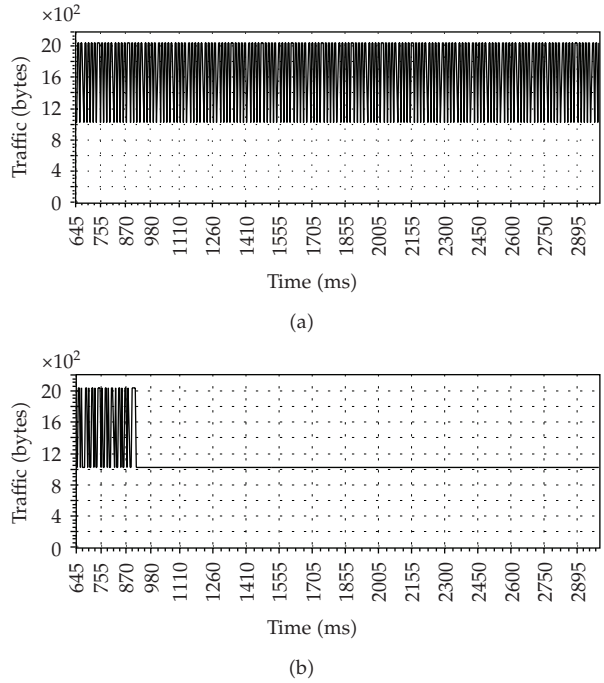

Figure 9: (a) The traffic chart of a noncommon user without adaptive control. (b) The traffic chart of a noncommon user under adaptive control.

Figures 8 and 9 show that the changes of a common user and a noncommon user are under traffic control. On one hand, the victim can maintain service for common users. Figure 8(b) shows the traffic is maintained under attacks. On the other, the traffic of noncommon users can be limited severely; the changes of traffic in Figure 9(b) can show the point clearly.

5. Conclusion

By traffic analysis, we can not only detect attacks, but also defend against DDoS attacks adaptively. Because of network congestion and the limit of router memory, we use a special application server to deal with the problems of information restore, analysis and transmitting. It can make the upper routers receive attack or warning messages from a victim under severe attacks. When the victim is being attacked, we hardly know whether a user is an attacker or not. Hence, it is not easy for us to compute the probabilities of false positive and false negative. The method in this paper does not use these parameters, but use statistic traffic to detect attacks. We give a nonlinear traffic control system based on a general knowledge, namely, stable number of common users who usually visit a server. Therefore, we classify the users of a victim as two types. The common users belong to one type; the other users are another type. In this way, not only can we protect the victim from attacks with our adaptive defense system, but also the victim can provide normal service for common users under DDoS attacks.

In this paper, we distinguish the common users from the others with the times of visiting a victim. In fact, we can classify them according to packet type, packet size, IP address and so on. We can also propose many types of users in order to control traffic as best as one can.

Note that this research is strongly related to the statistics of traffic that is in turn associated with fractal time series. Therefore, we shall further take into account those in traffic engineering and fractal time series; see, for example, [30–44], just naming a few. In addition, tools for analyzing traffic, such as wavelet [45–53], shall be carefully studied. Based on those, we all further explore better method to quickly get effectively statistical features of traffic under no attacks, to study the criterion used for classifying the victim users and the relation between $u(t)$ and $v(t)$, so that we can optimize our algorithm to defend against DDoS attacks.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (NSFC) under the project grant nos. 60573125, 60873264, and 60703112, by the Research and Development Project of Shandong Provincial Education Department (J07WJ29), and the PHD Program Scholarship Fund of ECNU 2009 (2009053).

References

[1] A. Hussain, J. Heidemann, and C. Papadopoulos, "A framework for classifying denial of service attacks," in *Proceedings of the ACM Conference on Internet Measurement (SIGCOMM '03)*, pp. 99–110, Karlsruhe, Germany, August 2003.

[2] K. Hwang, Y. Chen, and H. Liu, "Defending distributed systems against malicious intrusions and network anomalies," in *Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium*, p. 286.1, April 2005.

[3] J. Mirkovic and P. Reiher, "A taxonomy of DDoS attack and DDoS defense mechanisms," *Computer Communication Review*, vol. 34, no. 2, pp. 39–53, 2004.

[4] D. Moore, C. Shannon, D. J. Brown, G. M. Voelker, and S. Savage, "Inferring internet denial-of-service activity," *ACM Transactions on Computer Systems*, vol. 24, no. 2, pp. 115–139, 2006.

[5] "Background on DDoS," 2008, http://www.ddos.com/index.php?content=products/background.html.

[6] Y. Chen, Y. Kwok, and K. Hwang, "MAFIC: adaptive packet dropping for cutting malicious flows to push back DDoS attacks," in *Proceedings of the IEEE International Conference on Distributed Computing Systems Workshops*, pp. 123–129, June 2005.

[7] H. Aljifri, "IP traceback: a new denial-of-service deterrent?" *IEEE Security and Privacy*, vol. 1, no. 3, pp. 24–31, 2003.

