Countermeasures on Application Level Low-Rate Denial-of-Service Attack*

Yajuan Tang

Department of Electronic Engineering, Shantou University
yjtang@stu.edu.cn

Abstract. Low-Rate Denial-of-Service (LRDoS) attack is an emerging threat to Internet because it can evade detection and defense schemes for flooding based attacks. LRDoS attack at application level is particularly difficult to counteract as it mimics legitimate client. Although there are several approaches proposed to mitigate LRDoS attacks, they are limited to particular protocols, target systems, or attack patterns that they are not able to detect this threat at application level. In this paper, we propose a nonparametric detection algorithm and a hybrid defense system to mitigate LRDoS attacks at application level. Our extensive experiments have confirmed the effectiveness of the detection and defense system.

1 Introduction

Denial-of-Service (DoS) attack is one of the most serious security concerns in the Internet as they prevent legitimate users from using Internet services. DoS attack is also known as flooding attack due to the fact that it sends out high rate requests or packets to consume resources. However, high rate traffic is statistically abnormal to legitimate traffic that it is easy to be detected [1]. On the contrary, Low-Rate Denial-of-Service (LRDoS) attack can evade detection and defense methods designed for flooding-based attacks, therefore attracts more and more interests in literature.

LRDoS attack is typically illustrated by ON/OFF traffic pattern because it sends out intermittent pulses of malicious packets or requests to a target [2–5]. It was originally designed to attack TCP mechanism [2,3,5] and later was generalized to application level by exploiting vulnerability of feedback control based Internet services [4].

In this paper, we focus on application level LRDoS attacks because they make detection and defense more difficult than network level attacks do. The difficulty lies in two major reasons. First, application level LRDoS attacks send requests following the same characteristic of legitimate clients such that it is difficult to distinguish attack requests from legitimate requests. Second, there is a variety of LRDoS attacks that exploit application specific knowledge to launch an attack,

* This work is supported by the National Natural Science Foundation of China (60903185) and Industry-Universities-Research Institutes Collaboration Foundation of Guangdong (cgzhzd0717).

T.W. Chim and T.H. Yuen (Eds.): ICICS 2012, LNCS 7618, pp. 70–80, 2012.

which requires a general countermeasure approach without knowing particular vulnerability the attack exploits. For example, approaches [2, 5–13] have been proposed to detect and defend LRDoS attacks. However, they are limited either to particular protocols and target victim or to specific attack patterns that are not able to protect Internet services. [2, 5, 6, 12] rely on specific features of TCP; [7,10] assume the attack is periodic; [11] is designed for the specific LRDoS attack proposed in [14] that needs to estimate victim's service time.

Motivated by these aspects, we provide a detection algorithm and a defense system to mitigate LRDoS attacks at application level. Our detection algorithm has two distinct features: (1) unlike most existing detection mechanisms, it can detect both periodic and non-periodic LRDoS attacks; (2) it exploits the feature directly affected by LRDoS attacks and uses nonparametric sequential test. To do this, it adopts a nonparametric method to identify the anomalies in admission rate resulted from LRDoS attacks. Admission rate is a parameter that determines whether a request is accepted or not. It is widely used in Internet services for the provision of guaranteed QoS. More importantly, it relies on no protocols or applications which makes our algorithm general for detecting application layer LRDoS attacks. Simulation and testbed results show our detection algorithm can effectively discover LRDoS attacks in various attack scenarios. In the testbed experiments, the attack can be found just after the arrival of the first pulse.

We propose a defense algorithm to quickly restore service immediately after detecting an attack. Our defense algorithm is a combination of rate-based and queue-length based method. It proactively drops requests according to a dropping probability before requests reach the admission controller. By doing so, our defense system overcomes slow reaction time of admission controller and also helps the victim to achieve a utilization level close to the desired value. Our experiments show the defense system can quickly restore system performance as soon as an attack is detected.

