Title	Filtering of shrew DDoS attacks in frequency domain
Author(s)	Chen, Y; Hwang, K; Kwok, YK
Citation	Proceedings - Conference On Local Computer Networks, Lcn, 2005, v. 2005, p. 786-793
Issued Date	2005
URL	http://hdl.handle.net/10722/45910
Rights	Creative Commons: Attribution 3.0 Hong Kong License

Filtering of Shrew DDoS Attacks in Frequency Domain*

Yu Chen, Kai Hwang, and Yu-Kwong Kwok

***Abstract*—The shrew *Distributed Denial of Service* (DDoS) attacks are periodic, bursty, and stealthy in nature. They are also known as *Reduction of Quality* (RoQ) attacks. Such attacks could be even more detrimental than the widely known flooding DDoS attacks because they damage the victim servers for a long time without being noticed, thereby denying new visitors to the victim servers, which are mostly e-commerce sites. Thus, in order to minimize the huge monetary losses, there is a pressing need to effectively detect such attacks in real-time.**

Unfortunately, effective detection of shrew attacks remains an open problem. In this paper, we meet this challenge by proposing a new signal processing approach to identifying and detecting the attacks by examining the frequency-domain characteristics of incoming traffic flows to a server. A major strength of our proposed technique is that its detection time is less than a few seconds. Furthermore, the technique entails simple software or hardware implementations, making it easily deployable in a real-life network environment.

***Index Terms*— Network security, distributed denial of service (DDoS), reduction of quality (RoQ), digital signal processing (DSP), Internet traffic analysis**

I. Introduction

*D*ISTRIBUTED *Denial of Service* (DDoS) attacks have become one of the major threats to Internet services and electronic transactions [5], [22], [26]. A typical DDoS attack prevents legitimate users from accessing the victim for certain services. The network resources could be denied by overwhelming the target with a huge amount of traffic flows launched through multiple *Zombies*. Essentially, such kind of attacks is targeting at undermining the availability of certain systems or services. DDoS attacks degrade the performance of the networks even though the links are not saturated [19].

As of now, there is no "silver bullet" against DDoS attacks although a plethora of research efforts has been injected into this area. A traditional DDoS attack can be characterized as brute-force, sustained high-rate, or specifically designed to take advantages of the protocol limitations or the software vulnerabilities.

Recently, a variant category of DDoS attack has been identified. This novel type of attack, with a *low average rate*, exploits the transient phases of a system's dynamic behavior. Such low-rate attacks introduce significant inefficiencies that tremendously reduce system capacity or service quality, yet exhibiting a *stealthy* behavior. In the literature, this kind of attacks is referred to as *shrew* attacks [17] or *Reduction of Quality* (RoQ) attacks [12], [13].

Comparing to traditional DDoS attacks, which are flooding in nature, shrew attacks are much more difficult to be detected. Therefore, they can damage the victim for a long time without being noticed [13]. Such a prolonged period of damage, if occurred on an e-commerce Web site (e.g., Amazon.com), can transparently repel new commercial transactions or frustrate existing customers. Significant monetary losses would then result from these attacks.

Unfortunately, it has been proven theoretically and experimentally that countermeasures developed for traditional DDoS attacks are ineffective in fighting against shrew attacks [13], [17], [21]. Furthermore, being "masked" by the background traffic, shrew attacks are very difficult to be identified in the time domain, which is the usual avenue of defense in combating network attacks.

Several security researchers have explored the usage of *digital signal processing* (DSP) and other signal analysis techniques for traffic analysis in network security control [1], [2], [4], [14], [15], [16], [24]. Luo and Chang [20] studied the characteristics of shrew attack with a wavelet approach. Sun, et al. [27] suggested detecting shrew attacks via a *dynamic time wrapping* (DTW) technique. Unfortunately, none of these defense schemes could identify and filter out the attack streams effectively and accurately.

Previously, we proposed an algorithm named HAWK [18] (*Halting Anomaly with Weighted choKing*) that works by judiciously identifying malicious shrew packet flows using a small flow table and dropping such packets decisively to halt the attack such that well-behaved TCP sessions can re-gain their bandwidth shares. One drawback of HAWK is its insensitivity to distributed shrew attacks.

* Presented at the *First IEEE LCN Workshop on Network Security* (WoNS) held in conjunction with the *30th Annual IEEE Conference on Local Computer Networks (LCN 2005),* Nov.15-17, 2005, Sydney, Australia. This work was supported by US National Science Foundation under the ITR Grant 0325409 at the University of Southern California.

Yu Chen and Kai Hwang are with the Internet and Grid Computing Laboratory, Viterbi School of Engineering, University of Southern California, Los Angeles, CA 90089, USA. They can be reached by e-mail: cheny@usc.edu and kaihwang@usc.edu, respectively.

Yu-Kwong Kwok is with the Department of Electrical and Electronic Engineering, University of Hong Kong (HKU), China. This work was done while he visited the University of Southern California during his sabbatical leave from the HKU (e-mail: ykwok@hku.hk).

In this paper, we propose a novel approach to filtering out shrew attack flows by analyzing the *amplitude spectrum distribution* in the frequency domain. Taking samples of packet arriving rate as the time-domain signal, followed by transforming it into frequency domain by DFT (*Discrete Fourier Transform*), we construct a filter by using the *hypothesis-test theory*. Based on analysis of more than 10,000 simulation test points, our detection achieved a confidence interval of 99.9% (with error level ±3.29σ).

