Filtering Shrew DDoS Attacks Using A New Frequency-Domain Approach

Yu Chen, Yu-Kwong Kwok, and Kai Hwang[1]
University of Southern California, Los Angeles, CA 90089, USA

Abstract—The stealthy shrew *Distributed Denial of Services* (DDoS) attacks, also known as *Reduction of Quality* (RoQ) attacks, could be even more detrimental than the more widely known flooding DDoS assaults. The reason is that such shrew attacks 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 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. Our proposed technique is effective in that its detection time is less than a few seconds. Furthermore, the technique entails simple implementation, making it deployable in real-life network environments.

Keywords: *Distributed Denial of Service* (DDoS) attacks, *Reduction of Quality* (RoQ) attacks, stealthy attacks, shrew attacks, periodic TCP attacks, spectral analysis, frequency domain, signal processing.

1. Introduction

Distributed Denial of Service (DDoS) attacks have become one of the major threats to Internet services and commercial transactions [5], [20], [24]. 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 huge amount of traffics launched through many *Zombies*. Essentially, such kind of attack is targeting at undermining the availability of certain systems or services. DDoS attacks impact the performance of the networks even the links are not saturated [18]. As of now, there is no "silver bullet" against DDoS attacks although a plethora of research efforts has been done 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 transients of a system's dynamic behavior. The 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 network assault is called *shrew* attack [16] or *Reduction of Quality* (RoQ) attack [11], [12].

[1] The research work reported here was supported by US National Science Foundation ITR Grant 0325409. Manuscript submitted to *The First IEEE LCN Workshop on Network Security (WoNS 2005)*, Sydney, Australia, on June 20, 2005.

Corresponding author: Kai Hwang, USC Internet and Grid Computing Lab, EEB 212, Los Angeles, CA 90089. E-mail: kaihwang@usc.edu, Tel.: (213) 740-4470.

Comparing to traditional DDoS siblings, which are flooding in nature, shrew attacks are much more difficult to detect and therefore can damage the victim for a long time without being noticed [12]. 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. Taking advantages of vulnerabilities of system's dynamic behavior, shrew attacks could achieve similar effects of DDoS while escaping from detection by occupying unsuspicious fraction of resources. Instead of constantly injecting traffic flows with huge rates into the network, shrew attackers send burst pulses periodically. Such low-rate attacks have high peak rate while maintaining a low average rate to exhibit "stealthy" behavior.

Specifically, a shrew attack exploits the deficiencies in the RTO (*Retransmission Time-Out*) mechanism of TCP. It throttles legitimate TCP flows by periodically sending burst pulses with high data rate. As such, TCP flows always "see" congestion on the attacked link every time it recovers from RTO time out. This type of attack may cut the throughput of TCP applications down to almost zero [16]. This is a significant problem. Indeed, given that more than 80% of traffics on Internet today are using TCP protocol [18], a majority of existing applications and commercial services are at stake. Unfortunately, it has been proven theoretically and experimentally that countermeasures developed for traditional DDoS attacks are ineffective in fighting against shrew attacks [12], [16], [19]. Furthermore, being "masked" by the background traffic, shrew attacks are very difficult to identify in the time domain, which is the usual avenue of defense in combating network attacks.

In recent years, researchers have explored the usage of signal analysis technologies in traffic analysis for network policies enforcement [1], [2], [4], [14], [22]. Due to behaviors defined by protocols or applications, such as windowing mechanism and other periodic protocol operations, the periodicity of traffic could be used as a signature for traffic monitoring or attack detection. The dominant link frequency is independent of the number of flows. Instead, it depends on the link bandwidth and packet size distribution [13]. TCP traffics exhibit a periodicity on its PSD (*Power Spectrum Density*) when the packet arrival rate is analyzed in frequency domain. Thus, the lack of periodicity could indicate that DoS attacks are going on [7]. In addition, the PSD of multi-sourced DDoS attacks are distributed in lower frequency band comparing to single-sourced DoS attacks [15]. However, none of these research efforts is capable of distinguishing malicious attack flows from legitimate ones.

Previously, we proposed an algorithm named HAWK (*Halting Anomaly with Weighted choKing*) which 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 [17]. One drawback of HAWK is that it is insensitive to distributed shrews that occupy small bandwidth shares. Sun, Liu, and Yau [25] suggested to detect shrew attacks by matching pattern with obtained attack signature via *dynamic time wrapping* (DTW) technique, then use the *deficit round robin* (DRR) algorithm to provide bandwidth allocation and protection between flows. Although this technique can tell whether shrew streams exist among legitimate flows, unfortunately it cannot distinguish malicious flows precisely. And legitimate flows are still suffered and their throughputs are lower than what it could achieve when shrew attack is not launched.

