Lightweight Traffic Monitoring and Analysis Using Video Compression Techniques

Marat Zhanikeev[1] and Yoshiaki Tanaka[2,3]

[1] School of International Liberal Studies, Waseda University
1-6-1 Nishi-Waseda, Shinjuku-ku, Tokyo, 169-8050 Japan
[2] Global Information and Telecommunication Institute, Waseda University
1-3-10 Nishi-Waseda, Shinjuku-ku, Tokyo, 169-0051 Japan
[3] Research Institute for Science and Engineering, Waseda University
17 Kikuicho, Shinjuku-ku, Tokyo, 162-0044 Japan
maratishe@aoni.waseda.jp, ytanaka@waseda.jp

Abstract. Traffic analysis based only on IP address is a new research area where traffic anomalies can be detected by studying clusters of IP addresses extracted from traveling packets. Such analysis is normally spatial and needs IP addresses to be put in a multi-dimensional map. This paper proposes a novel method that converts such maps to 2-dimensional graphical form and applies video compression techniques to create MPEG-2 VBR movies where frames are individual snapshots of IP space in time. The paper proves that this combination is suitable for traffic monitoring and detection of DDOS attacks as well as large-scale traffic anomalies caused by social phenomena.

Keywords: traffic analysis, traffic monitoring, anomaly detection, IP space, video compression.

1 Introduction and Research Target

Traditional traffic analysis and especially anomaly detection relies on traffic flow information exported by NetFlow. There are several other standards and software products which compare to NetFlow, but the underlying concepts are the same, - packets are sampled and joined into flows, which are exported to a remote location where the traffic analysis takes place.

It is common for traffic analysis methods to "unwrap" flows in order to extract other information. IP space extracted from flows could be used to study communication patterns or one might need to count how many sources communicate to a single destination.

Whenever NetFlow-like technology is used as basis for traffic analysis, performance overhead becomes important. First, it takes time to aggregate raw packets into flows. Secondly, it takes time and traffic overhead to collect flow records from a network device for analysis. Overhead also occurs when almost any information it extracted from IP flows.

On the other hand, many traffic anomalies exhibit themselves in IP space. The term "IP space" throughout this paper is used to describe the totality of

C.S. Hong et al. (Eds.): APNOMS 2009, LNCS 5787, pp. 92–101, 2009.

IPv4 addresses found in trace. IP space is normally a temporal properly because a limited time interval is used to collect IP addresses before they are analyzed by graphical methods introduced in this paper.

The two immediate examples of traffic anomalies which “show up” in IP space are DDOS and Flash Crowds. The former often uses spoofed source IP addresses which have strong effect on IP space created by otherwise normal traffic. Flash Crowds, on the other hand, are local by nature, - Flash Crowds are created as the result of temporarily increased network activity directed to a single destination. Sources of these communication pairs are not completely random but rather “social” and tend to create local clusters in IP space.

There is a new emerging kind of social traffic phenomena which is generated by distributed botnet attacks. Botnets are zombie computers which are controlled by hidden attackers and tend to become active at roughly the same time, thus, resembling Flash Crowds. There are, however, subtle differences between the two.

If IP space is converted to a 2-dimensional graphical form, traffic analysis can be enhanced by a very well developed area of video compression. As will be shown in the paper, traffic analysis and video compression have common goals as long as the data in question comes in graphical form. Specifically, MPEG-2 was found to be the most suitable compression format given that it uses variable bitrate (VBR), which can be used to represent the degree of change occurring in graphical data.

This paper uses only IPv4 addresses [1]. Although it is predicted that at the current level of consumption, IPv4 address pool will run out by the end of year 2011 [2], it is still unclear as to whether IPv6 is the only alternative. In fact, a few scenarios are considered in research, most of which are planning to extend the use of IPv4 addresses. On the other hand, conversion of IPv4 addresses into graphics images is not new. There are several research projects in this area, one of which is CAIDA IPv4 WHOIS Map project [5]. However, to the extent of authors’ knowledge no methods have yet been proposed to explore these graphical images for the purpose of detecting traffic anomalies.