Specifically, we make the following contributions:

1. Using a hypothesis test theory and Gaussian distribution model, we show that our shrew-filtering algorithm achieves pretty higher accuracy. Thus, our scheme blocks malicious shrew flows with high confidence level (> 99.9%), while exhibiting low probability (< 0.1%) in losing legitimate TCP flows.
2. One of the distinct advantages of our approach is that DFT and frequency-domain analysis are standard DSP methods that could be implemented efficiently in hardware, thanks to the modern VLSI technology. Thus, our shrew-filtering algorithm would not incur much overhead in routers since the whole processing could be carried out in fast hardware, while the routers perform their normal routing operations.
3. Another advantage of shrew-filtering algorithm is to cut off the malicious shrew streams totally, which is similar to our MAFIC algorithm [6] that block flooding DDoS flows. In this manner, we minimize the damages of shrew streams on legitimate flows.

The rest of this paper is organized as follows. In Section 2, we present the rationale of this work. With introduction of shrew attack and a discussion of frequency domain properties of the shrew streams and TCP flows, we set up our hypothesis test framework and determine the optimal detection threshold. Section 3 introduces our simulation setup and performance matrices. Simulation results and performance analysis are given in Section 4. Finally we conclude in Section 5.

II. The Proposed Shrew Filtering Algorithm

We first introduce the fundamentals of shrew attack. Then, we compare its frequency domain properties with legitimate TCP flows. Based on their differences, a hypothesis test framework is set up and the optimal detection threshold will be chosen. In the last subsection, we present in detail our novel shrew-filtering algorithm for cutting off shrew attack flows.

A. Overview of Shrew Attacks

The earliest case of low-rate TCP-targeted DDoS attack was reported in 2001. But it had not been studied thoroughly until Kuzmanovic and Knight [17] pioneered the work in identifying and characterizing such type of attacks. They studied the rationale of the shrew attack and analyzed the critical parameters that affect the efficiency on TCP flows. They also indicated the limitation of currently available DDoS defense mechanism. However, they have not proposed any efficient countermeasures against the low-rate attacks.

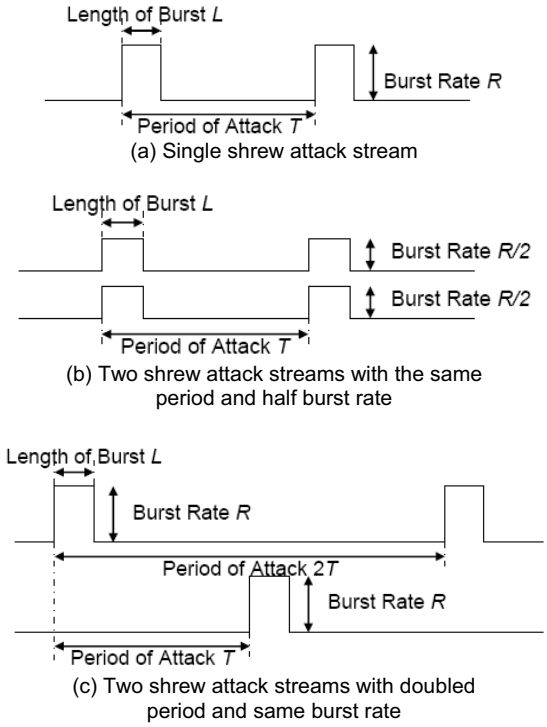


Fig. 1. An illustration of various types of shrew attack streams.

As shown in Fig. 1, a single source shrew attack is modeled as a square waveform packet stream with an attack period T, length of the burst L, and the burst rate R. The period T is calculated by the estimated TCP RTO timer implementations at legitimate sources. During the burst with a peak rate R, the shrew pulses create a bursty but severe congestion on the links to the victim. The legitimate TCP flows will decrease their sending rate as defined by the rate-limiting mechanism that cuts the window size and adapts to the network capacity.

For higher throughput, the TCP protocol uses a predefined value of RTO with a fixed RTO incrementing pattern [25]. The shrew attacks take advantage of this RTO recovery feature by adjusting the attack period to match with the RTO period. The feature causes the shrew attack streams to occupy the link bandwidth periodically by sending pulses (Fig. 1). This makes the legitimate TCP flows always "see" heavily burdened links. Such legitimate TCP flows may undergo a congestion control and reduce their rates significantly.

A successful shrew attack may occupy bandwidth lower than 10% of the legitimate TCP flows [17]. Such kind of periodic pulses is very difficult to detect by traffic management algorithms and by methods based on existing traffic volume analysis at the time domain. This is because the average share of bandwidth consumption is not very high.

In distributed scenarios, attacks launched by multiple zombies could lower their individual traffic rates even further, thereby making detection much harder. As shown in Figs. 1(b) and 1(c), the distributed attack sources could decrease its

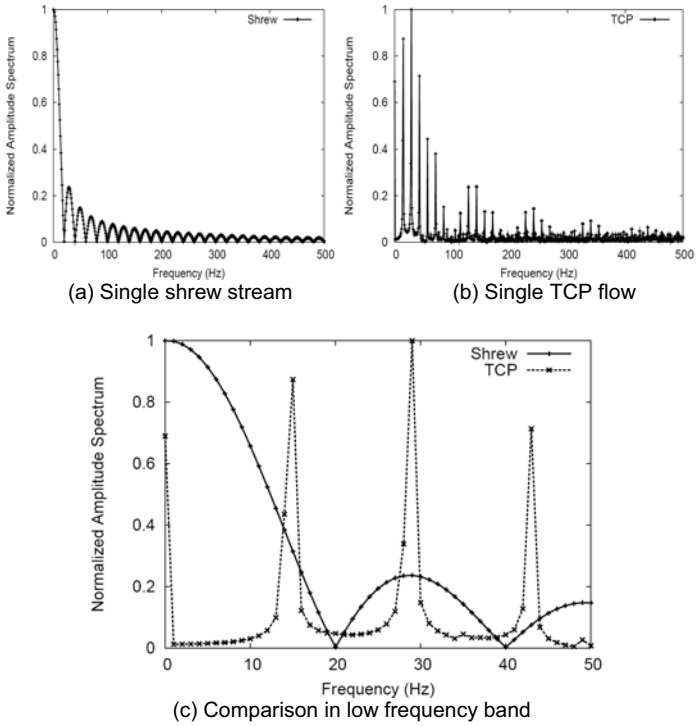


Fig. 2. Normalized amplitude spectrum of the shrew pulse stream and of the TCP flow.

average traffic rate either by lowering the peak rate or using longer attack periods. Detecting the signs of such attacks using traffic time series in time domain is therefore ineffective.

