# A Wavelet-Based Anomaly Detection for Outbound Network Traffic

Kriangkrai Limthong
The Graduate University
for Advanced Studies (Sokendai)
Tokyo 101-8430 Japan
e-mail: krngkr@nii.ac.jp

Pirawat Watanapongse
Computer Engineering Graduate School
Kasetsart University
Bangkok 10900 Thailand
e-mail: pw@ku.ac.th

Kensuke Fukuda
National Institute of Informatics /
PRESTO JST
Tokyo 101-8430 Japan
e-mail: kensuke@nii.ac.jp

***Abstract*—Monitoring and detecting network anomalies are indispensable activities for network administrators. Most anomaly detection techniques focus on inbound traffic (traffic from the Internet entering a customer network) rather than outbound traffic. However, anomalous inbound traffic patterns will be significantly different from anomalous outbound traffic. For network operators, outbound traffic is as important as inbound traffic because they can monitor unwanted activities in their networks to prevent it from affecting other networks.**

**In this paper, we propose a statistic-based anomaly detection method for outbound traffic. Our method involves wavelet-based analysis and a statistical distance calculation of 3 month-long traces on outbound traffic from the computer center in Kasetsart University, which had about 1,300 users per day. We added six types of synthetic incidents to four original protocol-based time series (TCP SYN, TCP SYN/ACK, ICMP, and UDP) and investigated ability of our method to detect these anomalies.**

**Our technique could discover short duration malicious behavior in a moderate volume of packets as well as long duration anomalous behavior in a small volume of packets. The experimental results include the detection accuracy and the false positive rates of several wavelet components, and they indicate that our technique is useful for detecting malicious and anomalous behavior in outbound traffic at a network edge.**

***Index Terms*—wavelet, statistical distance, anomaly detection, time series, outbound, network traffic**

## I. Introduction

The goal of network anomaly detection is to identify packets or flows which do not conform with established normal behavior. The forms and causes of network anomalies can vary enormously. Anomalous traffic may be symptomatic of attacks, outages, misconfigurations, flash crowds, etc., and almost all network anomalies impact network traffic. For example, distributed denial of service (DDoS) attacks and flash crowds can seriously congest a network with unwanted traffic. Some anomalies do not affect network traffic but have an emotional effect on valuable customers or users. They may be evidence of computer fraud or cyber crime. For these reasons, anomaly detection is a vital responsibility of network administrators.

Previous studies tried to develop detection techniques that would help network administrators to detect various abnormal behaviors in network traffic accurately and promptly. Most of these studies focused on detection techniques in inbound traffic (traffic sent from the Internet to a customer network) rather than outbound traffic (traffic sent from a customer network to the Internet). Unfortunately, the traffic patterns of anomalies in inbound traffic are significantly different from those in outbound traffic. For example, in a denial of service (DoS) attack, all flooding packets from uncounted sources are combined and sent to a target or victim network. It would difficult to use the same technique as the target network for detecting DoS attack packets sent by an internal attacker, because the volume of attack packets in outbound traffic at the source network would usually be smaller than the inbound traffic at the target network.

We propose a technique to assist network administrators in analyzing outbound network traffic. The method enables us to ascertain abnormal behavior caused by an internal attacker before our computer systems start to punish another network by transmitting the attack packets. Moreover, administrators can discover unusual behavior due to internal incidents, such as outages, misconfigurations, and flash crowds. We applied our method to 3 month-long traces that were measured in a large campus network. We added numerous synthetic incidents, namely port scans, DoS attacks, flash crowds, outages, misconfigurations, and viruses, to the four original traffic time series of these traces: TCP SYN, TCP SYN/ACK, ICMP, and UDP, in order to investigate the detection ability of our method.

Our contribution is to reveal the capability of wavelet components in detecting various types of strange behavior in outbound traffic. At around level 5 to 7 of the wavelet components, our detection technique can detect not only malicious behavior, such as port scans, DoS attacks, and viruses, but also non-malicious abnormal behavior, e.g. flash crowds, outages, and misconfigurations. Network administrators can use our technique to detect concentrated abnormal behavior in their networks.