[8] S. Savage, D. Wetherall, A. Karlin, and T. Anderson, "Practical network support for IP traceback," *Computer Communication Review*, vol. 30, no. 4, pp. 295–306, 2000.

[9] P. Barford, J. Kline, D. Plonka, and A. Ron, "A signal analysis of network traffic anomalies," in *Proceedings of the 2nd Internet Measurement Workshop (IMW '02)*, pp. 71–82, November 2002.

[10] M. Li, "An approach to reliably identifying signs of DDOS flood attacks based on LRD traffic pattern recognition," *Computers and Security*, vol. 23, no. 7, pp. 549–558, 2004.

[11] M. Li, M. Li, and X. Jiang, "DDoS attacks detection model and its application," *WSEAS Transactions on Computers*, vol. 7, no. 8, pp. 1159–1168, 2008.

[12] M. Li and M. Li, "A new approach for detecting DDoS attacks based on wavelet analysis," in *Proceedings of the 2nd International Congress on Image and Signal Processing (CISP '09)*, October 2009.

[13] Y. Gu, A. McCallum, and D. Towsley, "Detecting anomalies in network traffic using maximum entropy estimation," in *Proceedings of the 5th ACM Conference on Internet Measurement (SIGCOMM '05)*, pp. 345–350, 2005.

[14] M. Cai, Y. Chen, Y. K. Kwok, and K. Hwang, "A scalable set-union counting approach to pushing back DDoS attacks," Tech. Rep. TR-2004-21, USC GridSec, October 2004.
[15] C. C. Zou, N. Duffield, D. Towsley, and W. Gong, "Adaptive defense against various network attacks," US patent no. US7,587,761 b2, September 2009.
[16] Y. Kim, W. C. Lau, M. C. Chuah, and H. J. Chao, "PacketScore: a statistics-based packet filtering scheme against distributed denial-of-service attacks," *IEEE Transactions on Dependable and Secure Computing*, vol. 3, no. 2, pp. 141–155, 2006.
[17] Y.-K. Kwok, R. Tripathi, Y. Chen, and K. Hwang, "HAWK: halting anomalies with weighted choKing to rescue well-behaved TCP sessions from shrew DDoS attacks," in *Proceedings of the 3rd International Conference on Computer Network and Mobile Computing (ICCNMC '05)*, vol. 3619 of *Lecture Notes in Computer Science*, pp. 423–432, February 2005.
[18] K. Park and H. Lee, "On the effectiveness of route-based packet filtering for distributed DoS attack prevention in power-law Internets," in *Proceedings of the ACM Conference on Internet Measurement (SIGCOMM '01)*, pp. 15–26, August 2001.
[19] A. Dainotti, A. Pescapé, and G. Ventre, "Wavelet-based detection of DoS attacks," in *Proceedings of IEEE Global Telecommunications Conference (GLOBECOM '06)*, pp. 1–6, San Francisco, Calif, USA, November 2006.
[20] G. Carl, R. R. Brooks, and S. Rai, "Wavelet based Denial-of-Service detection," *Computers and Security*, vol. 25, no. 8, pp. 600–615, 2006.
[21] J. Gao, G. Hu, X. Yao, and R. K. C. Chang, "Anomaly detection of network traffic based on wavelet packet," *EURASIP Journal on Advances in Signal Processing*, pp. 1–16, 2009.
[22] M. Hamdi and N. Boudriga, "Detecting Denial-of-Service attacks using the wavelet transform," *Computer Communications*, vol. 30, no. 16, pp. 3203–3213, 2007.
[23] B. Liu, Y. Li, Y. Hou, and X. Sui, "The identification and correction of outline based on wavelet transformation of traffic flow," in *Proceedings of the International Conference on Wavelet Analysis and Pattern Recognition*, pp. 2–4, Beijing, China, November 2007.
[24] M. Li, "Change trend of averaged Hurst parameter of traffic under DDOS flood attacks," *Computers and Security*, vol. 25, no. 3, pp. 213–220, 2006.
[25] C. Jin, H. Wang, and K. G. Shin, "Hop-count filtering: an effective defense against spoofed DDoS traffic," in *Proceedings of the 10th ACM Conference on Computer and Communications Security*, pp. 30–41, October 2003.
[26] W. Lee, W. Fan, M. Miller, S. J. Stolfo, and E. Zadok, "Toward cost-sensitive modeling for intrusion detection and response," *Journal of Computer Security*, vol. 10, no. 1-2, pp. 5–22, 2002.
[27] X. Lu, W. Hu, and M. Li, "Capturing packets of network traffic using WinPcap," *International Journal Electronics and Computers*, vol. 1, no. 2, pp. 169–172, 2009.
[28] K. J. Astrom, *Adaptive Control*, Prentice Hall, Upper Saddle River, NJ, USA, 2nd edition, 1994.
[29] M. Li, J. Li, and W. Zhao, "Experimental study of DDOS attacking of flood type based on NS2," *International Journal Electronics and Computers*, vol. 1, no. 2, pp. 143–152, 2009.
[30] M. Li, "Fractal time series—a tutorial review," *Mathematical Problems in Engineering*, vol. 2010, Article ID 157264, 26 pages, 2010.
[31] M. Li and W. Zhao, "Representation of a stochastic traffic bound," *IEEE Transactions on Parallel and Distributed Systems*. Preprint.
[32] M. Li and W. Zhao, "Variance bound of ACF estimation of one block of fGn with LRD," *Mathematical Problems in Engineering*, vol. 2010, Article ID 560429, 14 pages, 2010.
[33] M. Li, "Generation of teletraffic of generalized Cauchy type," *Physica Scripta*, vol. 81, no. 2, Article ID 025007, 10 pages, 2010.
[34] M. Li and J.-Y. Li, "On the predictability of long-range dependent series," *Mathematical Problems in Engineering*, vol. 2010, Article ID 397454, 9 pages, 2010.
[35] M. Li and S. C. Lim, "Modeling network traffic using generalized Cauchy process," *Physica A*, vol. 387, no. 11, pp. 2584–2594, 2008.
[36] M. Li, S. C. Lim, and W. Zhao, "Investigating multi-fractality of network traffic using local Hurst function," *Advanced Studies in Theoretical Physics*, vol. 2, no. 10, pp. 479–490, 2008.
[37] M. Li, "Modeling autocorrelation functions of long-range dependent teletraffic series based on optimal approximation in Hilbert space—a further study," *Applied Mathematical Modelling*, vol. 31, no. 3, pp. 625–631, 2007.
[38] M. Li and S. C. Lim, "A rigorous derivation of power spectrum of fractional Gaussian noise," *Fluctuation and Noise Letters*, vol. 6, no. 4, pp. C33–C36, 2006.