This paper proposes a method which converts IP addresses collected from raw packet headers to a graphical form, stores them as individual graphics image files and finally compresses them into a MPEG-2 variable bitrate stream where each frame corresponds to graphics images of IP space. The proposed method exploits the fact that MPEG-2 encoding process achieves high compression ratio by performing interframe compression where it leaves in intermediate frames only the parts of graphical image which are different compared to the graphical image from the previous frame. The effectiveness of interframe compression translates almost directly into the notion of video stream bitrate, where the larger the difference between frames the higher is the bitrate.

When IP address space is converted into graphical image, stream bitrate of encoded still images will directly represent change and will make it possible to detect several important traffic anomalies. Additionally, the paper proves that the proposed method of traffic monitoring is effective especially in cases when traffic anomalies are rooted in social phenomena. With the Internet rapidly becoming a fully distributed and decentralized endeavor, traffic anomalies will exhibit more social features in the future.

Apart from the proposed method, the paper also presents details about its practical implementation, which was made possible with several open source software products. This in turn facilitates the discussion of practical applications of the proposed method at the end of this paper.

2 Construction of 2D Graphical IP Maps

The trick with video compression is to keep in mind that each frame is analyzed by the encoder based on 8x8 or in some cases 16x16 blocks of pixels. This block size imposes rigid requirements on the frame size in pixels. Normally, values are required to be divisible by 8 or 16. Specifically, MPEG-2 requires frame dimensions to be divisible by 16.

The numbers 8 and 16 also make perfects sense in the world of traffic analysis. When prefixes are used to analyze IP addresses, the two most commonly used prefixes are 8/ and 16/. The "forward slash" notation will be used throughout this paper to represent the length of the prefix. 8/ means that only 8 bits from the head of IP address are used. Naturally, 16/ is twice longer.

The 8/ prefix also bears geographical meaning in most of its 256 values. Originally and still sometimes today, IANA used to distribute the 8/ addresses based on geographical location. When it comes to 16/ prefix, address distribution there is not so regular. Statistically, only from 5 to 20 percent of each 8/ address is occupied permanently. Some of the rest are either dynamic or reserved addresses which are active much less often. In any case, 16/ addresses are normally distributed among large private or public organizations including ISPs.

The method to position IP address in 2-dimensional space is shown in Fig.1. First, 8/ and 16/ of an IP address are converted to simple integer values and used to position the block in 2048x1536 space. If to denote 8/ as a and 16/ as b, then $x = 8a$ and $y = 6b$.

The reason for allocating 8 pixels for each bit in first byte of each IP address is simple two-fold:

- if only one pixel were allocated for each value in 8/ prefix, the resulting image would be only 256 pixels wide, which is too small;
- horizontal cluster of 8 pixels can be used to present additional statistics about each address in 8/; in this paper it is used to create visual difference among active IP addresses with high packet counts and those which send/receive only a few or even one packet.

The vertical structure positioned by the value of 16/ prefix also contains 6 pixels for each value in 16/. However, here each pixel is mapped to a bit in 22/ prefix map, thus, visualizing activity in addresses within 6 more bits of each 16/ prefix. So, position of each pixel vertically is $y = 6b + c$ where c takes values from 0 through 6 and represents bits between 16/ and 22/. Prefixes from 20/ to 22/ are