In the next section, we explain the related work that applies signal processing and wavelet analysis to the task of anomaly detection. Section III explains our technique for detecting malicious and abnormal behavior in outbound traffic. Section IV describes the experiment to evaluate the detection accuracy and false positive rate of our technique. In section V, we show experimental results from our extensive empirical study. We conclude and mention future work in Section VI.

## II. Related Work

Many researchers have had the idea of using signal processing techniques to detect anomalies in network traffic. Such techniques can detect novel incidents and attacks that cannot be detected by using signature-based approaches. A signature-based approach like Snort® must be updated with new rules in order to recognize new anomalies in network traffic. This means such approaches cannot detect new attacks or zero-day attacks that would not be included in the current rules. Signal processing techniques, on the other hand, could be used to detect such attacks. Examples of signal processing techniques include wavelet analysis, entropy analysis, principal component analysis, and spectral analysis.

Ref. [1] introduced time series analysis for detecting aberrant behavior through network monitoring by using the Holt-Winters forecasting algorithm. Ref. [2] proposed a spectral analysis technique to distinguish between normal TCP network traffic and traffic that was dropped or rate-limited by DoS attacks. Ref. [3] used signal processing and an abrupt change detection technique in order to detect several anomalies in IP networks. Moreover, Ref. [4] applied spectral analysis to TCP flows so as to defend against reduction of quality attacks.

Network traffic entropy analysis has been used in Ref. [5] to detect network traffic anomalies including different kinds of SYN attacks and port scans. Ref. [6] also used an entropy based method to detect anomalies and worms propagating in fast IP networks like the Internet backbone. Furthermore, Refs. [7] and [8] used principal component analysis for diagnosing anomalies in network-wide traffic. The authors of Ref. [9] discussed the sensitivity of the parameters used in principal component analysis for network traffic anomaly detection.

Wavelet analysis is a well-studied signal processing technique for detecting anomalies in network traffic. There are many studies that apply wavelet transformations to network traffic. For example, Refs. [10]–[12] employed a wavelet approach to detect numerous anomalies in network traffic. Refs. [13]–[15] used wavelet transformations to discover DoS attacks, and Ref. [16] used a wavelet technique for proactive detection of network misconfigurations. In contrast to these studies using signal processing techniques, Refs. [17]–[19] instead expanded on other methods of statistics analysis or cumulative sums.

Almost all of the above work focused on inbound network traffic. In our work, we extend detection methods to cover anomalies in outbound network traffic. Our detection technique works in an online as well as offline manner. Moreover, it can detect irregular events, such as flash crowds, outages, and misconfigurations, that the other techniques may have trouble finding.

## III. Methodology

### A. General Method of Anomaly Detection

Our detection method has four major steps, as shown in Fig. 1. In the first step, we generate a time series from aggregate network traffic measured in terms of the number of packets per time interval. We filter the traffic according to particular network protocols, IP addresses or ports. The output from this step is a time series of aggregate network traffic with a 1 second sampling rate. Note that we chose the value of 1 second from a prior study [15] in which the smallest time interval for a wavelet transformation was 1 second.

