

# Modeling Network traffic as Images

Seong Soo Kim and A. L. Narasimha Reddy

Department of Electrical Engineering  
Texas A&M University  
College Station, TX 77843-3128, USA  
{skim, reddy}@ee.tamu.edu

**Abstract**—This paper presents a network measurement approach to represent samples of network packet header data as frames or images. With such a formulation, a series of samples can be seen as a sequence of frames or video. This enables techniques from image processing and video compression to be applied to the analysis of packet header data to reveal interesting properties of traffic. We show that traffic images can reveal sudden changes in traffic behavior or anomalies. Using a combination of visual modeling and trace-driven simulation, we evaluate how the design factors impact the representation of dynamic network traffic. In particular, we study the impact of sampling rate and retained DCT coefficients on the network traffic data representation.

**Keywords**—Image difference analysis, Network anomaly detection, Network traffic, Time-varying imagery, Video analysis, Network Security, Network Traffic Modeling and Characterization.

## I. INTRODUCTION

At present, attacks on Internet infrastructure, in the form of denial-of-service (DoS) floods, worms and other forms, have become one of the most serious threats to the network security. If efficient visual and analysis tools are available to network administrators, it could become possible to detect the attacks, anomalies and to appropriately take action to contain the attacks before they have had much time to propagate across the network. In this paper, we describe an approach that represents network traffic as images to enable traffic-analysis and visualization.

A number of tools such as FlowScan [5], Cisco's FlowAnalyzer, and AutoFocus [1], have been used to study and classify traffic on the network based on usage and the employed protocols. These tools have been effectively utilized for traffic engineering and postmortem anomaly detection. However, rigorous real-time analysis is needed for detecting and identifying the anomalies so that mitigation action can be taken as promptly as possible. Some of these tools are based on the volume of traffic such as byte counts and packet counts. When links are congested, it is possible to always observe a fully utilized link without giving further information about possible changes in network traffic. Sophisticated low-rate attacks [2] and replacement attacks, which don't give rise to noticeable variance in traffic volume, could go undetected when only traffic volume is considered. The tools that collect and process individual flow data may not scale to high-speed links. While earlier work illustrated characteristics of network traffic flow anomalies at flow level [6], our approach tries to look at aggregate packet header data in order to improve scalability.

Our work here brings techniques from image processing and video analysis to visualization and real-time analysis of traffic patterns. Various forms of approaches have been traditionally utilized for detecting scene changes in image processing [9, 10, 11, 12].

Our approach passively monitors packet headers of network traffic at regular intervals and generates images of this packet header data. These images are analyzed to find whether any abnormalities are observed in the traffic. Recent studies have shown that the traffic can have strong patterns of behavior over several timescales [3], and our previous work has shown the possibility of analysis of wide-sense stationary (WSS) property in network traffic [4]. By observing the traffic and correlating it to the previous states of traffic, it may be possible to see whether the current traffic is behaving in an anomalous manner. In case of anomalous traffic such as flash crowds and DoS attacks, the usage pattern of network may be changed and peculiarities could be represented in visual images. When anomalies are detected, further analysis can characterize the anomalies by their nature into several categories (random attack, targeted attack, multi-source attack, portscan attack etc.) and help in mitigating the attacks.

In this paper, we will report on our measurements conducted on real traces of traffic at two networks, the University of Southern California [14] and KREONet2 [16]. This paper will make the following significant contributions: (a) employing packet header data as images for traffic visualization, (b) providing guidance for visual representation of traffic, and (c) employing image processing and compression techniques for efficiently storing and processing such visual data.

## II. NETWORK TRAFFIC AS IMAGES

### 2.1 Network Traffic

We employ packet header data collected at a network access point for traffic analysis. This data includes source, destination addresses, port numbers, traffic volume in bytes, packets and other useful information. Each sample of data is represented as an image. For example, a pixel in such an image may represent traffic volume originating from each source address. Similarly, a pixel in the image may represent traffic volume in bytes or packets going to a destination, or in a flow between a (source, destination) pair. Similarly, the image may represent the number of port numbers or flows seen between a (source, destination) pair.