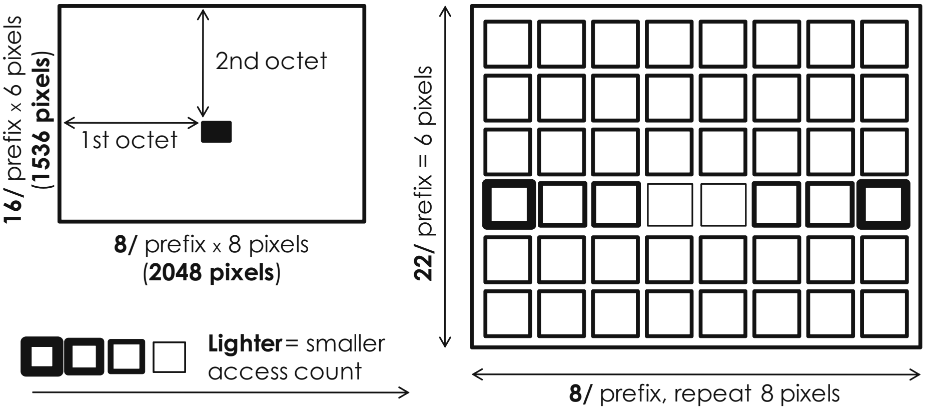

Fig. 1. Method used to position a single IPv4 address on a 2-dimensional map using 8/, 16/ and 22/ prefixes as integer values

at the longest prefixes found in global BGP advertisements and can be used in packets traveling through both intra- and inter-ISP backbone links.

In addition to the above positioning rules, some coloring rules are used as well. The proposed method does not actually use color, so, instead, the level of lightness is used on the scale from black to white colors. All graphics and video processing algorithms are much more sensitive to changes in brightness than in color. The word "color" hereinafter always means "brightness" of white color. Rules are as follows:

1. Default color of each block is black. The first packet that carries the IP address of the bock will color its horizontal line in white color. All successive packets with the same IP address will gradually make the color "lighter" until a threshold of 128 (half-grey) is reached where the color saturates with no further change.
2. When 2 or more bits between 16/ and 22/ are active, they affect each other's colors by making them brighter. Each neighbour doubles the count of all other 22/ bits in the block.
3. The coloring of each horizontal line is exponential where ends are colored in the color defined by packet count directly while the color of each inner pixel is defined by the number remaining after a right bitwise shift operation (division by two). In short, small packet counts fade out faster than large ones.

The above coloring rules exist with a sole purpose of creating graphical diversity. Given that video compression treats all frames as visual images, the more diversity exists in them the more effective is the monitoring technique based on the bitrate of the output video stream. Naturally, other methods to graphically present IP address blocks in 2-dimensional maps may exist, but for the sake of simplicity this paper only presents one.

3 Construction of MPEG-2 Stream from IP Maps

Number of pixels in a graphical image is the main contributor to the file size. The more pixels, the larger the graphics image file. Although this may sound as a potential method for anomaly detection, in practice the size of the image cannot be used for analysis because similarities in strings of pixels are aggressively exploited by image compression techniques.

On the other hand, all video compression technology cares about is human perception. This is why most compression techniques transform graphical images to frequency domain and use only a portion of first transform coefficients to recreate (decode) the image at receiver side. In plain words, the more local "abnormalities" an image (or computed difference between frames) contains, the higher the probability this local area is "noticed". Taken the pixel representation model presented in Fig.1 earlier, detectability of individual IP addresses placed in graphics image can be estimated by the following equation:

$$D = p \; var_{j=i-1,i,i+1}(2^{n_j} e^{-(c_j-1)}), \; i = 0..255. \tag{1}$$

In (1), c stands for packet count to/from the target IP address, n is the number of "vertical" neighbours, i.e. IP addresses which share the 22/ prefix, var stands for mathematical variance among values for the three horizontal neighbour 8/ addresses, and p represents the probability that the previous or the next IP address on 16/ prefix is active at the same time. According to statistics [2], p is normally between 0.8 and 0.95 and can be dropped from the equation for its insignificance.

The scope of this paper cannot accommodate elaboration on (1), but a few notes are in place. First, $c = 1$ guarantees highest detectability unless the 8/ neighbours on both sides are the same. Secondly, increasing values of c will exponentially decrease detectability unless 8/ neighbours exhibit very different trends. In general, detectability is higher when a particular IP address has packet counts and 22/ neighbours different from those of its 8/ neighbours.