[39] S. C. Lim and M. Li, "A generalized Cauchy process and its application to relaxation phenomena," *Journal of Physics A*, vol. 39, no. 12, pp. 2935–2951, 2006.
[40] S. C. Lim, M. Li, and L. P. Teo, "Locally self-similar fractional oscillator processes," *Fluctuation and Noise Letters*, vol. 7, no. 2, pp. L169–L179, 2007.
[41] S. C. Lim, M. Li, and L. P. Teo, "Langevin equation with two fractional orders," *Physics Letters A*, vol. 372, no. 42, pp. 6309–6320, 2008.
[42] M. Li and P. Borgnat, "Forward for the special issue on traffic modeling, its computations and applications," *Telecommunication Systems*, vol. 43, no. 3-4, pp. 145–146, 2010.
[43] M. Li, W.-S. Chen, and L. Han, "Correlation matching method for the weak stationarity test of LRD traffic," *Telecommunication Systems*, vol. 43, no. 3-4, pp. 181–195, 2010.
[44] M. Li and S. C. Lim, "Power spectrum of generalized Cauchy process," *Telecommunication Systems*, vol. 43, no. 3-4, pp. 219–222, 2010.
[45] C. Cattani and J. Rushchitsky, *Wavelet and Wave Analysis as Applied to Materials with Micro or Nanostructure*, vol. 74 of *Series on Advances in Mathematics for Applied Sciences*, World Scientific, Hackensack, NJ, USA, 2007.
[46] C. Cattani, "Harmonic wavelet approximation of random, fractal and high frequency signals," *Telecommunication Systems*, vol. 43, no. 3-4, pp. 207–217, 2010.
[47] C. Cattani and A. Kudreyko, "On the discrete harmonic wavelet transform," *Mathematical Problems in Engineering*, vol. 2008, Article ID 687318, 7 pages, 2008.
[48] C. Cattani and A. Kudreyko, "Application of periodized harmonic wavelets towards solution of eigenvalue problems for integral equations," *Mathematical Problems in Engineering*, vol. 2010, Article ID 570136, 8 pages, 2010.
[49] C. Cattani, "Harmonic wavelet analysis of a localized fractal," *International Journal of Engineering and Interdisciplinary Mathematics*, vol. 1, no. 1, pp. 35–44, 2009.
[50] G. Toma, "Specific differential equations for generating pulse sequences," *Mathematical Problems in Engineering*, vol. 2010, Article ID 324818, 11 pages, 2010.
[51] E. G. Bakhoum and C. Toma, "Mathematical transform of traveling-wave equations and phase aspects of quantum interaction," *Mathematical Problems in Engineering*, vol. 2010, Article ID 695208, 15 pages, 2010.
[52] W. S. Chen, "Galerkin-Shannon of Debye's wavelet method for numerical solutions to the natural integral equations," *International Journal of Engineering and Interdisciplinary Mathematics*, vol. 1, no. 1, pp. 63–73, 2009.
[53] C. Cattani and A. Kudreyko, "Harmonic wavelet method towards solution of the Fredholm type integral equations of the second kind," *Applied Mathematics and Computation*, vol. 215, no. 12, pp. 4164–4171, 2010.