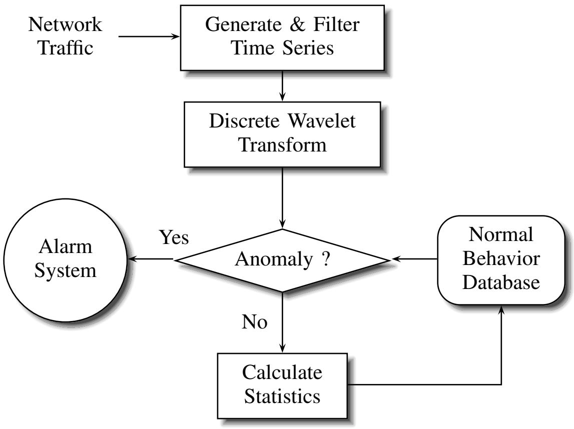

Fig. 1. The Block Diagram of Our Detection Method

In the second step, we use a discrete wavelet transform (DWT) to transform the original time series into two parts, a detail part and approximation part. The DWT acts like a high-pass filter to make the detail part and a low-pass filter to make the approximation part. The high-pass filter removes noise from the original signal, whereas the output from the low-pass filter essentially indicates the ordinary behavior of the signal. The low-pass filtered network traffic, i.e., the approximation part, corresponds more to anomalous behavior than the high-pass filtered network traffic, i.e., the detail part, does. Both filtered outputs are subsampled by 2 according to Nyquist's rule. We use a recursive pyramidal algorithm to make a multi-level wavelet transform from the DWT, as shown in Fig. 2. That means the approximation part is put through the DWT again to produce a second (third, fourth, fifth, etc.) level of detail and approximation parts. In addition, we use the Haar wavelet because of its simplicity and small computation cost.

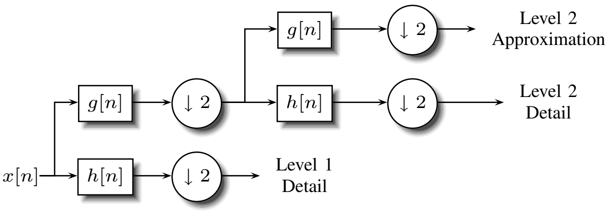

Fig. 2. Recursive Pyramidal Algorithm for Multi-Level Wavelet Transform

In the third step, we compare the approximation parts on the different levels with two baselines generated from a database containing traffic parameters exhibiting normal behavior. The baselines are the boundaries of the ordinary behavior of network users and were calculated from four attributes stored in the database; we will explain this database

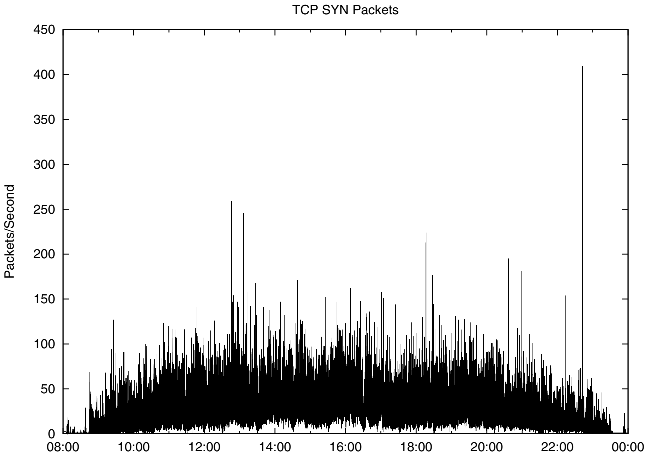

Fig. 3. Time Series of TCP SYN Packets from 9 July 2008

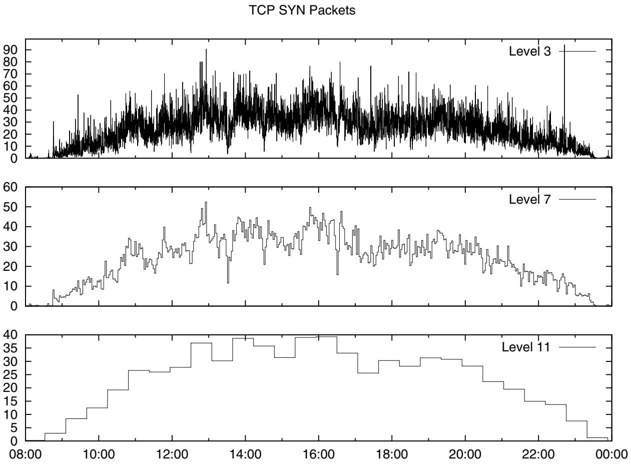

Fig. 4. Approximation Part Level 3, 7, 11 of TCP SYN Packets

in the next subsection. If the approximation parts of the outgoing network traffic fall outside of baseline region, the algorithm will send an alarm that the network traffic is likely to be anomalous. If the approximation parts of the outgoing network traffic do not have any anomalies, they are sent to the final step in order to calculate appropriate statistics. The following subsections explain the normal behavior database and the detection method.