The visualization in Fig.2 is called "mirror bar plot" and is used when two sequences are synchronized in time and have to be displayed together both for visual clarity and to enable visual comparison. Both upper and lower sequences in such plots contain only positive values. Values in all such plots in this paper are also normalized since real values convey no meaning.

The notion of bitrate comes directly from the processing stage in video compression called quantization. In quantization, a frame is analyzed and compared to previous (key frame in most cases) in order to find differences in image. If there are no differences, bitrate remains low since it requires little throughput to transmit little change. However, major changes in graphical content will result in larger encoded frames which in turn will require high bitrate (throughput) to transmit them. In plain words, video compression looks into the graphical content of frames rather than frame size and can detect changes in content even when frame size remains roughly the same.

The dynamics of bitrate are clearly shown in Fig.2 where the start-up phase of traffic monitoring is shown. In start-up, frames have no pixels in them in

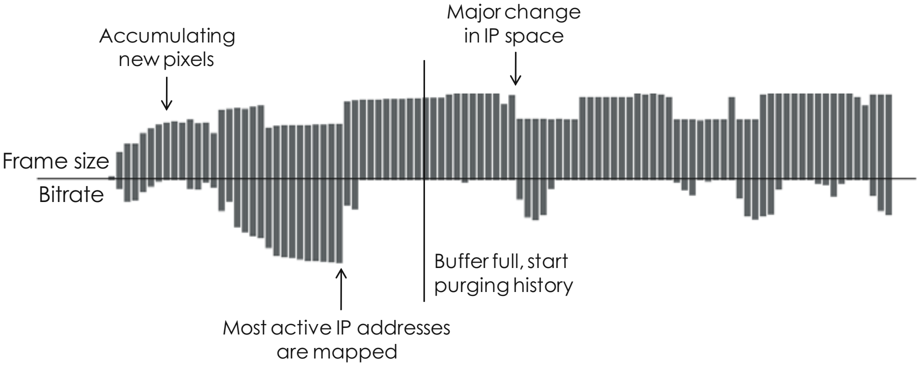

Fig. 2. Example of a start-up phase. History is set to 50 frames, each frame accumulates 1s of traffic.

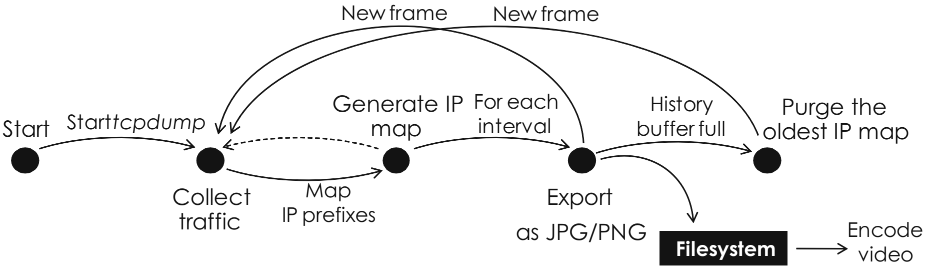

Fig. 3. The program logic starting from raw IPv4 packet collection and up to export of individual frames as JPEG images

the beginning which is why the frame size will gradually increase as pixels are accumulated. Bitrate also reacts accordingly by increasing until a relatively balanced state is achieved. In Fig.2, the balanced state is reached even before frame buffer was filled and started to purge old frames. From that point on, bitrate is relatively low and reacts only to local changes in graphical content.

The program logic used to convert IP space into graphical form is shown in Fig.3. The process starts at *tcpdump* which is used for collecting raw packets from network interfaces and is constantly looping between traffic collection and IP map generation as each packet header has to be inspected and either its source of destination address is mapped to the image. Only source addresses are mapped in the proposed method, while destination addresses may be a better choice in other cases.