Recently, a wavelet approach has been reported to study the characteristics of low-rate TCP-targeted DoS attack [26]. The authors observed anomalies in fluctuation of incoming traffic rate and declining of outgoing TCP ACKs incurred by pulse streams. Based on these properties, they proposed a two-stage algorithm to detect the existence of pulse streams. They have verified that their approach could detect pulse streams effectively through NS-2 simulation and experiments on a NIST Net testbed. Unfortunately, with a detecting method developed, they did

not provide an efficient response scheme that could identify and filter attacking streams accurately. Furthermore, since the wavelet detection outcomes are largely dependent on the choice of detection parameters, it is unclear how to find optimal parameters that are sensitive enough to detect low-rate distributed attacks while maintaining a acceptable false positive alarm rate.

In this paper, we propose a novel approach to filter shrew attack traffics by analyzing the *amplitude spectrum distribution* in the frequency domain. Taking samples of packet arrival 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 $\pm 3.29\sigma$).

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 traffics and TCP flows, we setup our hypothesis test frame and determine the optimal detection threshold. Section 3 introduces our simulation setup and performance matrices. The simulation results and performance analysis are included in Section 4. Finally we conclude our paper in Section 5.

2. The Proposed ShrewFilter 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 frame is set up and the optimal detection threshold will be chosen. At the last subsection, we present in detail our novel shrew-filtering algorithm for cutting off shrew attack flows.

2.1 Overview of Shrew Attacks

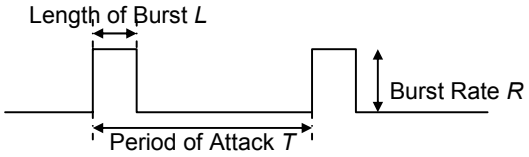
Kuzmanovic and Knight [16] pioneered the work in identifying and characterizing the TCP targeted shrew attack. They studied the rationale of the shrew attack and analyzed the critical parameters that affect the efficiency on TCP flows. They also indicated the limitation of currently available DDoS defense mechanism. However, they have not proposed any efficient countermeasures against the low-rate attacks.

As shown in Figure 1, a single source shrew attack is modeled as a square waveform packet stream with an attack period of 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. Then the legitimate TCP flows will decrease their sending rate as defined by its congestion control rate-limiting mechanism that cuts sending window size and adapts to the network capacity.

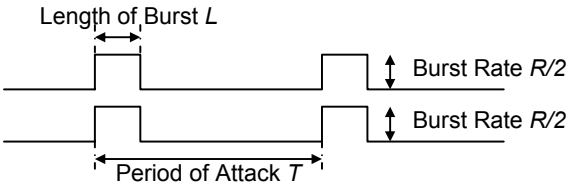
For higher throughput, the TCP protocol uses a predefined value of RTO with a fixed RTO incrementing pattern [23]. The shrew attack takes advantage of this RTO recovery mechanism by adjusting its attack period. The mechanism occupies the link bandwidth periodically by sending pulses as shown in Fig. 1. That makes the legitimate TCP always “see” a heavy burdened link every time when they try to send packets. Such legitimate TCP flows will then undergo congestion control and reduce their rates significantly. A successful shrew attack may make the throughput of legitimate TCP traffics lower than 10% of the normal level [16].

Such periodic pulses are 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 remarkably high. In distributed scenarios, attacks

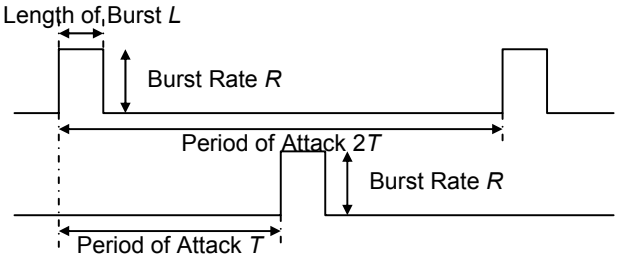
launched by multiple zombies could lower their individual traffic rates further, thereby making detection much harder. As shown in Fig. 1(b) and 1(c), the distributed attack sources could decrease its average traffics either by lowering the peak rate or using longer attack periods. Detecting the signs of such attacks using traffic time series in time domain is therefore ineffective.