In the final step, we construct a model representing the normal behavior of a given normal training data set. We store the statistical parameters of the normal behavior in the database. Then we merge the approximation parts of non-anomalous traffic that we found in the previous step with data in the normal behavior database by referring to the statistical parameters. Note that the output from previous step with anomalies will not be added to the database.

For example, we generated and filtered an original TCP SYN time series from the 9 July 2008 trace, as shown in Fig. 3. At the discrete wavelet transform step, we decomposed the original time series into levels of wavelet components. Fig. 4. shows examples of approximation parts at level 3 (Top), level 7 (Middle), and level 11 (Bottom). We compared each approximation part with the baselines on the same level of the wavelet components. If some of the approximation parts fall outside baseline region, the alarm is sent to network operators.

Fig. 5. (Top) shows a time series that simulated incidents at 9:00, 10:00, 11:00, 16:00, 17:00, and 18:00 by adding them to the attack-free time series on 9 July 2008. The two dashed lines in Fig. 5. (Bottom) illustrate the top and bottom of the baseline region. Some data points of the approximation part at level 7 are above the baseline region around 9:00, 10:00, and 13:00; and some data points of this part are below the baseline region around 16:00, 17:00, and 20:00. The alarm system activated at these times to notify network administrators about the anomalies.

Note that Figs. 3.-5. above are only meant to be of help to explain our detection method from the generate time series step to the anomaly detection step. They were not used to estimate the detection accuracy or false positive rate of our method.

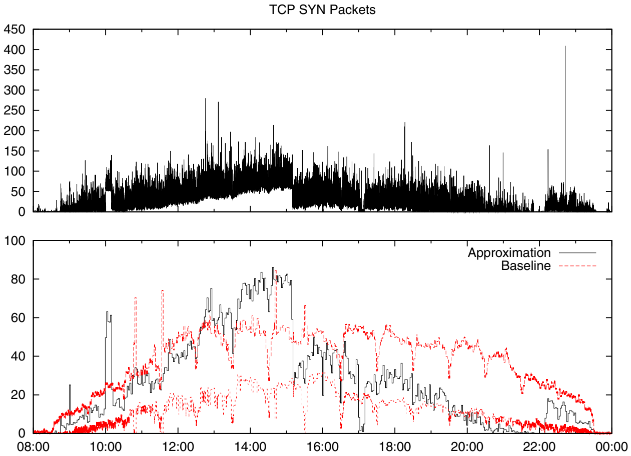

Fig. 5. Approxiamtion Part Level 7 with Baselines

## B. Normal Behavior Database

The normal behavior database stores the statistical parameters for generating baselines used to identify anomalies in the detection step. There are four parameters for each level $j$ of wavelet component, namely the time stamp ($t$), number of data ($n$), sum of data ($\Sigma x$), and sum of squares data ($\Sigma x^2$). The data, $x_{(i,j,t)}$, indicate the volume of packets on day $i$ of the approximation part on level $j$ with time stamp $t$. We used aggregate functions for these parameters because they are simple to calculate and require little storage. Accordingly, we do not need to store network traffic every day in order to create the baselines.

## C. Detection Method

We generate a baseline region for each level of wavelet component so as to compare the baselines with the outgoing network traffic. We derived the baselines from the statistical parameters stored in the normal behavior database. Eq. (1) represents upper and lower boundaries of the baseline region of approximation part at level $j$ with time stamp $t$. The constant $c$ is used for adjusting the confidence interval of the baselines. The variable $\overline{x}_{(j,t)}$ means the expected value of the random variable $x$ of the approximation part at level $j$ with time stamp $t$. Eq. (2) is expected value determined from $\Sigma x$ and $n$, where $n$ is the number of days. As shown in Eq. (3), we formulated $E[X^2_{(j,t)}]$ from $\Sigma x^2$ and $n$ in order to calculate the variance ($\sigma^2$), Eq. (4).