Such a representation allows simple visualization of traffic data as each sample is seen as a frame in a video sequence. Traffic data can then be efficiently stored through such

techniques as video compression. Multiple pieces of data, IP address, port numbers and flows, can be represented as different colors of an image leading to unified treatment and multidimensional analysis of traffic data.

Image processing and video analysis techniques can be applied to such a representation to decipher patterns of traffic. Scene change analysis could reveal sudden changes in traffic patterns leading to traffic anomaly detection. For example, single source attacking multiple destinations will be represented by horizontal lines in the (source, destination) traffic volume image. Similarly, a DDoS (Distributed Denial-of-Service) attack against a single destination would be represented by vertical lines in the (source, destination) image. A portscan attack would be similarly visible in the port number based images.

## 2.2 Visual Representation

We illustrate our approach with specific examples of image generation and analysis. There are several possibilities for generating images over address domain, port number domain, protocol domain etc. and for utilizing various metrics for generating each pixel in such a domain through the use of traffic volume in bytes, packet numbers, number of flows etc. We use packet counts in the address domain in the following example.

For each address,  $a_m$ , in the traffic, we count the number of packets,  $p_{mn}$ , sent in the sampling instant,  $s_n$ . We can define normalized packet count in sampling point  $n$  as (1).

$$p(m, n) = p_{mn} / \sum_m p_{mn} \quad (1)$$

We suggest a simpler alternative with a data structure with a smaller amount memory as used in [4] for reducing the storage and computation complexity over  $2^{32}$  discontinuous address space from  $O(n)$  to  $O(\lg n)$ . This data structure consists of 4-by-256 array “ $count[4][256]$ ”. Each 1-dimensional array expresses one of the 4 bytes in an IP address structure. A location  $count[i][j]$  is used to record the packet count for the address  $j$  in  $i^{th}$  field of the IP address. The packet counts on the link are recorded to the corresponding position of each IP

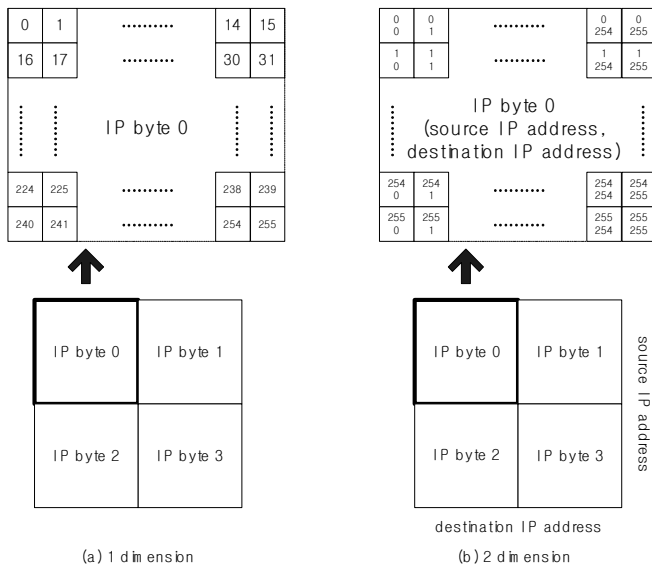


Figure 1. The visualization of network traffic signal in IP address

address byte-segment and normalized packet count is quantized and represented as shown in (2).

$$p_{ijn} = \frac{count[i][j][n]}{\sum_{j=0}^{255} count[i][j][n]}, \quad i = 0,1,2,3 \quad j = 0, \dots, 255 \quad (2)$$

Each resultant normalized packet count is converted to corresponding pixel intensity in image representation of traffic. We arrange the normalized packet count of the 256 entries of the each byte in to a 16-by-16 square for visual representation at the sampling point. Due to 4-byte structure of IP address, we have 4 such 16-by-16 squares as a frame for the source and destination addresses respectively as shown in Fig. 1(a). Instead of 16-by-16, with 256-by-256 squares, we can express the normalized values for a (source, destination) pair simultaneously as in Fig. 1(b). Here the intensity of the pixel is directly proportional to the normalized packet count [15].

Similarly, a flow-based visual representation ( $F_{ijn}$ ) employs the number of flows instead of packet counts over the address domain. The number of flows could vary from the norm at the outbreak of network attacks. Through monitoring changes in the number of flows, it is feasible to perceive the anomalies. We used the triple of source address / destination address / destination port as the definition of a flow.