(a) Single shrew attack stream



(b) Two shrew attack streams with same period and half rate



(c) Two shrew attack streams with doubled period and same rate

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

2.2 Analysis of Amplitude Spectrum Distribution

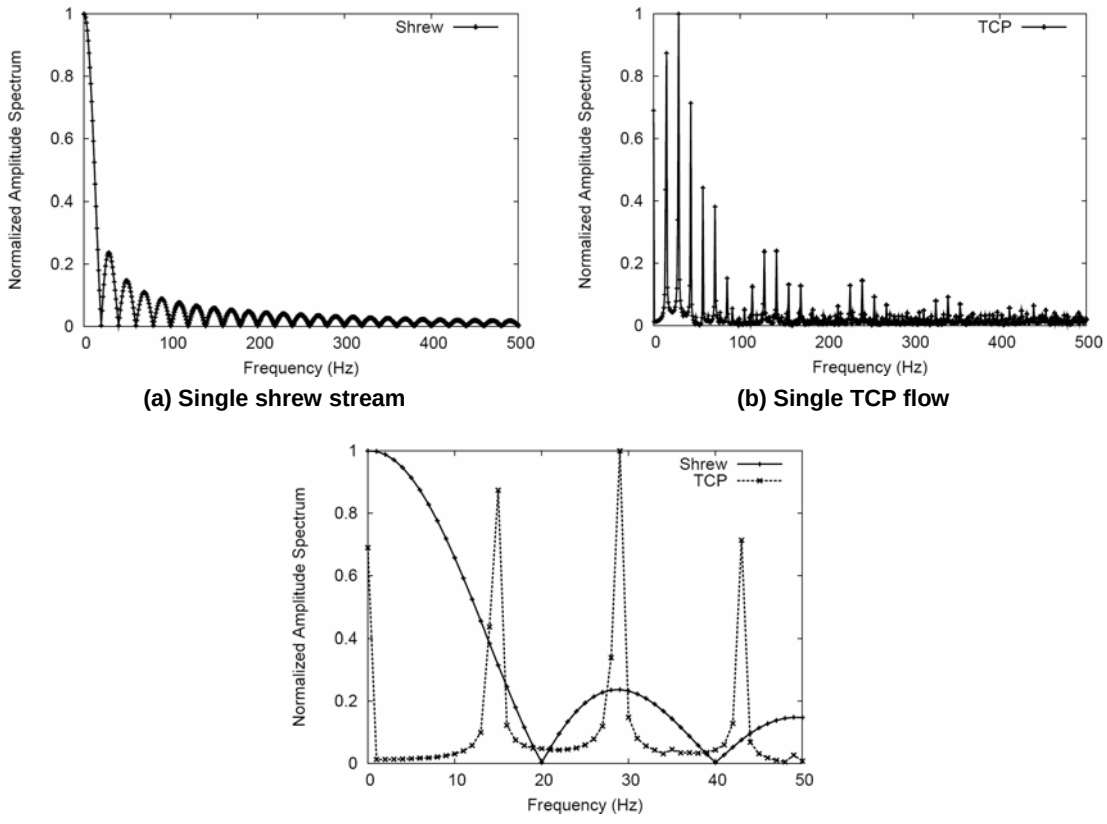
Although it is very challenging to detect and response to the low-rate attacks using defense measures developed against DDoS attacks, the periodicity itself provides a clue for developing new defense mechanism. Periodic signal and non-periodic signals present different properties in frequency domain.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

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

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

Figure 2(a) shows the normalized amplitude spectrum of shrew attack and Figure 2(b) is the normalized amplitude spectrum of legitimate one TCP flow. Nyquist sampling theorem [3] indicates that the highest frequency of our analysis is 500 Hz. Comparing to TCP flow, more energy of shrew pulse stream appears in lower frequency bands. This property is more profound in Figure 2(c) that zooms into the low frequency band of [0Hz, 50Hz].

Based on observing the normalized amplitude spectrum, we know that it is feasible to design a detection algorithm by comparing their energy density in the low frequency band of 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



(a) Single shrew stream **(b) Single TCP flow**

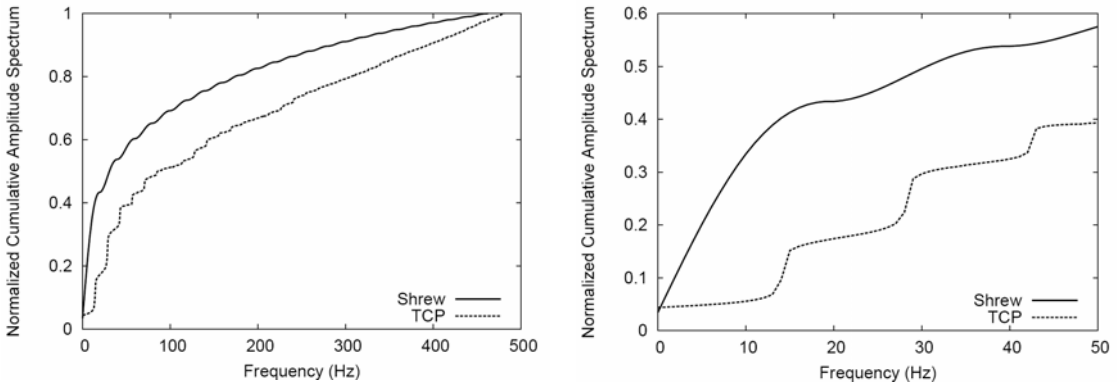
(**c) Comparison in low frequency band**