$$b_{(j,t)} = \overline{x}_{(j,t)} \pm c\sigma_{(j,t)} \tag{1}$$

$$\overline{x}_{(j,t)} = E[X_{(j,t)}] = \frac{\sum_{i=1}^{n} x_{(i,j,t)}}{n} \tag{2}$$

$$E[X^2_{(j,t)}] = \frac{\sum_{i=1}^{n} x^2_{(i,j,t)}}{n} \tag{3}$$

$$\sigma^2_{(j,t)} = E[X^2_{(j,t)}] - E[X_{(j,t)}]^2 \tag{4}$$

From these equations, we obtain the baseline boundaries $b_{(j,t)}$ for each level of the approximation part. If some data points in the approximation parts fall outside the baseline boundaries, we decide that the outgoing network traffic is anomalous. On the other hand, if the approximation parts fall inside the baseline boundaries, we integrate the approximation parts of the outgoing network traffic into the normal behavior database.

### D. Data Traces

In order to evaluate our method, we collected attack-free traces from the outbound router at the Internet service center of Kasetsart University for three months. This center is for college students, educators, and researchers so that they can ascertain advantageous information for their studies from the Internet. There are around about 1,300 users per day, and the service time is between 8:00 and 24:00 every day. Users cannot change or install any software in the computer client, and administrators provide appropriate software for all ordinary users. Moreover, administrators regularly update the virus signatures of the anti-virus software installed on all of the clients. Our experiment used data traces that were recorded from June to August 2008 as the attack-free network traffic.

## IV. Experiment

### A. Implementation

We implemented our method in C with libpcap and GNU scientific libraries. The average processing time for a 16 hour-long trace in text file format was about 1 second on a Linux PC (CPU:Intel® Core™ i7, Memory:6 GB).

TABLE I
Synthetic Incident Parameters

| Parameters | Values |
|---|---|
| Starting Time | Each hour from 9:00-23:00 |
| Volume (Packets/Second) | ± 5, 10, 15, 20, 25, 30, 35, 40, 45, 50 |
| Duration (Minutes) | 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 |

### B. Simulation

To evaluate the anomaly detection accuracy and false positive rate, we simulated six patterns of network traffic incidents and added them to the attack-free traces. The six types, namely port scan attacks, DoS attacks, flash crowds, outages, misconfigurations, and viruses, were simulated with the parameters listed in Table I.

First, we simulated different incidents of the same type by adding or subtracting 5 packets per second over a 1 minute duration. There was 1 incident per hour from 9:00 to 23:00 (a total 15 incidents). Then, we increased the volume of incidents by adding or subtracting from 10 to 50 packets per second. Next, we expanded the time duration from 1 minute to 10 minutes in 1 minute steps. We iterated this process 1,500 times for each day of data.

We chose the constant $c$ of the baselines such that there would be a 95% confidence interval in detecting on each level of the approximation part. We evaluated our technique on all approximation parts, from level 1 to level 11. However, in the next section, we shall only show the results for odd levels because of the page limitation.

## V. Results

The experimental results in Figs. 6.-9. show the detection accuracy and false positive rates of our detection technique. The detection rates and false positive rates were calculated from Eqs. (5) and (6), and are based on the number of alarms in a day.

$$detection\ rate = \frac{number\ of\ true\ alarms}{total\ number\ of\ alarms} \tag{5}$$

$$false\ positive\ rate = \frac{number\ of\ false\ positive}{total\ number\ of\ alarms} \tag{6}$$

From our traces, we collected the network traffic from 8:00 to 24:00 (57,600 seconds) on every day. Thus, the total number of alarms in one day was the number of time slots (57,600). Let us assume that a simulated incidents occurred between 8:00:00 and 8:09:59. The detection rate in this case equals the sum of the number of alarms from 8:00:00 to 8:09:59 and the number of alarms that did not occur from 8:10:00 to 23:59:59 divided by 57,600. Moreover, the false positive rate equals the number of alarms from 8:10:00 to 23:59:59 divided by 57,600.