The data structure (or image) processes each byte of the IP address independently. This allows classification of anomalous traffic based on a targeted address range, especially useful in a local-preferential spreading approach employed by some worms. However, our approach could introduce errors during the identification of the complete IP addresses of attackers or victims when the segments of different addresses from each quadrant are combined. Apparent patterns like solid lines could not, statistically, result from the simple result of a union of disconnected dots, but reflect the nature of anomalous traffic. In particular, a horizontal line such as in IP byte 3 of Fig. 4(f) is highly likely to be generated through a hostscan of several destinations by a single source machine. If the number of large distinct flows in byte 3 domain is  $k$  ( $\leq 256^2$ ) and the flows are distributed uniformly, the probability of generating single dot (on the line) is  $k/256^2$  and of forming  $m$  consecutive dots is  $(k/256^2)^m$  respectively. So the probability of forming the horizontal line from unrelated flows is  $(k/256^2)^{256}$ .

A horizontal line in such an image indicates that a source is accessing multiple destinations (with proximate addresses), for example during worm propagation. This indicates a hostscan of destination machines by a single source with a high probability. A vertical line indicates that several sources are accessing a single destination. This could indicate the accesses to popular servers (such as google.com) or DDoS attacks being staged from multiple machines (with proximate addresses) on a single destination.

## III. REQUIREMENTS FOR REPRESENTING NETWORK TRAFFIC AS IMAGES

### 3.1 Sampling rates

Recent study has shown that Gaussian approximation should work well for aggregated traffic if the level of aggregation in

the number of traffic sources and observed time scales is high enough so that individual sources are swallowed according to Central Limit Theorem [7]. Our previous work has shown the possibility of analysis of WSS property in network traffic [4]. Based on these results, if we appropriately select the sampling rate for generating images, we could acquire normally distributed and stationary images. For discriminating current traffic situation based on the stationary property, we should select a sampling frequency for deriving the most stable images.

Fig. 2 shows the inter-frame Mean Square Error (MSE) by (3) of various sampling intervals. As the sampling interval exceeds 10 seconds, the inter-frame MSE decreases significantly. When sampling intervals are small, the variation of traffic information in successive frames is not negligible. Larger sampling intervals lead to larger latencies for analysis and detection of anomalies. Based on these two observations, we choose 30-60 second samples of traffic data.

$$MSE = \frac{\sum [I(i,j) - I'(i,j)]^2}{N^2}, \quad (3)$$

for intra - frame  $\begin{cases} I(i,j) \text{ is original image} \\ I'(i,j) \text{ is reconstructed image} \end{cases}$   
for inter - frame,  $I(i,j)$  and  $I'(i,j)$  are consecutive images

Fig. 3 illustrates the effects of sampling periods by the type of network traffic. It is sampled only for 10 seconds of every T seconds ( $10/T$  sec), where T varies from 10 seconds to 540 seconds. In normal traffic, as shown in Fig. 3(a), the inter-frame MSE remains at a nearly similar level regardless of the sampling periods. It shows that the traffic is stationary in normal times and the selection of the sampling period is not crucial. However, in attack traffic in Fig. 3(b), the MSE increases significantly with increasing sampling periods. It means that the traffic changes dynamically with time in attack times and the sampling period is a critical factor.

### 3.2 Visual modeling Network Traffic as Images

Automated self-propagating codes could be generally classified by their spreading approaches and convergence to the destination into single targeted, semi-random targeted and random targeted attacks. Single destination attack can be considered as a special case of semi-random attack. We generally consider traffic as in normal, in semi-random and in random attack modes.

#### 3.2.1 Visual patterns in normal network traffic

Fig. 4 shows the visual measurement of  $F_{ijn}$ , flow

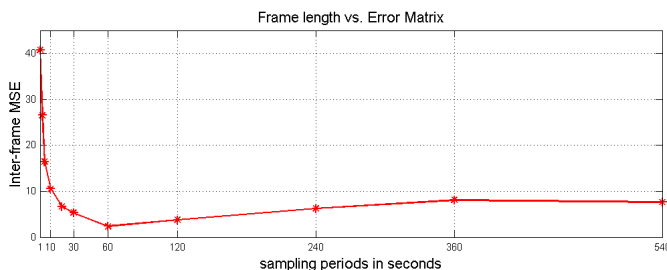


Figure 2. The relationship between MSE and Sampling rates.