The remainder of this paper is organized as follows. The next section reviews related works. The nonparametric approach is presented in Section 3, followed by an introduction for the defense system in Section 4. Section 5 details the LRDoS attack simulation and testbed results. Section 6 concludes this paper.

2 Related Works

Since an LRDoS attack has ON/OFF traffic pattern, it can evade the detection schemes targeting at flooding-based DoS attacks and therefore motivates the design of several new countermeasure approaches [2, 5–13]. However, these approaches cannot mitigate LRDoS attacks at application level because of two reasons. First, since all of these approaches aim at LRDoS attacks on TCP or particular systems (e.g., wireless network, P2P network, etc.), they rely on features specific to TCP or those systems. For example, Luo et al. proposed a detection scheme that exploits anomalies in incoming TCP data traffic and outgoing TCP ACK traffic [5]. Shevtekar et al. regarded a TCP flow as malicious if its period is equal to the fixed minimal retransmission timeout (RTO) and its burst length

is no less than other connections' RTTs [9]. Maciá-Fernández et al. [11] established the defense goal as reducing service queue positions seized by the attacker and discussed some possible defense techniques. To detect distributed LRDoS attacks, Xiang et al. [13] used generalized entropy and information distance to quantify the anomalies in packets. Their solution requires to control all routers in the network.

Second, the majority of previous work focuses on the Shrew attack [2] that has a fixed attack period equal to TCP's minimal RTO. For example, spectral-analysis approaches rely on the spectrum difference between Shrew attack flows and normal flows [7,8,10]. Sun et al. suggested using autocorrelation and dynamic time warping (DTW) to detect Shrew attacks, because its traffic bursts are the same and have fixed period [6]. However, LRDoS attacks are not necessarily periodic. Our detection method makes no assumption of periodic attacks that it is more general than these approaches.

Although some mechanisms have been proposed to detect application level DoS attacks, they could not effectively detect LRDoS attacks because their assumptions and detection features are usually not applicable to LRDoS attacks. For example, Ranjan et al. [15] assumed that inter-arrival time of requests decreases over time and requests also ask for specific resources that could overload the server. LRDoS attacks dispatch requests following the ON/OFF pattern and need not send specific requests that could lead to severer damage. Xie and Yu proposed a hidden semi-Markov model to represent normal user's browsing behaviors [16]. Although their method may notice a high request rate when an LRDoS attack sends request, it may miss the attack when the attacker dispatches nothing because it has difficulty in distinguishing high request rate caused by an LRDoS attack from a flash crowd. Moreover, such parametric method depends on the accuracy of the model. Our detection scheme employs suitable feature and nonparametric method to uncover LRDoS attacks, thus it avoids the disadvantages of existing detection mechanisms.

3 Detecting LRDoS Attacks

We propose a new detection scheme to identify LRDoS attacks, which distinguishes itself from other detection mechanisms against LRDoS attacks in two aspects. First, it aims at both periodic and non-periodic LRDoS attacks while the majority of existing detection mechanisms focuses on periodic LRDoS attacks. Second, for the sake of effectiveness and efficiency, it exploits the feature directly affected by LRDoS attacks and uses nonparametric sequential test. More precisely, our scheme employs admission rate for the detection, because an LRDoS attack intends to force the victim server to drop normal requests by throttling its admission rate [17]. Admission rate can detect both periodic and non-periodic LRDoS attacks because it does not rely on LRDoS attack's frequency character. Moreover, we adopt a nonparametric CUSUM algorithm [18] and light-weight detection algorithm to avoid unrealistic assumptions on arrival patterns of legitimate requests and to achieve online detection.

As the CUSUM method assumes that the mean value of the variable changes from negative to positive when a change occurs, we define the detection measure as

$$Z(t) = d(t) - median(d_n) - \nu \times IQR(d_n), \tag{1}$$

where $d(t) = 1 - \alpha(t)$, $\alpha(t)$ is admission rate, d_n is the training dataset obtained in the absence of LRDoS, and ν is a parameter adjusted by the user. To avoid noise in the training dataset, we adopt robust statistics [19] in the design of $Z(t)$. That is, we use median instead of mean and employ interquartile range (IQR) defined as the difference between the third and the first quartiles [19] to replace standard deviation.