Fig. 6. shows average percentile values of true detection rates for each level when we varied the volume of incidents from 5 to 50 packets per second. All of the results indicated that the detection rates increase when the volume of incidents rises. In particular, the detection rates at level 5 are quite higher

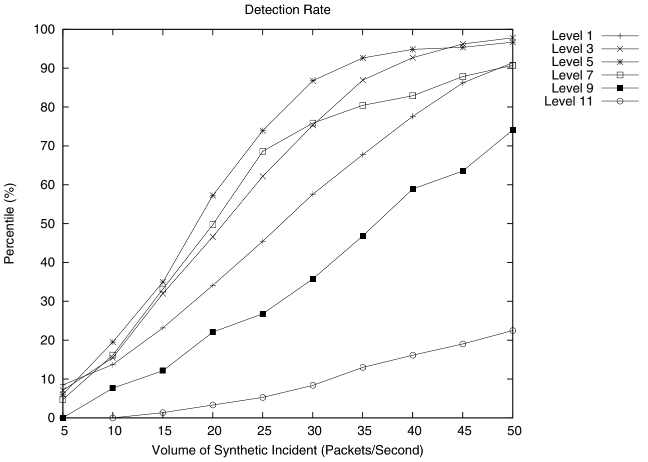

Fig. 6. Anomaly Detection Rates vs Volume of Synthetic Incidents

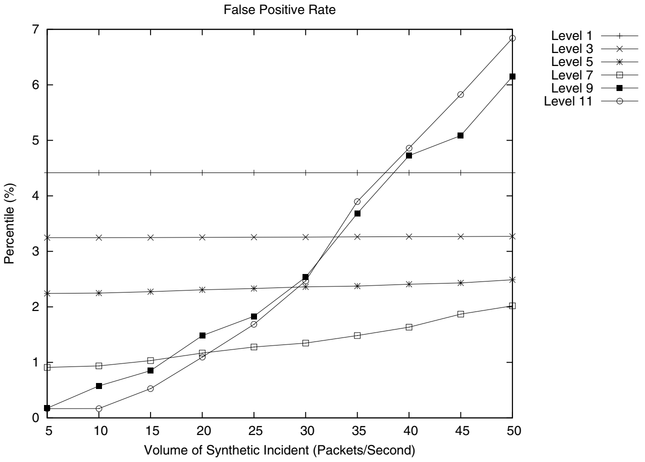

Fig. 8. False Positive Rates vs Volume of Synthetic Incidents

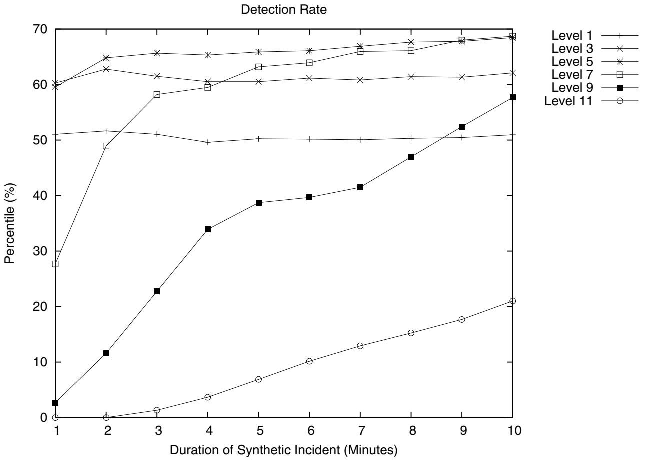

Fig. 7. Anomaly Detection Rates vs Duration of Synthetic Incidents

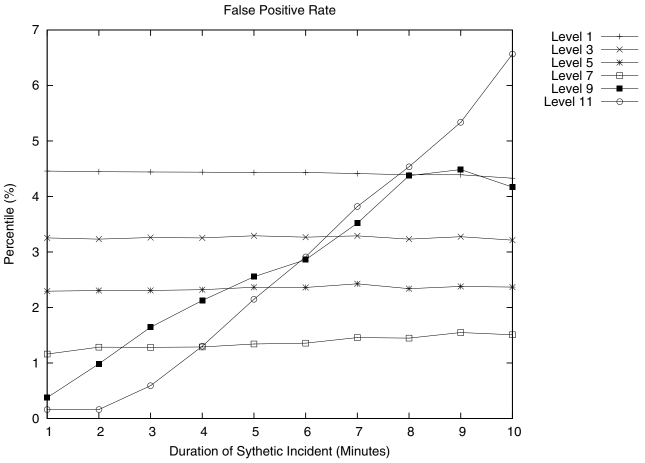

Fig. 9. False Positive Rates vs Duration of Synthetic Incidents

than at other levels. In addition, when we varied the duration of synthetic incidents from 1 to 10 minutes, the approximation part at level 5 still had higher detection rates than at any other level, as shown in Fig. 7.