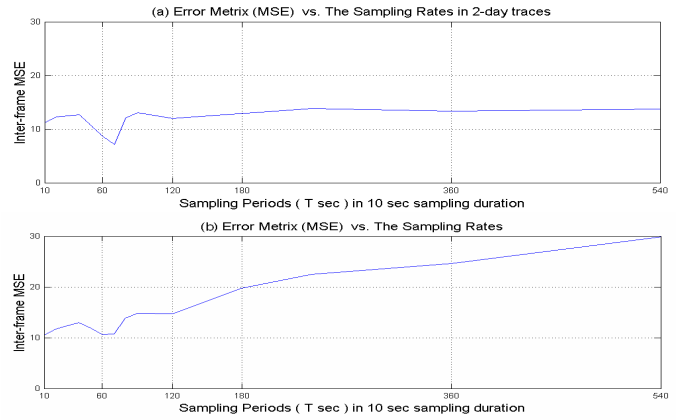


Figure 3. The relationship between sampling rates and nature of the traffic which are ambient traffic in (a) and attack traffic in (b).

distribution in the source / destination IP address domains in diverse modes of traffic, based on the excerpts from KREONet2 traces. We exemplify the flow-based visual representation here.

Fig. 4(a) through 4(c) visually illustrates the normalized flow numbers during the absence of anomalous traffic. Fig. 4(a) shows source addresses, 4(b) destination addresses, and 4(c) shows the traffic in source-destination domain. The aggregate traffic does not form any regular shape due to dispersibility of traffic of various and numerous flows in time and space. The color and darkness of each pixel points up the intensity of traffic (the number of flows) of corresponding IP address(es). The black lines in Fig. 4(f) and 4(i) explicitly illustrate more concentrated traffic than the orange lines in Fig. 4(c).

#### 3.2.2 Visual patterns in semi-random targeted attacks

Semi-random targeted attacks target hosts with the same address prefix. Code-Red was a representative local-preferential worm. Fig. 4(d) through 4(f) visually illustrates the measurement of  $F_{ijn}$  in the source/destination IP address domains during a semi-random target attack.

From Fig. 4(e) destination IP addresses, a specific area of IP byte2 data structure is shown in a darker shade. It illustrates that the current traffic is concentrated on the (aggregated) single destination or the subnet. It was observed that this darker portion shifted as time progressed during the attack.

From the 3<sup>rd</sup> and 4<sup>th</sup> byte of Fig. 4(f), it shows that a specific source, i.e. an attacker, monopolizes network traffic in the form of a horizontal stripe. The attacker employed large number of flows while it sequentially changed the source port and only the 3<sup>rd</sup> and 4<sup>th</sup> bytes of victims' IP addresses at a specific destination port. From the viewpoint of statistical analysis, because the difference in the number of flows between attackers (or victims) and legitimate users is significant, the variance shows much higher values than normal traffic cases.

#### 3.2.3 Visual patterns in random targeted attacks

Random targeted attacks use a uniform random probe strategy, SQL Slammer was a typical random propagation worm. Fig. 4(g) through 4(i) visualizes the traffic during a random attack.

From Fig. 4(h), bytes 1, 2 and 3 of the destination address data structure are shown in darkness. It means that, in general, traffic is behaving in an inconsistent pattern and attacks are targeting randomly generated destinations. Actually the two compromised attackers employ a large number of flows while they randomly change the victims' IP addresses with specific source/destination port. Because almost all of the destination addresses are exploited in hostscan attacks, the distribution is inordinately homogenous such that variance in pixel intensity shows much lower values relative to normal traffic pattern. Fig. 4(i) shows that two specific sources, visible through 2 horizontal lines in each quadrant, scan all possible destinations.

In an actual implementation, our tool offers these visual measurements as a real-time motion picture. It can help the network operators recognize the traffic transition trends.