Let T^{det} be the detection time for $Z(t)$ when

$$y_{z(t)} > C_{cusum} = 0, \tag{2}$$

where $y_{z(t)}$ is the CUSUM value of $Z(t)$ and C_{cusum} is the threshold of CUMSUM that is defined as the mean of sequence $d_n - median(d_n)$. The detection system reports the existence of an LRDoS attack when (2) holds.

To make it sequential, we update $y_{z(k)}$, $k \in \mathbb{Z}^+$ every time unit as

$$y_{z(k)} = \begin{cases} y_{z(k-1)} + Z(k) & k \in \mathbb{Z}^+, \\ 0 & k = 0. \end{cases} \tag{3}$$

Let T^{att} be the start time of an LRDoS attack, and the detection delay τ^{delay} is

$$\tau^{delay} = T^{att} - T^{det}. \tag{4}$$

The averaged detection delay (ADD) and false alarm rate (FAR) are denoted as [20]

$$ADD(T^{det}) = E(T^{det} - T^{att}), \tag{5}$$

$$FAR(T^{det}) = \frac{1}{E_0 T^{det}}, \tag{6}$$

where $E(\cdot)$ is the expectation function and E_0 is the expectation of $y_{z(k)}$, $k \in \mathbb{Z}^+$, before the attack. As our target of detection is a sequence of pulses, we define averaged detection pulse (ADP) instead of ADD as the performance metric. ADP is defined as number of pulses before the detection alarm is raised. For example, suppose the attack sequence starts at $t = 0$ with $\tau = 1$ second. If the algorithm reports at $t = 3.2$ second that an attack is present, $ADP = 3$ as three pulses are passed before the alarm.

4 Defending against LRDoS Attacks

In this section we present a defense scheme. As the attack traffic exhibits ON/OFF pattern, we could drop packets adaptively according to the packet arrival rate. This can be done through a virtual queue [21] whose virtual capacity is updated by the change in the arrival rate. However, our defense mechanism not only needs

to be sensitive to rate changes, but also needs to regulate the queue length to a target value such that the Internet service has a high utilization rate.

A queue-length based algorithm can achieve the goal of regulating traffic. For example, Queue Regulated Virtual Queue (QRVQ) scheme [22] updates the virtual capacity according to the deviation of the queue length from its target value. QRVQ is reported to be more robust under various traffic situations at the cost of slow adaptation to changes of arrival rate.

Recognizing the complementary strengths of rate-based and queue-based algorithms, we propose a hybrid algorithm for our defense system that maintains a *virtual target arrival rate*, denoted by λ_v, and *a virtual queue* that has a capacity of C_v requests per second. C_v and λ_v are updated according to

$$\begin{cases} \dot{C}_v(t) = K_1(\lambda_v(t) - \lambda(t)), \\ \dot{\lambda}_v(t) = K_2(q^* - q(t)), \end{cases} \tag{7}$$

where $q(t)$ is the queue length, q^* is the target queue length, K_1 and K_2 are constants, and $\lambda(t)$ is the arrival rate of the requests. Therefore, when $q(t) > q^*$, λ_v is reduced. Consequently, C_v also decreases, which results in rejecting more requests. The dropping probability is defined as

$$\pi(t) = (\lambda(t) - C_v(t))/\lambda(t). \tag{8}$$

This new algorithm has two advantages. First, it updates C_v directly based on traffic rates, which allows for a fast response to traffic changes. Second, by maintaining a target queue length, the traffic rate is more predictable than rate-based one.

Our defense scheme is summarized in Algorithm 1.