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

Figure 3(a) compares the ***normalized cumulative amplitude spectrums*** (NCAS) of TCP and shrew flows, and Fig. 3(b) zooms in low frequency band of 0 Hz to 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 such low-rate shrew attacks. Since the shrew attacking streams are aiming at the dynamic deficiency in the RTO mechanism of TCP protocol while trying to minimize the average bandwidth utility, they have to construct congestions periodically at the moments when victims are recovering from RTO. This implies that if the attacker would like to blur the signature, he/she has to input more packets into the network at other time points. However, this increases bandwidth occupation and lost the major advantage of low-rate shrew attacks.

Now we need a rule to identify the signature and make decision when a cumulative amplitude spectrum value at the K-point has been calculated. Since there are two choices, the binary hypothesis test [10] appears to be suitable in this application.

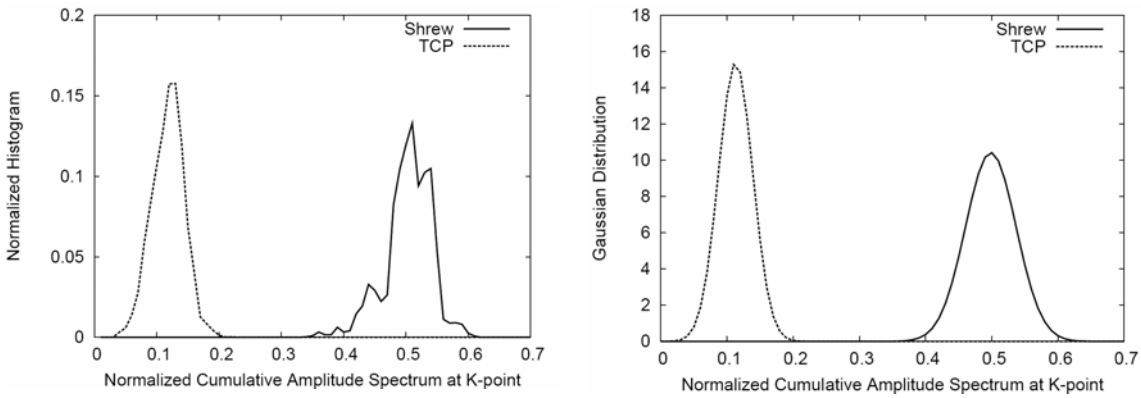


(a) On the whole frequency band **(b) On the low frequency band**

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

2.3 Hypothesis Test Analysis

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



(a) Normalized histogram **(b) Gaussian distribution curves**

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

Figure 4(a) present 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 statistical results of TCP and shrew stream are given below:

$$TCP: \begin{cases} Average(\mu) = 0.1131, \\ S\tan dard\_Deviation(\sigma) = 0.026 \end{cases} \qquad Shrew: \begin{cases} Average(\mu) = 0.4985 \\ S\tan dard\_Deviation(\sigma) = 0.038 \end{cases}$$

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 [10]. Thus, we can describe distribution of NCAS at K-point using Gaussian distribution model:

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

Figure 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% [10] that error level of ±3σ is good enough even in high precision detection scenarios. Table 1 below lists the confidence levels of TCP and shrew streams and their corresponding threshold settings.

Table 1. Gaussian Distributions' Confidence Levels

Error Level Name	Error Level	Prob. That Error is Smaller	Prob. That Error is larger	TCP threshold	Shrew threshold
One Sigma	±σ	68%	~1:3	0.1311±0.026	0.4985±0.038
90% Error	±1.65σ	90%	1:10	0.1311±0.043	0.4985±0.046
"Two" Sigma	±1.96σ	95%	1:20	0.1311±0.051	0.4985±0.074
Three Sigma	±3σ	99.7%	1:370	0.1311±0.078	0.4985±0.114
Maximum Error	±3.29σ	99.9%	1:1000	0.1311±0.086	0.4985±0.125