We also evaluated the false positive rates of our detection technique, as shown in Fig. 8. and Fig. 9. When the volume of synthetic incidents was small, false positives at level 7 were not fewer in number than at higher levels such as level 9 or level 11, as shown in Fig. 8. However, the false positive values at level 7 were the smallest of levels when the volume of the incidents was more than 20 packets per second. Furthermore, when we varied the duration of incidents over 4 minutes, the false positive values at level 7 were still the smallest in this situation. When the volume of incidents was small and duration was short, the false positive rates at levels 9 and 11 were smaller than at other levels; however, the false positive rates increased right after the volume of incidents exceeded 20 packets per second or the duration exceeded 4 minutes.

In summary, if administrators employ our detection technique with a 1 second time interval on outbound network traffic, levels 5-7 of the discrete wavelet transform are suitable for anomaly detection. At level 5, the detection accuracy is high but the false positive rate is also high. On the other hand, the detection rates and false positive rates at level 7 are lower than those at level 5. Moreover, on the same day, we found that if the baselines are derived from less than 30 days worth of traffic data, the false positive rates show quite a large deviation. Meanwhile, if the baselines are derived from more than 30 days worth of traffic data will yield consistent false positive rates.

## VI. Conclusion and Future Work

We proposed an analysis technique to detect small volumes of anomalies in outbound network traffic. Our technique translated aggregate network traffic into a time series and applies discrete wavelet transform (DWT) to the time series so as to convert the original time series into wavelet components. We compared the wavelet components of outgoing traffic with baselines that stand for wavelet components of normal traffic. If the wavelet components of outgoing traffic fall outside the baseline region, our technique sends an alarm to network administrators that the outgoing traffic has an anomaly. The baselines are statistically calculated from the network traffic of previous days and consist of the time stamp, number of data, sum of data and sum of squares data. Our empirical results showed that this technique can usually find anomalies in low volumes of anomalous packets. Moreover, from this experimental data set, we found that the wavelet components at level 5-7 with a 1 second time interval are suitable for detecting anomalies in outbound network traffic.

Although our detection technique has been designed for outbound network traffic, it is possible to use it on inbound network traffic, because it can learn network behavior without network traffic archiving. Our next job will be to apply this technique to backbone traffic traces so that we can evaluate its detection accuracy on network-wide traffic. Furthermore, we would like apply our technique to an online traffic environment.

## Acknowledgment

This work was supported by the National Science and Technology Development Agency (NSTDA), and the Thailand Graduate Institute of Science and Technology (TGIST) under contract number TGIST 01-51-092. The authors would like to thank all of the anonymous reviewers for their excellent suggestions that have greatly improved the quality of this paper.

## References

[1] J. D. Brutlag, "Aberrant behavior detection in time series for network monitoring," in *LISA '00: Proceedings of the 14th USENIX conference on System administration*. Berkeley, CA, USA: USENIX Association, 2000, pp. 139–146.

[2] C.-M. Cheng, H. Kung, and K.-S. Tan, "Use of spectral analysis in defense against dos attacks," in *Global Telecommunications Conference, 2002. GLOBECOM '02. IEEE*, vol. 3, Nov. 2002, pp. 2143–2148 vol.3.

[3] M. Thottan and C. Ji, "Anomaly detection in ip networks," *Signal Processing, IEEE Transactions on*, vol. 51, no. 8, pp. 2191–2204, Aug. 2003.

[4] Y. Chen and K. Hwang, "Spectral analysis of tcp flows for defense against reduction-of-quality attacks," in *Communications, 2007. ICC '07. IEEE International Conference on*, June 2007, pp. 1203–1210.