Algorithm 1. Hybrid Algorithm Framework

Input: capacity of defense system C,
arrival rate to the defense system λ,
virtual capacity C_v,
virtual queue λ_v.
Output: drop rate π

1: **for** each arrived request **do**
2: $q(t) = q(t) - C \times \Delta t + \lambda(t)\Delta t$, update queue size;
3: $\pi = (\lambda(t) - C_v)/\lambda(t)$, update drop rate;
4: update C_v and λ_v according to (7);
5: drop packet in real queue according to π;
6: **end for**

5 Evaluation

5.1 Target Victim

In this section we adopt a web server [4] as targeted victim whose model is given in Figure 1. Details of the server please refer to [4]. In the presence of

attack, the server evolves through three different stages before returning to the steady state: saturation, recovery I, and recovery II. Figure 2(a) shows how the admission rate and the system utilization behave during these three stages. We have proven that according to various attack periods, the sequence of these stages forms three general cases as shown in Figure 2(b). Details please refer to [23]. In the following experiments, we will investigate the performance of our proposed scheme in these three cases. It is worth noting that examining this web server is just an example of our methodology that can be applied to investigate the effect of detection and defense system on other Internet services.

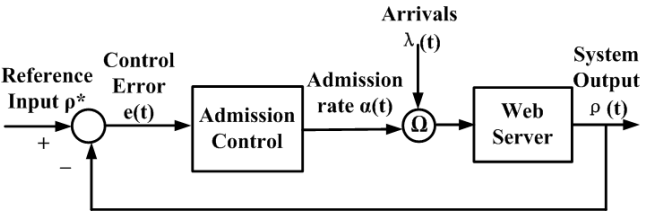
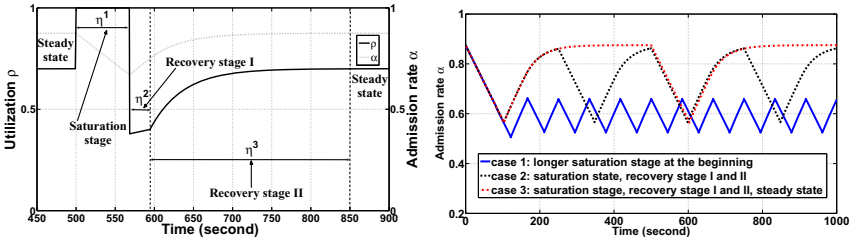


Fig. 1. Structure of the web server used in this section



(a) The effect of one attack pulse at $t = 0$ on admission rate and utilization.

(b) The effect of a sequence of attack pulses on admission rate.

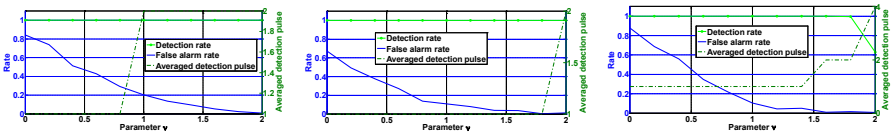
Fig. 2. The effects of the attack on the admission rate. η^1, η^2 and η^3 are time elapsed between different stages, respectively.

5.2 Simulation Results

This subsection presents MATLAB simulation results to evaluate the performance of detection algorithm and defense system. We use the parameters from [4]: $A = 0.00267$, $B = 0.2$, $C = 0.024$, $D = -1.4$, $\ell = 75$, $K = 0.01$, $\mu = 90$, and $\rho^* = 0.7$. Details of these parameters please refer to [4].

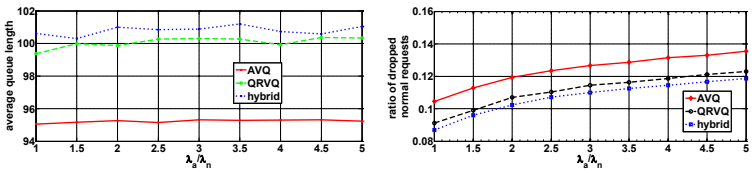
When evaluating the performance of the detection algorithm, we generated three kinds of background traffic [24–26]: log-normal, pareto and poisson distributed traffic. The parameters of the distributions were set such that for each of the distribution the mean arrival rate was 100 requests per second. We also generated periodic and random attack sequences in the simulation. Due to paper limit, we only present poisson distributed background traffic and periodic attack as illustrations, the others can be found in the supplementary [27].