Figure 4(b) presents that distance between distribution curves of TCP and shrew traffics is larger than ±3.29σ. As indicated in Table 1, 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 another 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 pseudo-code of our detection process is specified below:

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
```

2.4 Shrew-Filtering

Based on the above hypothesis test framework, 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 tuple *{Source IP, Source Port, Destination IP, Destination Port}* as flow labels. To minimize the storage overhead incurred by the extra lists needed to implement shrew-filtering algorithm, we store only the output of a hash function with the label as the input instead of the label itself.

Our shrew-filtering algorithm tries to handle incoming packets according to records in *Permanent Drop Table* (PDT), *Suspicious Flow Table* (SFT) and *Nicely-behaved Flow Table* (NFT). As shown in Fig. 5, if an incoming packet label is in NFT, this packet is routed normally. If it is in the 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 the SFT, 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 move it into the PDT and this flow will be cutoff.

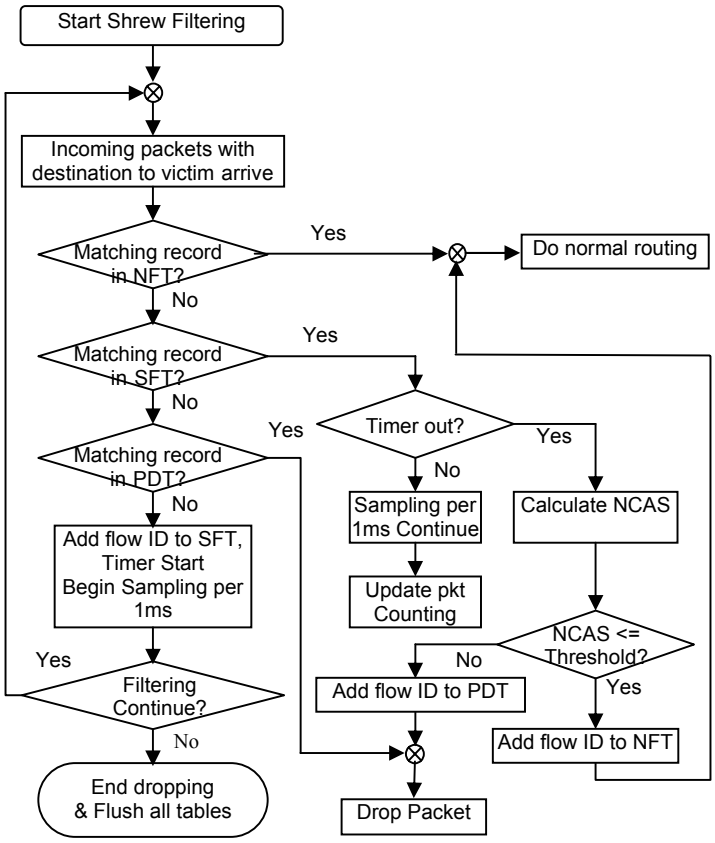


Figure 5. The Shrew-Filtering algorithm for dropping malicious packets.
(NFT: *Nicely-behaved Flow Table*, SFT: *Suspicious Flow Table*, PDT: *Permanent Drop Table,* NCAS: *Normalized Cumulative Amplitude Spectrum*)

3. NS-2 Simulation Setup

We have implemented the shrew-filtering algorithm in the NS-2 simulator, which is a widely recognized a packet level discrete event simulator [21]. A subclass of Connector named as *ShrewFilter* is added to the head of each *SimplexLink.* A *TrafficMonitor* is coded into the simulator to compute the traffic matrices. The *ShrewFilter* class is used to process the sampled array and to calculate the NCAS of flows that leading to the victim. Then, the PDT or NFT entries would be set accordingly. The system configuration of the simulation scenario is shown in Fig. 6.

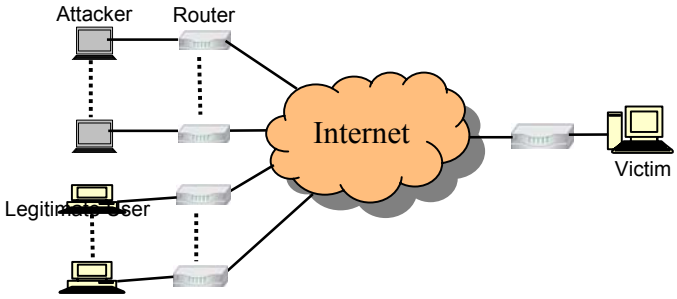


Figure 6. The simulation scenarios and experimental setting.

Our simulation consists of a variety of traffic patterns in the Internet environment. Multiple scenarios are studied including single TCP vs. single shrew, single TCP vs. distributed shrews, multiple TCP vs. single shrew, and multiple TCP vs. distributed shrew. The distribution attack patterns include cases shown in Figs. 1(b) and 1(c).

4. Simulations 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 examine the response time performance of our algorithm since it determines the duration of damage to a victim site. Our notation used throughout the simulation is listed in Table 2.

Table 2. Definition of Notation

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

4.1 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 a *normalized throughput* (ρ) as the comparison metric defined by:

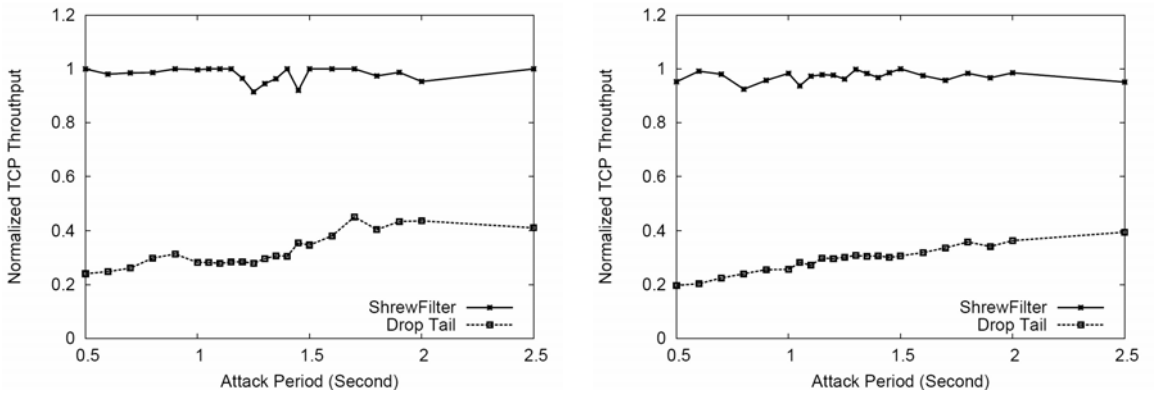