At the end of each collection interval, graphical image is saved to a JPEG file and is later used to encode video. If frame buffer is full at the time, oldest frames are discarded at the end of each collection interval. The reason for the buffer is simple. Without the buffer, graphical images would be too random and

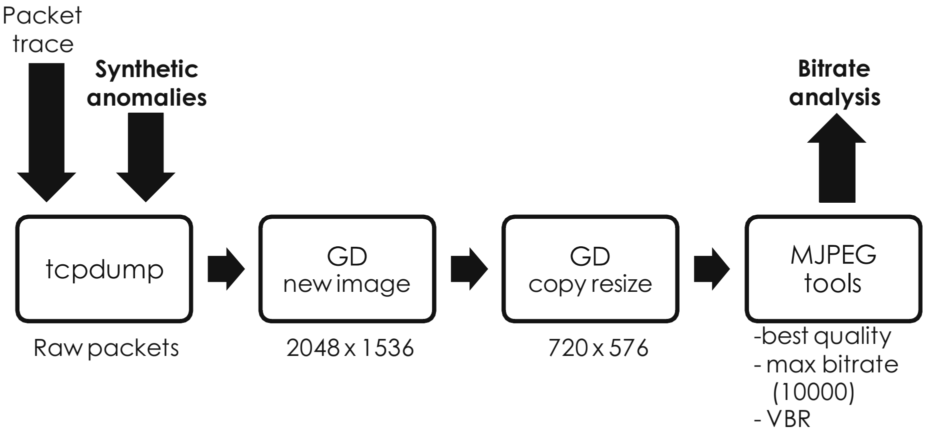

Fig. 4. Overall process from packet collection to analysis of MPEG-2 stream generated from IP maps

would always contain major changes in each frame. To counteract this, an n-frame history is retained at all times and are merged together before graphical content is saved to a file.

Finally, Fig.4 contains the overall process including video compression and analysis of end results. Two open source tools are used in the process. First, GD library [7] is used to generate JPEG images from IP maps stored by program internally. It is also used to resize the image from its full 2048x1536 frame size down to 720x576, which is standard for MPEG-2 streams. Resulting JPEG images are used as input into *mjpegtools* software [8] which converts multiple still JPEG images into a MPEG-2 variable bitrate stream.

Traffic monitoring itself is performed only on the resulting video stream. Although bitrate is not the only metric which can be used to monitor traffic, for the sake of simplicity only bitrate is used in this paper. In networking terminology, bitrate is directly equal to throughput.

4 Evaluation Results

This section considers practical uses of the proposed traffic monitoring framework. Specifically, DDOS attacks and FlashCrowds [4] are evaluated and compared.

When DDOS attacks happen, it is common to encounter IP addresses which are deliberately spoofed by attackers. This scenario can easily be synthesized. As per Fig.4 above, the process starts from packet collecting using tcpdump. However, to have a controlled and reiterable environment, packet traces from the WIDE traffic archive [3] were used instead. On top of using a recent trace from F-samplepoint of the WIDE network (average throughput 70 to 120Mbps), DDOS was synthesized and fed into the tcpdump stream. A simple DDOS synthesis

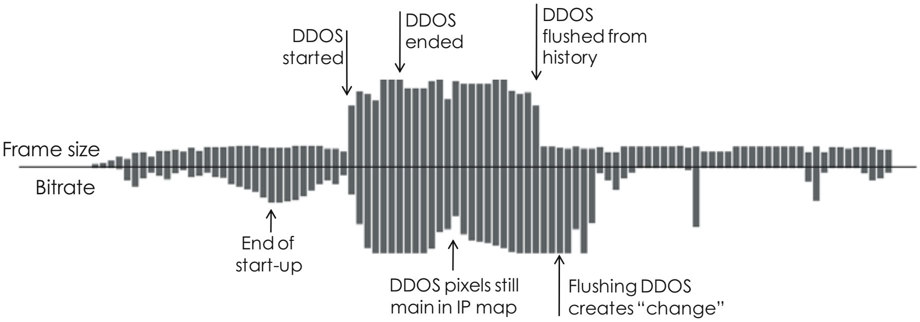

Fig. 5. Example of how DDOS attacks reveal themselves in video stream

was used where about 30% of IP address pool at the selected time in trace was artificially randomized.