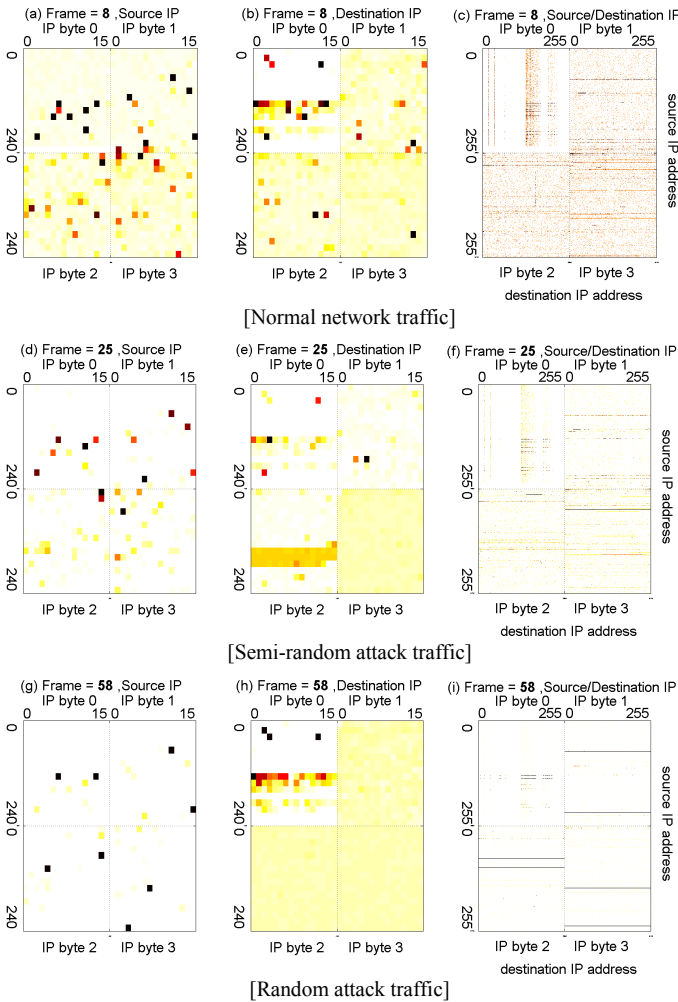


Figure 4. Visual measurement of the *Flow-based* Network traffic. The (a), (d) and (g) sub-pictures show the intensity of network traffic of the source IP addresses in each frame. The (b), (e) and (h) sub-pictures show that of destination IP addresses. The color of each pixel shows the intensity of traffic at the source or destination, and the descending order of intensity is black, red, orange, yellow and white. The (c), (f) and (i) sub-pictures show the intensity of network traffic of the (source, destination) pair in 2-dimension simultaneously. The x-axis corresponds to the distribution of the destination IP addresses, and y-axis does that of the source addresses. In each quadrant, source and destination addresses consist of 256-by-256 pixels.

## IV. IMAGE PROCESSING FOR NETWORK TRAFFIC

For efficient storage and processing, the images could be compressed. For a 2-dimensional image of (source, destination) domain with 16 bits per pixel, memory of about 0.5Mbytes/frame is required. We could employ image compression using the 2-D discrete cosine transform (DCT) to reduce the requirements.

We employ 8-by-8 blocks for DCT. We transform the 16 blocks of Fig. 1(a) using DCT. The DCT tends to concentrate information in the upper-left corner of the DCT matrix. The inverse DCT can be performed using a subset of the DCT coefficients. Among the 8-by-8 DCT coefficients, we select only the  $n$  most significant DCT coefficients in the zigzag pattern by discarding coefficients close to zero for compression. We can find out how many coefficients are necessary to create a reasonable approximation of the original traffic image. Fig. 5(a) shows the relationship, the error matrices and the number of retained DCT coefficients in 3 kinds of representative traffic images. Without losing the properties of traffic, we can choose the suitable number of the DCT coefficients according to system resources. The DCT coefficients could then be quantized, coded, and stored for future analysis.