B. Analysis of Amplitude Spectrum Distribution

Although it is very challenging to detect and respond to the low-rate attacks using defense measures developed against DDoS attacks, the periodicity itself provides a clue for developing new defense mechanism [8]. Periodic signals and non-periodic signals present different properties in frequency domain. These variants could be detected conveniently using signal processing techniques.

We take the number of arrived packets as the signal and sample it every 1 *ms*. At each step, we sample the number of arrived packets *x(n)*. Then we convert the time-domain series into its frequency domain representation using DFT (*Discrete Fourier Transform*) [3]:

$$DFT(x(n),K)=\frac{1}{N}\sum_{n=0}^{N-1}x(n)\times e^{-j2\pi kn/N} \quad \text{k=0,1,2,…,N-1} \quad (1)$$

Figure 2(a) shows the normalized amplitude spectrum of a shrew attack and Fig. 2(b) is that of a legitimate TCP flow. Nyquist sampling theorem [3] indicates that the highest frequency of our analysis is 500 Hz.

Comparing to the single TCP flow, more energy of shrew pulse stream appears in lower frequency bands. This property is more profound in Fig. 2(c) that zooms into the low frequency band of [0 Hz, 50 Hz].

Based on observing the normalized amplitude spectrum, we find that it is feasible to design a detection algorithm by comparing their energy density in the low frequency band from 0 Hz to 50 Hz. The difference between the summations of amplitude in this range could be large enough to segregate shrew pulse streams from the legitimate TCP flows.

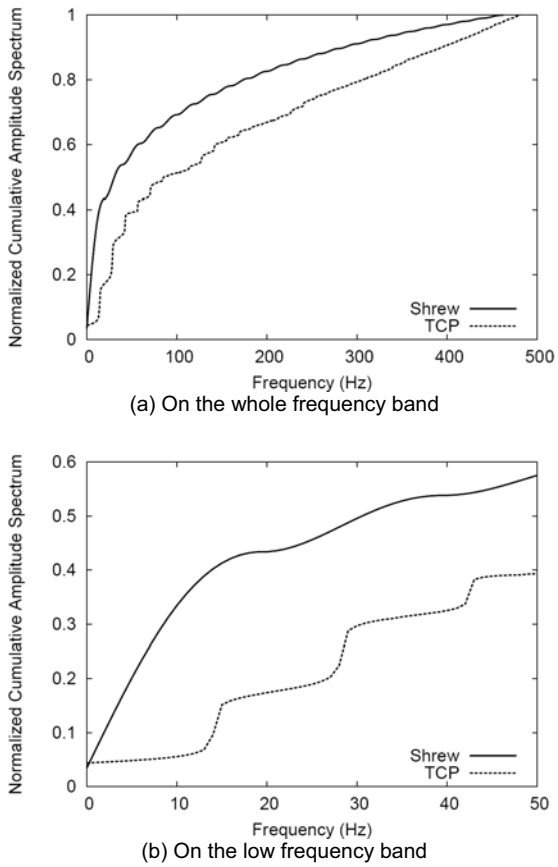


Fig. 3. Normalized cumulative amplitude spectrum of the shrew stream and of the TCP flow.

Fig. 3(a) compares the *normalized cumulative amplitude spectrums* (NCAS) of TCP and shrew flows, and Fig. 3(b) zooms into the low frequency band of [0 Hz, 50 Hz]. It is around the frequency point of 20 Hz that the distance of the two curves is the maximum. As such, we call this point as the *K-point*. It is also the ending point of the first peak of amplitude spectrum curve of shrew pulse in Fig. 2(c).

Actually such a lower frequency band biased energy distribution could be used as the signature of low-rate shrew attacks. Since the shrew attack streams are aiming at the dynamic deficiency in the RTO mechanism of TCP protocol while trying to minimize the average bandwidth utilization, they have to construct congestions periodically at the moments when victims are recovering from RTO.

This implies that if an attacker would like to blur the signature, he has to input more packets into the network at other time points. This will increase the bandwidth occupation and thus destroy the stealthy nature of low-rate shrew attacks. We need a rule to identify the signature and make the decision on when a cumulative amplitude spectrum value at the K-point has