$$\rho = \frac{\text{Average throughput achieved by the TCP flow (or aggregate) with DDoS stream}}{\text{Throughput achieved without DDoS stream}} \quad \textbf{(3)}$$

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 simulation, 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 [16], we use TCP-Reno for the purpose of experiment. And the shrew stream sources are

located as the topology shown in Fig.6. Their delay to the Internet is a random variable uniformly distributed from 60 ms to 120 ms.

We start with single shrew-stream scenarios. Figure 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 flow achieved. Figure 7(a) is the scenario of single TCP flow under attack of single shrew stream modeled in Fig. 1(a). Figure 7(b) is scenario of five TCP flows under attack from a single shrew stream.

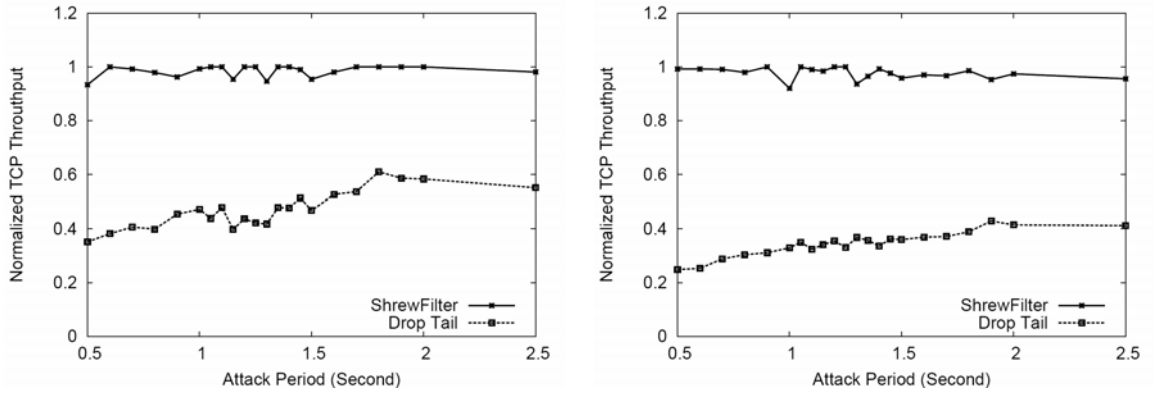
It is clear that under the Drop Tail algorithm, the throughput of legitimate TCP flow is far below the actual attainable throughput and the link utilization is very inefficient. With our shrew-filtering algorithm, the gain in TCP throughput is significant. It reaches what legitimate flows can reach when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



(a) Single TCP vs. Single Shrew attacks **(b) Five TCP vs. Singled Shrew attacks**

Figure 7. Scenarios of TCP Flows under single shrew attacks.

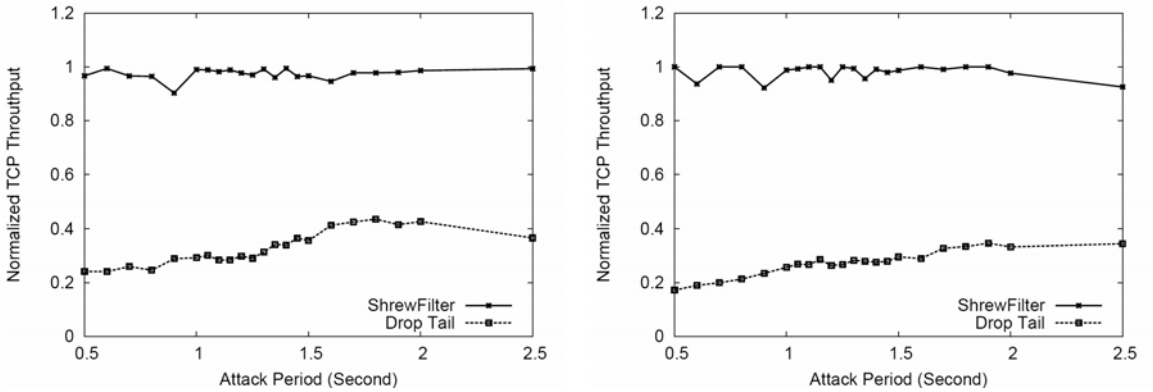
Distributed shrew streams are hard to be detected because of their much lower average traffic rates. Simulations are carried out using four shrew streams that distributed in either space domain (as in Fig. 1(b)) or time domain (as in Fig. 1(c)), respectively. Again, we studied their effects on single and five legitimate TCP flows. Figure 8 presents the case where shrew streams are distributed in space but synchronized in Fig. 1(b).



(a) Single TCP vs. 4 spatial distributed shrews **(b) Five TCP vs. 4 spatial distributed shrews**

Figure 8. Normalized throughput of TCP vs. spatial distributed shrews.

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



(a) Single TCP vs. 4 timely distributed shrews **(b) Five TCP vs. 4 timely distributed shrews**

Figure 9. Normalized throughput of TCP vs. timely distributed shrews.

Figure 8 and 9 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 utility, their frequency spectrum would possess the same properties.

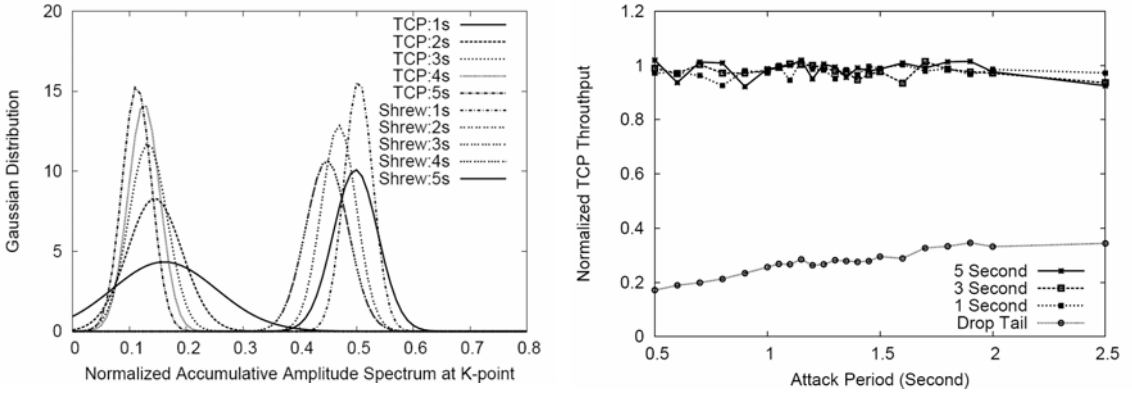
In other words, the shrew filtering mechanism is effective even if the attack is launched through larger number of streams with lower burst peak rate. In fact, if zombies use longer individual attacking periods, higher percentages of its energy will be located in the low frequency band we are monitoring.

4.2 Response Time

The response time is a critical parameter to evaluate the performance of our shrew-filtering algorithm. In general, how long the time a DDoS defense algorithm takes to detect whether malicious flows exists or not is the time used to monitor the traffic conditions. That 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 traffics, where we used the same sampling time of 5 second.