Fig.5 clearly detects DDOS both in frame size and in bitrate. DDOS synthesis was started past the start-up phase so that it is not affected by it. Although DDOS attacks normally have short lifespan, they will affect video stream as long as they remain in the frame buffer. Bitrate will also be affected by it the whole time. Additionally, when DDOS is finally flushed from frame buffer, it will generate abrupt change in IP space of the monitor and will cause bitrate to rise again. In plain words, DDOS will reveal itself in 2 pulses, - one when the attack itself takes place and the other when it it flushed from the buffer.

Fig.6 shows how a Flash Crowd anomaly would reveal itself in stream bitrate. Synthesis of a Flash Crowd is much more complicated given that one has to mimic social behaviour. In the above example, the number of 8/ random prefixes gradually grew up to 100 with exponential rate of 1.18, while each 8/ prefix hosted up to 100 random 16/ prefixes growing with the rate of 1.15, each containing up to 200 random 32/ prefixes (IP addresses) growing at the rate of 1.18. The exponential growth rate helps Flash Crowd to grow, reach its peak, and exponentially fade out. The rates were tuned over several experiments and resulted in around 40 frames in the synthetic trace.

As can be seen from Fig.6 Flash Crowd-like behaviour does not exhibit drastic changes which is why frame sizes both with and without Flash Crowd presence are almost identical. However, the change in bitrate can still be detected and comes from the fact that many local clusters of IP addresses light up in the image and pass change detection thresholds when encoded in the video stream.

As far as the overhead (time consumption) in the overall process from Fig.4 goes, clearly, the packet collection phase is the most performance-hungry. Once the IP address is extracted from the packet, the process is relatively easy and is entirely based on bitwise operations in software. Bitwise operations cause only mild computational overhead since they happen directly in low-level registers in CPU instead being the result of complex computations. Also, since

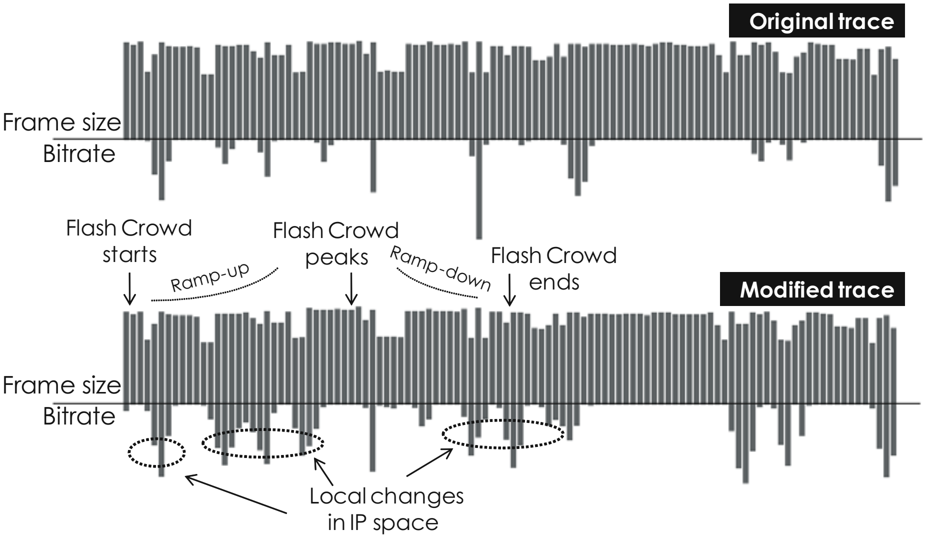

Fig. 6. Example of how Flash Crowd anomaly can reveal itself in stream bitrate

8/, 16/, and 22/ prefixes are treated as integers and used directly to address an element of an array in memory, overhead in positioning IP address in graphical image is minimal.