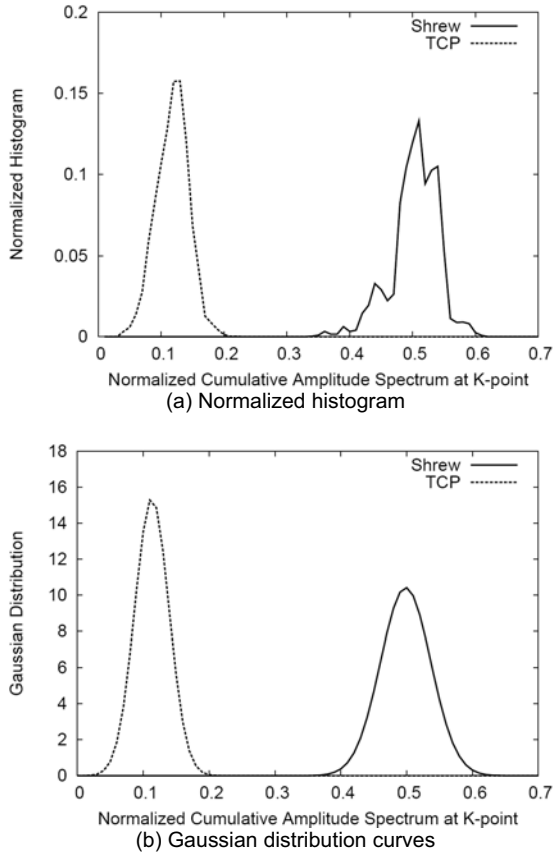


Fig. 4. Normalized NCAS distribution of the shrew stream and of the TCP flow at the K-point.

been calculated. Since there are two choices, the binary hypothesis test [11] appeals to this application.

C. Hypothesis Test Analysis

Since noise signals existing in communication channels and introduced in the sampling process are random, we need to confirm statistically that the variation of NCAS at the K-point is limited in such a range that allows us to distinguish shrew pulse streams from TCP flows with high confidence.

Fig. 4(a) presents the normalized histogram of NCAS' distribution at the K-point. Both TCP and shrew streams' data are calculated in a sample space of more than 8,000 data points. The statistics of TCP and shrew streams are given below:

$$TCP: \begin{cases} Average(\mu) = 0.1131 \\ Standard\_Deviation(\sigma) = 0.026 \end{cases}$$

$$Shrew: \begin{cases} Average(\mu) = 0.4985 \\ Standard\_Deviation(\sigma) = 0.038 \end{cases}$$

According to *Central Limit Theorem* that given a distribution with a mean μ and variance σ^2, the sampling distribution approaches a *Gaussian (Normal)* distribution [11]. Thus, we can describe the distribution of NCAS at K-point using Gaussian distribution model:

TABLE I
GAUSSIAN DISTRIBUTIONS' CONFIDENCE LEVELS

Error Level	Prob. That Error Is Smaller	Prob. That Error Is Larger	TCP Threshold	Shrew Threshold
±σ	68%	~1:3	0.1311±0.026	0.4985±0.038
±1.65σ	90%	1:10	0.1311±0.043	0.4985±0.046
±1.96σ	95%	1:20	0.1311±0.051	0.4985±0.074
±3σ	99.7%	1:370	0.1311±0.078	0.4985±0.114
±3.29σ	99.9%	1:1000	0.1311±0.086	0.4985±0.125

$$G(x;\mu,\sigma) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\} \quad (2)$$

Fig. 4(b) is the Normal Distribution curves of TCP flow and shrew pulse stream. In detection theory, 3σ Error Level could give us a confidence interval of 99.7% [11], meaning that error level of ±3σ is good enough even in high precision detection scenarios. Table I lists the confidence levels of TCP and shrew streams and their corresponding threshold settings.

Fig. 4(b) presents that the distance between distribution curves of TCP and shrew flows is larger than ±3.29σ. As indicated in Table I, the detection threshold at K-point could be safely selected to be 0.3 and this choice ensures us with confidence interval larger than 99.9%.

In other words, the probability of cutting off a TCP flow as shrew stream or vice versa is lower than 0.1%. This shows that our hypothesis detection approach achieves pretty high accuracy and precision. The algorithm of our detection process is specified below in pseudo code:

Hypothesis Detection Algorithm:

```
01: While shrew filtering algorithm is on
02:    While sampling is not done
03:       Continue sampling packets number per 1ms
04:    Convert the time-domain series into frequency domain
05:    Calculate the NCAS value at K-point
06:    If NCAS ≤ Threshold Then
07:       Mark the flows as legitimate
08:    Else
09:       Mark the flows as shrew flow
```

D. Shrew-Filtering Algorithm

Based on the hypothesis test, we proposed an algorithm to cut off flows with NCAS value at the K-point higher than the detection threshold. Although the source IP addresses are generally spoofed in attack packets, it is safe to use the 4-tuple *{Source IP, Source Port, Destination IP, Destination Port}* as the traffic flow labels.

To minimize the storage overhead incurred by the extra data structures needed, we store only the output of a hash function with the label as the input instead of the label itself. Our shrew-filtering algorithm drops malicious packets using the data structures: *Permanent Drop Table* (PDT), *Suspicious Flow Table* (SFT) and *Nicely-Behaved Flow Table* (NFT) as shown in Fig. 5.