### 4.1 Validity of intra-frame DCT

The decreasing rates in Fig. 5(a) depend on the characteristics of traffic such as ratio of frequency components. It is worthy to note that the MSE and the decreasing rates are closely related with the variances of pixel intensity in traffic. Transformation Coding Gain (TCG or  $G_T$ ) by (4) measures the amount of energy packed in the low frequency coefficients. So the higher TCG leads to smaller MSE and higher compression. This is useful because if most of information is stored in the first  $n$  coefficients, then we can reduce the data storage size by retaining only the first  $n$  coefficients.

$$\text{DCT transform matrix } [A]_{i,k} = a_i \cos \frac{\pi(2k+1)i}{2N}, \text{ for } i, k = 0, \dots, N-1,$$

$$\text{, with } a_0 = \sqrt{1/N}, a_i = \sqrt{2/N}, \forall i \neq 0$$

$$\sigma_n^2 = \text{diagonal elements of } A\Theta A^T, \text{ where } \Theta \text{ is covariance matrix}$$

$$\rho \text{ is correlation coefficient}$$

$$G_T = \frac{1}{N} \sum_{n=0}^{N-1} \sigma_n^2 \sqrt{\prod_{n=0}^{N-1} \sigma_n^2} \quad (4)$$

With high TCG, such as in a random traffic image as shown in Fig. 5(a), most of the information can be packed within a few coefficients. With only one coefficient, we can see that DCT transforms of random traffic reduce the MSE by more than 50%. When we increased the number of coefficients to 3, MSE of most traffic falls by 50%. Traffic images lack significant spatial redundancy and contain more high frequency components than normal images. This results in slower decrease of the Normalized MSE and hence requires cautious consideration of DCT coefficient selection.

### 4.2 Inter-frame differential coding

If traffic has a low TCG, some significant portion of the original information is spread out into higher frequency

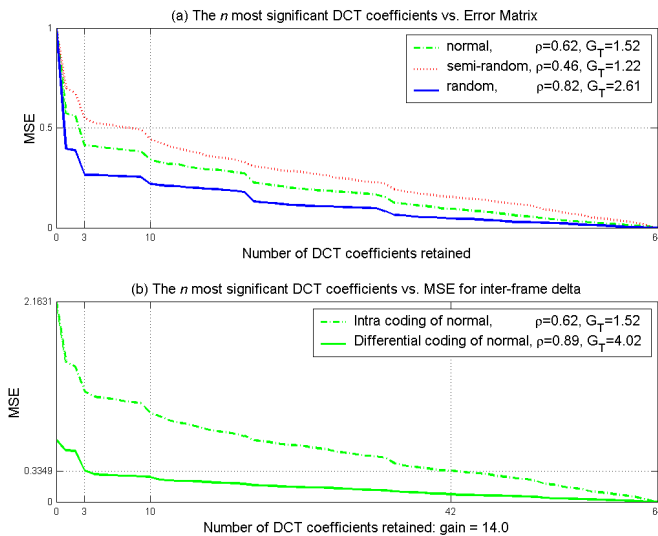


Figure 5. The relationship between intra-frame MSE and DCT coefficients.

coefficients. For improving the TCG, we can exploit the deltas of consecutive frames. Fig. 5(b) shows the effects of the delta coding through an increased TCG and a gain of about 14.0 compared to non differential coding in the number of required DCT coefficients.

### 4.3 Effect of Sampling rates on DCT coefficients

The effect of sampling rates on the number of retained DCT coefficients is illustrated in Fig. 6. A sampling rate of 60 seconds maintains the minimum intra-frame MSE over the entire range of retained DCT coefficients. When the sampling period is small, say 1 second, the effect of intra-frame compression is relatively insignificant. When the sampling period is 10 seconds, the intra-frame MSE is close to the optimal value seen at a sampling period of 60 seconds. Based on these results on intra-frame and inter-frame MSE, we conclude that a sample period of 30-120 seconds is a good choice.

## V. CONCLUSIONS AND FUTURE WORK

In this paper, we have presented an approach that represents traffic data as images or frames at each sampling point. Such an approach enabled us to view traffic data as a sequence of frames or video and allowed us to apply various image processing and video analysis techniques for studying traffic

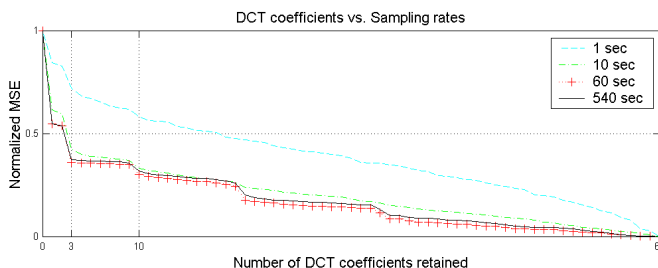


Figure 6. The relationship between DCT coefficients and Sampling rates in normalized intra-frame MSE.

patterns. We have demonstrated our approach through an analysis of traffic traces obtained at networks. Our results show that our approach leads to useful traffic visualization and analysis.