Figure 3 shows the detection rate (DET), FAR and ADP for case 1, 2, and 3. The background traffic was poisson distributed; ν varied from 0 to 2. We found for all the three cases, our detection algorithm is effective as $DET \geq 0.996$, $FAR \leq 0.024$ when $\nu = 1.8$. In case 3, the detection rate decreases as ν increases. This can be explained as follows. The period of case 3 is larger than the period of case 1 and 2, as shown in [28] and Figure 2(b). Therefore the oscillation of admission rate due to attack in case 3 has a low frequency than that in case 1 and 2. This low oscillation results in a low detection rate as increased ν means detection algorithm is tolerant to deviations. Also due to the fact that increased ν means normal oscillations of admission rate are likely to be classified as normal, FAR is a decreasing function of ν. In addition, we found there is a tradeoff between FAR and ADP for all the three cases. This is because increased ν reduces detection measure $Z(t)$, which in turn makes it longer to accumulate the deviation of attacked admission rate to normal value before the deviation exceeds the threshold. Among three cases, case 3 has the highest ADP in general. This results from the the fact that it has the largest attack periods that yields an oscillation with low frequency.



(a) The detection performance of case 1. (b) The detection performance of case 2. (c) The detection performance of case 3.

Fig. 3. The detection rate, false alarm rate, and averaged detection pulse of CUSUM detection algorithm. The background normal traffic is poisson distributed.



(a) Averaged queue length. (b) Ratio of dropped normal requests.

Fig. 4. Defense scheme comparison for periodic attacks

To evaluate the performance of our defense system, we compared it with AVQ [21] and QRVQ [22] by considering the average queue length and ratio of dropped normal requests to total normal requests. In our simulation, $q^* = 100$ requests, $K_1 = 0.8$, $K_2 = 0.5$, and the attack pulses are periodic. We also conducted random attacks, whose results can be found in the supplementary [27]. Figure 4 presents the results, which shows our hybrid approach achieves a good performance. Figure 4(a) illustrates the averaged queue length versus λ_a/λ_n,

where λ_a and λ_n are arrival rate of attack and normal requests, respectively. It can be seen that the queue length regulation achieved by AVQ is low because it is a pure rate based scheme. On the other hand, our hybrid scheme and QRVQ both regulate the queue length very closely to the predefined target value. The averaged queue length of our hybrid scheme is similar to that of QRVQ, however, they differ in the ratio of dropped normal requests to total normal requests, as shown in Figure 4(b). Figure 4(b) demonstrates that the hybrid scheme drops less normal requests than QRVQ. This is because the hybrid approach responds quickly to arrival rate changes, thus can accept more requests.

5.3 Testbed Results

A testbed was set up to emulate the attack scenarios and to evaluate the detection and defense algorithms. Figure 5 shows the diagram of the testbed. The target was a web server running `Minihttpd` 1.9 [29], which was equipped with a proportional-integral (PI) controller to perform admission control. A legitimate client generated HTTP requests continuously using `Httperf` [30] with an arrival rate of 100 requests per second. There were also seven attack zombies, each of which generated attack traffic at a rate of 100 requests per second. These zombies were synchronized by the network time protocol.