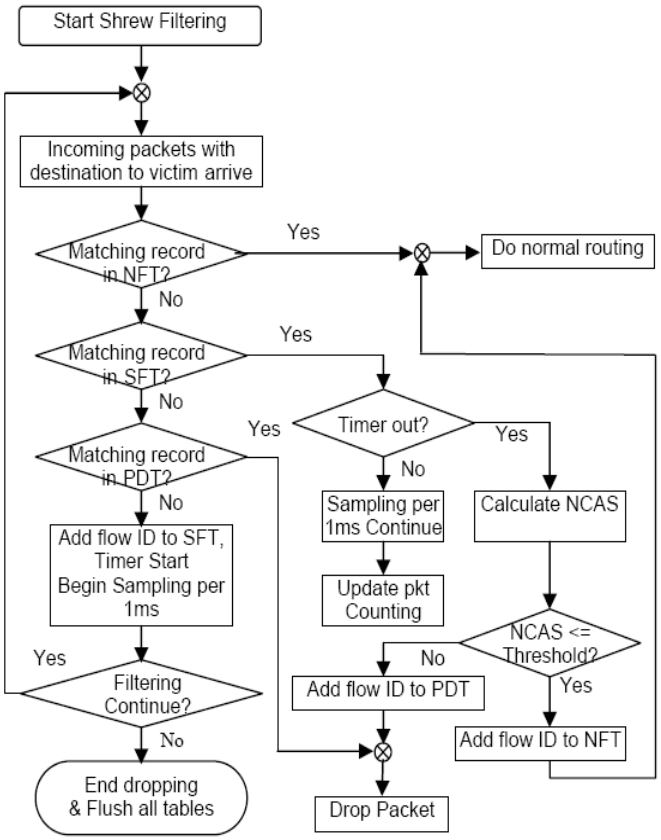


Fig. 5. The shrew-filtering algorithm for dropping malicious packets. (NFT: Nicely-Behaved Flow Table, SFT: Suspicious Flow Table, PDT: Permanent Drop Table, NCAS: Normalized Cumulative Amplitude Spectrum)

If an incoming packet label is in NFT, this packet is routed normally. If it is in PDT, this packet is dropped. If in SFT, we continue sampling until timer out. If there is no matching in any table, this packet belongs to a new flow and it would be added into SFT, then sampling begins and timer starts.

Once timer is expired for certain flow, we convert the time-domain series into its frequency domain representation using DFT, and compare its NCAS at K-point with detection threshold. If its NCAS value is lower than the threshold, we move its record into NFT. All further incoming packets in this flow will be routed normally. If the NCAS value is higher than the threshold, we cut off these flows into PDT.

III. NS-2 Simulation Setup

We have implemented the shrew-filtering algorithm in the NS-2 simulator, which is a widely recognized packet level discrete event simulator [23]. A subclass of connector named *ShrewFilter* is added to the head of each *SimplexLink.* A *TrafficMonitor* is coded into the simulator to compute the traffic matrices. The *ShrewFilter* is used to process the sample array and to calculate the NCAS of flows leading to the victim. Then, the PDT or NFT entries are set accordingly. The system configuration of the simulation scenario is shown in Fig. 6.

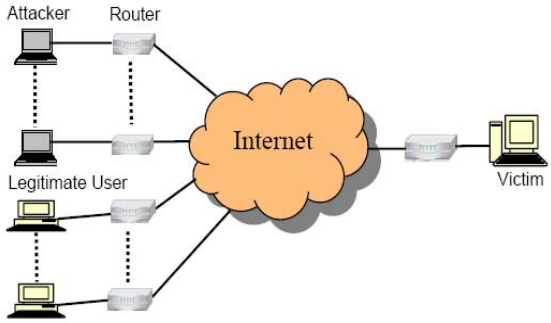


Fig. 6. The simulation scenario and experimental setting.

Our simulation consists of a variety of Internet traffic patterns. Multiple scenarios are studied including single TCP flow vs. single shrew flow, single TCP flow vs. distributed shrew flows, multiple TCP flows vs. single shrew flow, and multiple TCP flows vs. distributed shrew flows. The distributed attack patterns include the cases shown in Figs. 1(b) and 1(c). Our notation used in the simulation is listed in Table II.

TABLE II
Definition of Notation

Symbol	Definition
T	Attack Period (sec)
R	Attack Pulse Peak Rate
L	Attack Pulse Burst Length (sec)
N_S	Number of Shrew Flows
N_T	Number of TCP Flows
ρ	Normalized TCP Throughput
τ	Response Time

IV. Simulation Results and Analysis

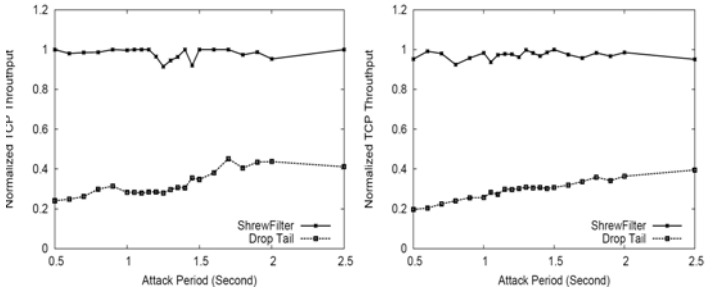
We compared the results of our shrew-filtering algorithm with the well-known *active queue management* (AQM) algorithm *Drop Tail*. We also examined the response time performance of our algorithm since it determines the duration of damage to a victim site.

A. Normalized Throughput

Our NS-2 simulations are carried out with the topology shown in Fig. 6 for different combinations of legitimate TCP flows and shrew attack streams. We compared the TCP throughputs achieved by the shrew-filtering algorithm and the Drop Tail algorithm using the comparison metric *normalized throughput* (ρ), which is defined as the ratio of average throughput achieved by the TCP flow(s) with DDoS stream to the throughput achieved without DDoS streams.

The normalized throughput indicates the severity of the damage that the shrew streams have done to the performance of legitimate TCP flows. The lower the normalized throughput is, the greater the damage. In our simulations, we consider the link capacity of the last hop to the victim as 2 Mbps.

Since all TCP variants are equally vulnerable to shrew DoS stream of 50 ms or higher [17], we use TCP-Reno for the purpose of experiment. The sources of the shrew attack streams are illustrated at the top left of Fig. 6. Their delay is a random



(a) Single TCP flow (b) Five TCP flows

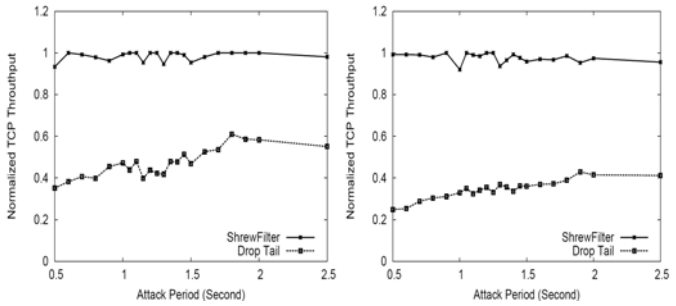
Fig. 7. Scenarios of TCP flows under single shrew attack.

variable uniformly distributed within (60 ms, 120 ms).