We plan to study detection and identification approaches along multiple dimensions of IP packet header data such as addresses, port numbers, and the number of flows. We plan to study the impact of our approach on real-time anomaly detection by deploying our tool in a production network.

## ACKNOWLEDGMENT

We are very grateful to Man Hee Lee and Dr. Okhwan Byeon at KISTI for their help in accessing KREONet2 traces.

## REFERENCES

- [1] C. Estan, S. Savage and G. Varghese, "Automatically Inferring Patterns of Resource Consumption in Network Traffic", in *Proc. of ACM SIGCOMM*, Karlsruhe, Germany, August 2003.
- [2] A. Kuzmanovic and E. Knightly, "Low-Rate TCP-Targeted Denial of Service Attacks", in *Proc. of ACM SIGCOMM*, Karlsruhe, Germany, August 2003.
- [3] P. Barford, J. Kline, D. Plonka and A. Ron, "A Signal Analysis of Network Traffic Anomalies," in *Proc. of ACM SIGCOMM Internet Measurement Workshop (IMW)*, Marseille, France, November 2002.
- [4] Seong Soo Kim, A. L. Narasimha Reddy and Marina Vannucci, "Detecting traffic anomalies through aggregate analysis of packet header data", in *Proc. of Networking 2004*, LNCS vol. 3042, pp. 1047-1059, Athens, Greece, May 2004.
- [5] Dave Plonka, "FlowScan: A Network Traffic Flow Reporting and Visualization Tool", in *Proc. of the USENIX 14<sup>th</sup> System Administration Conference*, New Orleans, LA, December 2000.
- [6] P. Barford and D. Plonka, "Characteristics of Network Traffic Flow Anomalies," in *Proc. of ACM SIGCOMM Internet Measurement Workshop (IMW)*, October, 2001.
- [7] Jorma Kilpi and Ilkka Norros, "Testing the Gaussian approximation of aggregate traffic," in *Proc. of ACM SIGCOMM Internet Measurement Workshop (IMW)*, Marseille, France, November 2002.
- [8] B. Krishnamurthy, S. Sen, Y. Zhang and Y. Chen, "Sketch-based Change Detection: methods, Evaluation, and Applications," in *Proc. of Internet Measurement Conference (IMC)*, Miami, USA, October 2003.
- [9] Dan Lelescu and Dan Schonfeld, "Statistical Sequential Analysis for Real-time Video Scene Change Detection on Compressed Multimedia Bitstream", *IEEE Transactions on Multimedia*, vol. 5, issue 1, pp 106-117, 2003.
- [10] H. Zhang, A. Kankanhalli, and S. W. Smoliar, "Automatic partitioning of Full-motion Video", *Multimedia Systems*, vol. 1, no. 1, pp 10-28, 1993.
- [11] R. Lienhart, C. Kuhmunch, and W. Effelsberg, "On the Detection and Recognition of Television Commercials", in *Proc. Of the International Conference on Multimedia Computing and Systems*, pp 509-516, Ottawa, Canada, 1997.
- [12] K. Shen and E. J. Delp, "A fast Algorithm for Video Parsing Using MPEG Compressed Sequences", in *IEEE Conference on Image Processing*, pp 252-255, 1995.
- [13] Gyaourova, A., C. Kamath, and S.-C. Cheung, "Block matching for object tracking," LLNL Technical report, October 2003. UCRL-TR-200271.
- [14] A. Hussein, J. Heidemann, and C. Papadopoulos, "A framework for classifying denial of service attacks", in *ACM Sigcomm*, Aug. 2003.
- [15] Seong Soo Kim and A. L. Narasimha Reddy, "A Study of Analyzing Network traffic as Images in Real-Time", in *Proc. of IEEE INFOCOM*, Miami, Florida, USA, Mar. 2005.
- [16] KREONet2 (Korea Research Environment Open NETwork2). Available : <http://www.kreonet2.net>