The use GD library and *mjpegtools* is also beneficial for the overall process since these tools have been developed for many years and are optimized for best performance. This means that these parts of the overall process from Fig.4 are also optimal in their performance. In general, the implementation of the proposed method is limited only by the speed with which packets can be collected from interface, which means that *tcpdump* is the only performance bottleneck in the process.

5 Conclusions

This paper proposed a novel method of traffic monitoring based entirely in graphics. IP addresses are extracted from raw packets and mapped to a 2-dimensional graphical image. Series of images are then encoded into MPEG-2 video stream using variable bitrate and highest possible quality. The use of bitrate which is directly affected by changes in graphical content of images and, thus, the original IP space, facilitates many traffic monitoring tasks at the client end. Although this paper only analyzed the bitrate of the video stream, MPEG-2 streams contain other information that can be used for traffic monitoring as well. For example, quantization coefficients for each intermediate frame are also stored in the stream and can be used to localize the change within the image.

The implementation of the proposed method was made possible using existing open source tools. In fact, the two tools which were used to generate graphical and video content are optimized to the best of performance due to high demands on the part of compression of video content. This area of research has been actively developed for many years and leaves little space for improvement.

The performance of the process used by the proposed method is very lightweight compared to NetFlow-based traffic monitoring. One particular performance boost comes from the fact that the proposed process does not have store IP addresses in hash tables as NetFlow does. Instead, bits from IP addresses are used directly to index elements within permanent array structures. In case of NetFlow, the necessity to contain a hash table of all flows is the biggest performance challenge.

The use of MPEG-2 streams can also have practical uses in network monitoring as well. Since bitrate is called throughput in networking world, when traffic experiences little change, video stream will require little throughput. On the other hand, drastic changes in traffic will raise bitrate of video stream and will require more throughput between network element and NMS. In whole, this is the description of a real-time monitoring system with very modest demands for the volume in meter-NMS communications. This makes it possible to implement the proposed system directly on network devices and stream monitoring video over the web.

In the future work, authors plan to look further into other models of translating IP addresses into blocks of pixels. Depending on the monitoring target, models other than the one proposed in this paper may offer better performance. Additionally, authors plan to address the need for benchmark traffic traces which can be used to test the performance of a given detection target. Such benchmarks already exist in the area of video compression where MPEG-2 encoding is tested on various graphical content. Similar benchmarks can be created using sample traffic traces with anomalies already inside. Development of such benchmark test cases is necessary for the future development of the proposed method.

References

1. Rekhter, Y., Moskowitz, B., Karrenberg, D., de Groot, G.J., Lear, E.: RFC 1918. Address Allocation for Private Internets (1996)
2. IPv4 Address Report, `http://www.potaroo.net/tools/ipv4/`
3. MAWI Working Group Traffic Archive, `http://tracer.csl.sony.co.jp/mawi/`
4. Jung, J., Krishnamurthy, B., Rabinovich, M.: Flash Crowds and Denial of Service Attacks: Characterization and Implications for CDNs and Web Sites. In: WWW Conference, Hawaii, USA, pp. 532–569 (2002)
5. IPv4 WHOIS Map,
   `http://www.caida.org/research/id-consumption/whois-map/`
6. Lakhina, A., Crovella, M., Diot, C.: Characterization of Network-Wide Anomalies in Traffic Flows. In: Internet Measurement Conference, Italy, pp. 201–206 (2004)
7. GD Graphics Library, `http://www.boutell.com/gd/`
8. MJPEG Tools, `http://mjpeg.sourceforge.net/`