We start with single shrew-stream scenarios. Fig. 7 compares the throughputs of TCP flows using the Drop Tail scheme and our shrew-filtering algorithm. The x-axis is the attack period and the y-axis is the normalized throughput TCP flows achieved. Fig. 7(a) shows the scenario of single TCP flow under attack of single shrew stream modeled in Fig. 1(a). Fig. 7(b) corresponds to the scenario of five TCP flows under attack from a single shrew stream.

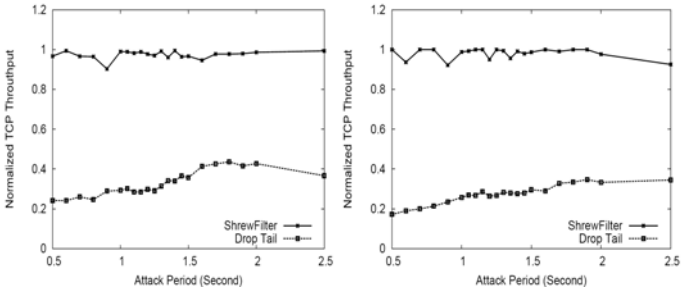
It is clear that under the Drop Tail algorithm, the throughput of legitimate TCP flows is far below the actual attainable throughput and the link utilization is very inefficient. With our shrew-filtering algorithm, the gain in TCP throughput is significant. It reaches what legitimate flows can reach when there is no shrew stream. Our hypothesis test model can identify shrew streams with a high confidence level. We filter out shrew streams before they hurt the legitimate flows.

Distributed shrew streams are hard to be detected because of their much lower average traffic rates. Simulations are carried out using four shrew streams that are distributed in either space domain (Fig. 1(b)) or time domain (Fig. 1(c)), respectively. Again, we studied their effects on single and five legitimate TCP flows. Fig. 8 presents the case where shrew streams are distributed in space but synchronized as in Fig. 1(b). Four shrew streams are from four difference sources with the same attack periods and the same burst lengths. However, their peak rate is only R/4. This means that their average traffic rate is only 1/4 of that of the single source attack.



(a) Single TCP flow (b) Five TCP flows

Fig. 8. Normalized throughput of TCP flows vs. 4 spatially distributed shrew attack flows.



(a) Single TCP flow (b) Five TCP flows

Fig. 9. Normalized throughput of TCP flows vs. 4 timely distributed shrew attack flows.

Fig. 9 compares the throughputs of TCP flows under the Drop Tail algorithm and our shrew-filtering algorithm in the case that shrew streams are distributed in time fashion but synchronized as in Fig. 1(c). Four shrew streams are from four difference sources with the same peak rates and the same burst lengths. However, their attack periods are 4T. This distribution makes the interval between pulses four times longer to bring down the average traffic rate to 1/4 of that of the single source attack pulse stream.

These results show that our shrew-filtering algorithm is indeed capable of recognizing distributed shrew streams with lower average traffic rate. This is one major advantage of frequency spectrum technique over bandwidth utilization analysis. Even if the shrew streams were launched from more zombies to further lower their average bandwidth utilization, their frequency spectrum would possess the same properties.

In other words, the shrew-filtering mechanism is effective even if the attack is launched through larger number of streams with lower burst peak rate. In fact, if zombies use longer individual attack periods, higher percentages of its energy will be located in the low frequency band we are monitoring.

B. Response Time

The response time is a critical parameter to evaluate the performance of our shrew-filtering algorithm. In general, the time a DDoS defense algorithm takes to detect whether malicious flows exist or not is used a measure to monitor the traffic conditions. The time is varied according to the traffic load on the link.

However, the load on the link does not affect the response time of our shrew-filtering algorithm. Results in Section 4.1 show that the performance of the shrew-filtering algorithm is coherent under different traffic conditions, where we used the same 5-second sampling time.

The effects of variant sampling length are determined by the signal's periodicity. If the sampled sequence presents similar frequency characteristics of original signal, then the variance of sampling time will not impact on our detection precision.

Fig. 10(a) presents the distributions of NCAS at the K-point of TCP flows and shrew streams. They are sampled from 1 second to 5 seconds. As the sampling time decreases, the NCAS at the K-point of TCP flows scatters wider. Therefore,

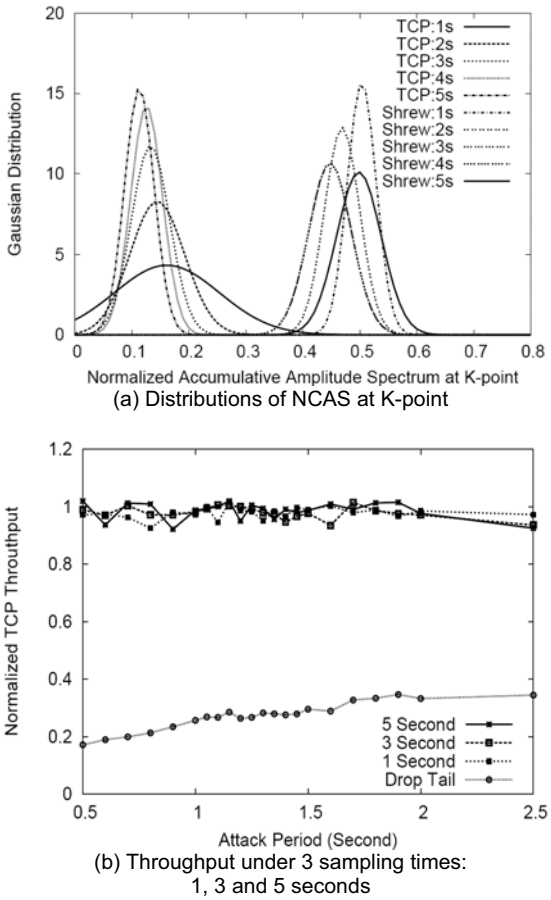


Fig. 10. Effects of sampling lengths on the TCP and shrew throughput.