[5] Y. Gu, A. McCallum, and D. Towsley, "Detecting anomalies in network traffic using maximum entropy estimation," in *IMC '05: Proceedings of the 5th ACM SIGCOMM conference on Internet Measurement*. Berkeley, CA, USA: USENIX Association, 2005, pp. 32–32.

[6] A. Wagner and B. Plattner, "Entropy based worm and anomaly detection in fast ip networks," in *Enabling Technologies: Infrastructure for Collaborative Enterprise, 2005. 14th IEEE International Workshops on*, June 2005, pp. 172–177.

[7] A. Lakhina, M. Crovella, and C. Diot, "Diagnosing network-wide traffic anomalies," in *SIGCOMM '04: Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications*. New York, NY, USA: ACM, 2004, pp. 219–230.

[8] B. I. P. Rubinstein, B. Nelson, L. Huang, A. D. Joseph, S.-h. Lau, N. Taft, and D. Tygar, "Compromising pca-based anomaly detectors for network-wide traffic," EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2008-73, May 2008. [Online]. Available: http://www.eecs.berkeley.edu/Pubs/TechRpts/2008/EECS-2008-73.html

[9] H. Ringberg, A. Soule, J. Rexford, and C. Diot, "Sensitivity of pca for traffic anomaly detection," *SIGMETRICS Perform. Eval. Rev.*, vol. 35, no. 1, pp. 109–120, 2007.

[10] P. Huang, A. Feldmann, and W. Willinger, "A non-instrusive, wavelet-based approach to detecting network performance problems," in *IMW '01: Proceedings of the 1st ACM SIGCOMM Workshop on Internet Measurement*. New York, NY, USA: ACM, 2001, pp. 213–227.

[11] P. Barford, J. Kline, D. Plonka, and A. Ron, "A signal analysis of network traffic anomalies," in *IMW '02: Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurment*. New York, NY, USA: ACM, 2002, pp. 71–82.

[12] W. Lu and A. A. Ghorbani, "Network anomaly detection based on wavelet analysis," *EURASIP J. Adv. Signal Process*, vol. 2009, pp. 1–16, 2009.

[13] L. Li and G. Lee, "Ddos attack detection and wavelets," in *Computer Communications and Networks, 2003. ICCCN 2003. Proceedings. The 12th International Conference on*, Oct. 2003, pp. 421–427.

[14] A. Dainotti, A. Pescape, and G. Ventre, "Nis04-1: Wavelet-based detection of dos attacks," in *Global Telecommunications Conference, 2006. GLOBECOM '06. IEEE*, 27 2006-Dec. 1 2006, pp. 1–6.

[15] K. Limthong, F. Kensuke, and P. Watanapongse, "Wavelet-based unwanted traffic time series analysis," in *Computer and Electrical Engineering, 2008. ICCEE 2008. International Conference on*, Dec. 2008, pp. 445–449.

[16] A. Magnaghi, T. Hamada, and T. Katsuyama, "A wavelet-based framework for proactive detection of network misconfigurations," in *NetT '04: Proceedings of the ACM SIGCOMM workshop on Network troubleshooting*. New York, NY, USA: ACM, 2004, pp. 253–258.

[17] J. Mirkovic and P. Reiher, "D-ward: a source-end defense against flooding denial-of-service attacks," *Dependable and Secure Computing, IEEE Transactions on*, vol. 2, no. 3, pp. 216–232, July-Sept. 2005.

[18] V. A. Siris and F. Papagalou, "Application of anomaly detection algorithms for detecting syn flooding attacks," *Computer Communications*, vol. 29, no. 9, pp. 1433 – 1442, 2006, iCON 2004 - 12th IEEE International Conference on Network 2004.

[19] A. Tartakovsky, B. Rozovskii, R. Blazek, and H. Kim, "A novel approach to detection of intrusions in computer networks via adaptive sequential and batch-sequential change-point detection methods," *Signal Processing, IEEE Transactions on*, vol. 54, no. 9, pp. 3372–3382, Sept. 2006.