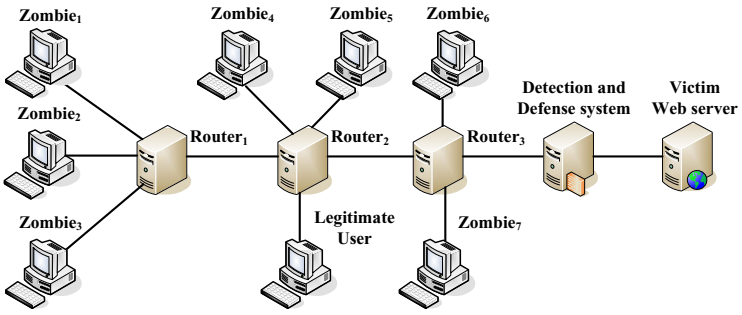


Fig. 5. The testbed topology

To evaluate the effectiveness of our detection method, we used it for the traces collected from the testbed. We wanted to know whether the algorithm we evaluated in simulation can still work in real situations. More importantly, whether the impact of ν is the same for DET, FAR and ADP in actual environment. Thus we let the detection parameters be the same as the ones used for simulation. We conducted 16 experiments, each one contained an attack period varied from 29.3 seconds to 249 seconds, covered all the three cases. We repeated each experiment 4 times and obtained the mean of each performance metric. The results are shown in Table 1. We found the impact of ν is the same as the simulation

Table 1. Detection performance on testbed

	case 1			case 2			case 3		
	DET	FAR	ADP	DET	FAR	ADP	DET	FAR	ADP
$\nu = 0$	1	0.14	1	1	0.33	1	1	0.25	1
$\nu = 1$	1	0.14	1	1	0.12	1	1	0	1
$\nu = 2$	1	0.09	1	1	0	1	1	0	1

results. That is, detection rate is 1 for all ν; increased ν causes small FAR. ADP equals 1 for all the three cases. These results show that the proposed CUSUM detection scheme can effectively discover LRDoS attacks.

The performance of the defense system was also assessed. Figures 6 illustrates how the defense system can mitigate the effects of an attack. It shows the arrival rates at the defense system and the resulting arrival rates at the server. The peaks in the request arrival rate caused by LRDoS attacks are clearly smoothed out by the defense system.

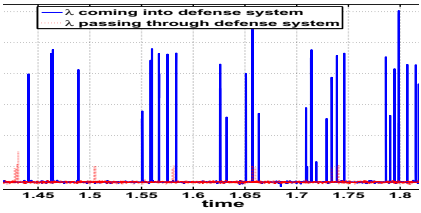


Fig. 6. Testbed results for the performance of the defense system

6 Conclusions

LRDoS attacks at application level exploit vulnerability of Internet services and are hard to counteract because of their low averaged rate and sending legitimate requests. Despite the fact that some approaches have been presented to mitigate LRDoS attacks, they are limited to certain protocols, system or attack patterns. Therefore, it is important to address the security issues and threats to these Internet services.

In this paper we restrict our attention to the countermeasures on LRDoS attacks at the application level. We have designed a nonparametric sequential test and an adaptive queue management algorithm to quickly restore system performance. Extensive simulation and testbed experiments have been carried out to validate the results.

References

1. Kuzmanovic, A., Knightly, E.: Low-rate TCP-targeted Denial-of-Service attacks and counter strategies. IEEE/ACM TON 14(4), 683–696 (2006)