the probability of treating a legitimate TCP flow as shrew stream increases. However, the distributions of NCAS at the K-point of shrew streams are pretty stable. If we stick on the threshold of 0.3, the high detection confidence level is maintained even the sampling time decreases to 3 seconds.

Table III shows the confidence levels of different sampling times. We observe ±1.96σ (95%), ±3σ (99.7%) and ±3.29σ (99.9%) error levels of TCP and shrew streams. When sampling time (the response time τ) is longer than 2 seconds, there is no overlap between the ±3.29σ error level ranges of TCP flow and shrew stream. Therefore, the confidence level of detecting and filtering shrew streams is very high (99.9%) while $\tau \geq 2$ seconds.

With $\tau = 1$ second, we observed an overlap in both ±3σ and ±3.29σ error ranges, but no overlap for ±1.96σ error level. This implies that information carried by sampled signal series cannot separate TCP flows from shrew streams with such a high confidence level (99.7%). However, the shrew-filtering algorithm still could respond to the shrew attacks in 1 second. We cut off it with little sacrifice in confidence level (95%). Figure 10(b) shows the throughput of five TCP flows under the attack of four distributed shrew streams. Clearly, all sampling series achieved much higher throughput than the Drop Tail algorithm.

TABLE III
CONFIDENCE LEVELS OF DIFFERENT SAMPLING TIMES

	Sampling Time	1 Second	2 Second	3 Second
±1.96σ/ 95%	TCP Flow	0.1614±0.176	0.1445±0.094	0.1327±0.067
	Shrew Stream	0.5036±0.050	0.4690±0.061	0.4508±0.067
±3σ/ 99.7%	TCP Flow	**0.1614±0.270**	0.1445±0.144	0.1327±0.102
	Shrew Stream	**0.5036±0.076**	0.4690±0.093	0.4508±0.103
±3.29σ/ 99.9%	TCP Flow	**0.1614±0.296**	0.1445±0.158	0.1327±0.112
	Shrew Stream	**0.5036±0.083**	0.4690±0.102	0.4508±0.113
	Sampling Time	4 Second	5 Second	
±1.96σ/ 95%	TCP Flow	0.1258±0.078	0.1131±0.051	
	Shrew Stream	0.4479±0.074	0.4985±0.074	
±3σ /99.7%	TCP Flow	0.1258±0.120	0.1131±0.078	
	Shrew Stream	0.4479±0.112	0.4985±0.114	
±3.29σ/ 99.9%	TCP Flow	0.1258±0.132	0.1131±0.086	
	Shrew Stream	0.4479±0.123	0.4985±0.125	

V. CONCLUSIONS

In this paper, we have proposed to cut off low-rate TCP-targeted DDoS attack flows using the periodicity properties of different flows in the frequency domain. Our analysis and simulations show that more energy of low-rate shrew attacks is located in the lower frequency band, comparing with the legitimate TCP flows.

There is one limitation in our shrew-filtering scheme. It is still difficult to identify malicious flows that exhibit "transient" behaviors such as "mice" flows. To deal with such scenarios, we believe that we can use our approach to detect the attacks at packet level instead of flow level. Indeed, our extended results [7] indicate that high detection accuracy was achieved using a collaborative distributed detection mechanism.

In our on-going efforts, we are implementing the shrew-filtering algorithm on the DETER test-bed to evaluate this work in an environment closer to the reality [9], [10]. With this practical study as the background, we can then extend the shrew-filtering algorithm and hypothesis test framework based detection methodology to address other types of DDoS attacks that present variant patterns in frequency domain.

Essentially, all Internet traffic flows could be abstracted and processed as continuous periodic signals in time domain. If a frequency "spectrum" of Internet traffic flow mix is available, the frequency domain processing technology could facilitate the traffic analysis process efficiently without incurring much extra burden to the routers.

REFERENCES

[1] P. Abry and D. Veitch, "Wavelet Analysis of Long-Range-Dependent Traffic," *IEEE Trans. Information Theory*, vol. 44, no. 1, 1998, pp. 2–15.

[2] P. Abry, R. Baraniuk, P. Flandrin, R. Riedi, and D. Veitch, "Multiscale Nature of Network Traffic," *IEEE Signal Processing Magazine*, vol. 19, no. 3, 2002, pp. 28–46.

[3] R. Allen and D. Mills, *Signal Analysis: Time, Frequency, Scale, and Structure,* John Wiley & Sons, 2004.

[4] P. Barford, J. Kline, D. Plonka, and A. Ron, "A Signal Analysis of Network Traffic Anomalies," *Proc. Internet Measurement Workshop*, 2002.

[5] R. K. Chang, "Defending Against Flooding-Based Distributed Denial-of-Service Attacks: A Tutorial," *IEEE Communications Magazine*, Oct. 2002.

[6] Y. Chen, Y.-K. Kwok, and K. Hwang, "MAFIC: Adaptive Packet Dropping for Cutting Malicious Flows to Pushback DDoS Attacks," *IEEE International Workshop on Security in Distributed Computing Systems* (SDCS-2005), 2005.