The only issue that affects the response time is how long the sampled series is required to make decision precisely. The DFT considers the sequence $x[n]$ with length N as a continuous periodic signal with a period N that $x[n]=x[n+rN]$ for any integer values of n and r [3]. 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 won't impact on our detection precision.

Figure 10(a) presents the distribution of NCAS at the K-point of TCP flows and the shrew streams. They are sampled from 1 sec. to 5 sec. As the sampling time decreases, the NCAS at the K-point of TCP flows scatters wider. Therefore, the probability of treat a legitimate TCP flows 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.



(a) Distributions of NCAS at K-point **(b) Throughput with sample time 1, 3 and 5 seconds**

Figure 10. The effects of different sample lengths on the TCP and shrew throughputs.

Table 3 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%).

Table 3. Confidence Levels of Different Sampling Time

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

We have simulated with sampling period of 1, 3 and 5 seconds, that is, to perform the detection using sampling series 1000, 3000 points and 5000 points since the sampling period is 1 ms. Figure 10(b) shows the throughput of five TCP flows under attack of four distributed shrew streams. Clearly, all sampling series achieved the throughput much higher than the Drop Tail algorithm. Statistically, there is no difference among them.

5. Conclusions

In this paper, we have proposed to cut off low-rate TCP-targeted DDoS attack flows by taking advantages of the periodicity properties of different traffics in the frequency domain. Our analysis and simulation show that more energy of low-rate shrew attacks is located in the lower frequency band comparing with the legitimate TCP flows. Using hypothesis test theory and Gaussian distribution model, we have shown that our shrew-filtering algorithm achieves pretty higher accuracy. Thus, our technique can block malicious shrew flows with high confidence level (> 99.9%), while exhibiting very low probability (< 0.1%) of blocking legitimate TCP flows.

One of the distinct advantages of our approach is that DFT and frequency domain analysis are standard *digital signal processing* (DSP) methods that could be implemented efficiently in hardware, thanks to the modern VLSI technology. Therefore, 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. Another advantage of shrew-filtering algorithm is to cut off the malicious shrew streams totally, which is similar to our MAFIC algorithm that block flooding DDoS attack flows [6]. In this manner, we minimized the influence shrew streams may have on legitimate flows.

There is one limitation in our shrew-filtering algorithm. Specifically, 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 (reported in [27]) indicated that high detection accuracy was achieved using a collaborative distributed detection mechanism.