2. Kuzmanovic, A., Knightly, E.: Low-rate TCP-targeted Denial-of-Service attacks: The shrew vs. the mice and elephants. In: ACM SIGCOMM (2003)
3. Guirguis, M., Bestavros, A., Matta, I., Zhang, Y.: Exploiting the transients of adaptation for RoQ attacks on Internet resources. In: IEEE ICNP (2004)
4. Guirguis, M., Bestavros, A., Matta, I., Zhang, Y.: Reduction of quality RoQ attacks on Internet end-systems. In: IEEE INFOCOM (2005)
5. Luo, X., Chang, R.: On a new class of Pulsing Denial-of-Service attacks and the defense. In: ISOC NDSS (2005)
6. Sun, H., Lui, J., Yau, D.: Defending against low-rate TCP attacks: dynamic detection and protection. In: IEEE ICNP (2004)
7. Chen, Y., Kwok, Y., Hwang, K.: Filtering Shrew DDoS attacks using a new frequency-domain approach. In: IEEE WoNS (2005)
8. Chen, Y., Hwang, K.: Collaborative detection and filtering of Shrew DDoS attacks using spectral analysis. JPDC 66(9), 1137–1151 (2006)
9. Shevtekar, A., Anantharam, K., Ansari, N.: Low rate TCP Denial-of-Service attack detection at edge routers. IEEE Communication Letters 9, 363–365 (2005)
10. Thatte, G., Mitra, U., Heidemann, J.: Detection of low-rate attacks in computer networks. In: IEEE Global Internet Symposium (2008)
11. Maciá-Fernández, G., Rodriguez-Góomez, R., Diaz-Verdejo, J.: Defense techniques for low-rate DoS attacks against application servers. Computer Networks 54(15), 2711–2727 (2010)
12. Chang, C., Lee, S., Lin, B., Wang, J.: The taming of the shrew: mitigating low-rate TCP-targeted attack. IEEE TNSM 7(1), 1–13 (2010)
13. Xiang, Y., Li, K., Zhou, W.: Low-rate DDoS attacks detection and traceback by using new information metrics. IEEE TIFS 6(2), 426–437 (2011)
14. Maciá-Fernández, G., Díaz-Verdejo, J., Garcia-Teodoro, P., Toro-Negro, F.: LoR-DAS: A Low-Rate DoS Attack against Application Servers. In: Lopez, J., Hämmerli, B.M. (eds.) CRITIS 2007. LNCS, vol. 5141, pp. 197–209. Springer, Heidelberg (2008)
15. Ranjan, S., Swaminathan, R., Uysal, M., Knightly, E.: DDoS-resilient scheduling to counter application layer attacks under imperfect detection. In: IEEE INFOCOM (2006)
16. Xie, Y., Yu, S.: A large-scale hidden semi-Markov model for anomaly detection on user browsing behaviors. IEEE/ACM TON 17(1), 54–65 (2009)
17. Guirguis, M., Bestavros, A., Matta, I., Zhang, Y.: Reduction of quality (RoQ) attacks on dynamic load balancers: Vulnerability assessment and design tradeoffs. In: IEEE INFOCOM (2007)
18. Brodsky, B., Darkhovsky, B.: Non-Parametric Statistical Diagnosis Problems and Methods. Kluwer Academic Publishers (2000)
19. Rousseeuw, P., Hubert, M.: Robust statistics for outlier detection. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 1(1), 73–79 (2011)
20. Tartakovsky, A., Rozovskii, B., Blazek, R., Kim, H.: A novel approach to detection of intrusions in computer networks via adaptive sequential and batch-sequential change-point detection methods. IEEE TOSP 54(9), 3372–3382 (2006)
21. Kunniyur, S., Srikant, R.: Analysis and design of an adaptive virtual queue (AVQ) algorithm for active queue management. In: ACM SIGCOMM (2001)
22. Deng, X., Yi, S., Kesidis, G., Das, C.: Stabilized virtual buffer (SVB) - an active queue management scheme for internet Quality-of-Service. In: IEEE Globecom (2002)

23. Tang, Y., Luo, X., Hui, Q., Chang, R.K.: Understanding the vulnerability of feedback-control based internet services to low-rate DoS attacks (manuscript for publication)
24. Karagiannis, T., Molle, M., Faloutsos, M., Broido, A.: A nonstationary Poisson view of internet traffic. In: IEEE INFOCOM (2004)
25. Park, K., Kim, G., Crovella, M.: On the effect of traffic self-similarity on network performance. In: SPIE PCNS (1997)
26. Downey, A.: Evidence for long-tailed distributions in the internet. In: ACM IMW (2001)
27. Tang, Y.: Supplementary to "countermeasures on application level low-rate Denial-of-Service attack"
28. Tang, Y., Luo, X., Chang, R.K.C.: Protecting internet services from low-rate DoS attacks. In: CIP (2007)
29. mini_httpd, `http://www.acme.com/software/mini_httpd/`
30. httperf, `http://www.hpl.hp.com/research/linux/httperf/`