[7] Y. Chen, Y.-K. Kwok, and K. Hwang, "Collaborative Defense Against Periodic Shrew DDoS Attacks in Frequency Domain," *ACM Trans. on Information and System Security (TISSEC)*, submitted May 2005.

[8] C.-M. Cheng, H. Kung, and K.-S. Tan, "Use of Spectral Analysis in Defense against DoS Attacks," *Proc. IEEE GLOBECOM*, Taipei, China, 2002.

[9] DETER and EMIST Team Members, "Cyber Defence Technology Networking and Evaluation," *Comm. ACM*, vol. 47, no. 3, Mar. 2004, pp. 58–61.

[10] "The DETER Testbed: Overview," http://www.isi.edu/deter/docs/testbed.overview.pdf.

[11] J. Devore and N. Farnum, "Applied Statistics for Engineers and Scientists," Duxbury Press, 1999.

[12] M. Guirguis, A. Bestavros, and I. Matta, "Bandwidth Stealing via Link Targeted RoQ Attacks," *Proc. 2nd IASTED Int'l Conf. on Communication and Computer Networks*, Nov. 2004.

[13] M. Guirguis, A. Bestavros, I. Matta, and Y. Zhang, "Reduction of Quality (RoQ) Attacks on Internet End Systems," *Proc. INFOCOM* 2005.

[14] X. He, C. Papadopoulos, J. Heidemann, and A. Hussain, "Spectral Characteristics of Saturated Links," under submission, http://www.isi.edu/~johnh/PAPERS/He04a.html.

[15] P. Huang, A. Feldmann, and W. Willinger, "A Non-Intrusive, Wavelet-Based Approach to Detecting Network Performance Problems," *Proc. ACM SIGCOMM Internet Measurement Workshop*, 2001.

[16] A. Hussain, J. Heidemann, and C. Papadopoulos, "A Framework for Classifying Denial of Service Attacks," *Proc. ACM SIGCOMM* 2003.

[17] A. Kuzmanovic and E. W. Knightly, "Low-Rate TCP-Targeted Denial of Service Attacks—The Shrew vs. the Mice and Elephants," *Proc. ACM SIGCOMM 2003*, Aug. 2003.

[18] Y.-K. Kwok, R. Tripathi, Y. Chen, and K. Hwang, "HAWK: Halting Anomaly with Weighted ChoKing to Rescue Well-Behaved TCP Sessions from Shrew DoS Attacks," *Proc. Int'l Conf. on Computer Networks and Mobile Computing (ICCNMC 2005)*, Zhangjiajie, China, Aug. 2-4, 2005.

[19] K. C. Lan, A. Hussain, and D. Dutta, "The Effect of Malicious Traffic on the Network," *Proc. PAM,* La Jolla, Apr. 6-8, 2003.

[20] X. Luo and R. K. Chang, "On a New Class of Pulsing Denial-of-Service Attacks and the Defense," *Network and Distributed System Security Symposium* (NDSS'05), San Diego, CA, Feb. 2-5, 2005.

[21] R. Mahajan, S. Floyd, and D. Wetherall, "Controlling High-Bandwidth Flows at the Congested Router," *Proc. ACM 9th Int'l Conf. on Network Protocols (ICNP),* Nov. 2001.

[22] D. Moore, G. M. Voelker, and S. Savage, "Inferring Internet Denial-of-Service Activity," *Proc. USENIX Security,* 2001.

[23] NS-2, http://www.isi.edu/nsnam/ns/, 2004.

[24] C. Partridge, D. Cousins, A. Jackson, R. Krishnan, T. Saxena, and W. T. Strayer, "Using Signal Processing to Analyze Wireless Data Traffic," *Proc. ACM Workshop on Wireless Security*, Atlanta, GA, Sept. 2002.

[25] V. Paxson and M. Allman, "Computing TCP's Retransmission Timer," *Internet RFC 2988,* Nov. 2000.

[26] S. M. Specht and R. B. Lee, "Distributed Denial of Service: Taxonomies of Attacks, Tools and Countermeasures," *Proc. PDCS,* Sept. 18, 2004.

[27] H. Sun, J. Lui, and D. Yau, "Defending Against Low-rate TCP Attacks: Dynamic Detection and Protection," *Proc. IEEE International Conference on Network Protocols* (ICNP), Berlin, Germany, Oct. 2004.

BIOGRAPHICAL SKETCHES

Yu Chen received his B.S. and M.S. from Chongqing University, China in 1994 and 1997 respectively, and currently he is a Ph.D. candidate in Electrical Engineering at University of Southern California (USC). His research interest includes Internet security, Internet traffic analysis, DDoS attack detection & defense, distributed security infrastructure. He can be reached at **cheny@usc.edu.**

Kai Hwang received his Ph.D. from UC Berkeley in 1972. He is a Professor and Director of Internet and Grid Computing Laboratory at USC. An IEEE Fellow, he specializes in computer architecture, parallel processing, Internet security, and distributed systems. Presently, he leads the GridSec project supported by NSF/ITR program. Dr. Hwang can be reached at **kaihwang@usc.edu.** The GridSec web site is **http://gridsec.usc.edu/**

Yu-Kwong Kwok received the Ph.D. from Hong Kong University of Science and Technology in 1997. He is an Associate Professor of Electrical and Electronic Engineering at the University of Hong Kong (HKU). He worked on this project during his academic visit at USC in 2004-05, while taking a sabbatical leave from HKU. His research interests include Grid and mobile computing, wireless communications and network protocols. He can be reached at **ykwok@hku.hk**.