In 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 [8], [9]. 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. The rationale is that 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] Ronald L. Allen, and Duncan W. Mills, "Signal Analysis: time, frequency, scale, and structure", Published by 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. C. 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] C.-M. Cheng, H. Kung, and K.-S. Tan, "Use of spectral analysis in defense against DoS attacks," *Proc. IEEE GLOBECOM*, Taipei, China, 2002.

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

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

[10] Jay L. Devore, and Nicholas R. Farnum, "Applied Statistics for Engineers and Scientists", Published by Duxbury Press, 1999

[11] M. Guirguis, A. Bestavros, and I. Matta, "Bandwidth Stealing via Link Targeted RoQ Attacks," *Proc. 2nd IASTED International Conference on Communication and Computer Networks*, Nov. 2004.

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

[13] X. He, C. Papadopoulos, J. Heidemann, and A. Hussain, "Spectral characteristics of saturated links," Under Submission, http://www.isi.edu/~johnh/PAPERS/He04a.html

[14] 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.

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

[16] 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.

[17] 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," *accepted to appear in the 2005 International Conference on Computer Networks and Mobile Computing (ICCNMC 2005)*, Zhangjiajie, China, August 2-4, 2005.

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

[19] R. Mahajan, S. Floyd, and D. Wetherall, "Controlling high-bandwidth flows at the congested router," *Proc. ACM 9th International Conference on Network Protocols (ICNP),* Nov. 2001.

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

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

[22] 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.

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

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

[25] Haibin Sun, John C. S. Lui, and David K. Y. Yau, "Defending Against Low-rate TCP Attacks: Dynamic Detection and Protection", *Proc. IEEE International Conference on Network Protocols* (ICNP), Berlin, Germany, October 2004

[26] X. Luo, and R. K. C. 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., February 2-5, 2005

[27] Yu Chen, Yu-Kwong Kwok, and Kai Hwang, "Collaborative Defense Against Periodic Shrew DDoS Attacks in Frequency Domain", submitted to *ACM Transactions on Information and System Security (TISSEC)*, May 2005

Biographical Sketches

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

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 computing systems. Presently, he leads the GridSec project supported by NSF/ITR program at USC. Dr. Hwang can be reached at **kaihwang@usc.edu.** The GridSec web site is **http://gridsec.usc.edu/**