IEICE TRANS. INF. & SYST., VOL.E92–D, NO.1 JANUARY 2009

PAPER

A Framework for Detection of Traffic Anomalies Based on IP Aggregation

Marat ZHANIKEEV[†a)], ***Member and*** **Yoshiaki TANAKA**[††,†††], ***Fellow***

SUMMARY Traditional traffic analysis is can be performed online only when detection targets are well specified and are fairly primitive. Local processing at measurement point is discouraged as it would considerably affect major functionality of a network device. When traffic is analyzed at flow level, the notion of flow timeout generates differences in flow lifespan and impedes unbiased monitoring, where only n-top flows ordered by a certain metric are considered. This paper proposes an alternative manner of traffic analysis based on source IP aggregation. The method uses flows as basic building blocks but ignores timeouts, using short monitoring intervals instead. Multidimensional space of metrics obtained through IP aggregation, however, enhances capabilities of traffic analysis by facilitating detection of various anomalous conditions in traffic simultaneously.

key words: *traffic analysis, IP aggregation, anomaly detection, performance monitoring, network management*

1. Introduction and Related Research

Traffic analysis is normally associated with offline processing of large bulk of data collected from a remote measurement point in network. Processing this data locally at the measurement point itself is not an option as the measurement point normally would reside on a router or other switching device and running processing tasks on it would cause serious deterioration of performance. Besides, that very device could collect the traffic continuously, which is another reason to process traffic at another location.

Traffic data normally comes in form of either raw packet traces which can be collected from a network node remotely using a meter MIB [1] or locally using a *tcpdump*-like tool, or traffic flows, which are merely statistics collected over a sequence of packets that traversed from an original IP address and port number at source to an original IP address and port number at destination. Flow-based monitoring is defined in NetFlow protocol originally developed by Cisco [2].

The main shortcoming of both traffic capture approaches is the need to analyze results in offline mode, especially in view of constantly growing capacity of network links. Besides, neither *tcpdump*, nor *NetFlow* are capable of monitoring the entire traffic on any particular backbone link due to memory and performance restrictions. Instead, both tools have a pre-run phase when they compile "the ruleset", - a set of rules that define which packets or flows are to be monitored while all the remaining traffic can be ignored. The ruleset is hardcoded into the operation of each tool and cannot be changed at runtime. In fact, this part of each tool is a target of multiple research works that strive to increase the effectiveness of filtering.

This paper proposes a new multidimensional space for traffic analysis based on IP aggregation. Individual IP addresses are aggregated into traffic nodes which are represented by aggregate statistics. Multiple dimensions are created automatically by counting IP addresses in aggregated traffic nodes, counting nodes themselves, and performing several other simple operations on IP aggregation space. The use of variable-length prefix masks further deepens the capability of such multidimensional analysis to detect all common traffic anomalies even when they occur simultaneously.

The physical meaning of aggregation used in this paper is merely a number of bits from the end of an IP address that are masked, thus making all IP addresses with the same remaining prefix to appear the same. By applying masks of various lengths, referred to in this report as mask level or masks depth, sources can be grouped depending on the hierarchical structure of their physical connectivity. Dynamics of such clusters as a function of mask depth, as well as patterns in distribution of traffic volume and other metrics among clusters, proved to be useful for various purposes in traffic analysis ranging from online traffic monitoring to detection of anomalous conditions.

The use of IP aggregation is not unique and has been used in various network-related technologies for many years. Some technologies, like BGP, use IP aggregation intrinsically as an efficient method of storing many IP addresses in memory while retaining capability of fast search through the list. Also, since the design of the Internet is intrinsically hierarchical, IP aggregation is a natural way to view traffic characteristics.

IP aggregation is also used in traffic analysis although in a way different from that proposed by this paper. For example, the research in [3] uses the hierarchical view of traffic in order to study distributional characteristics of network attacks. In particular, this paper focuses on Flash Crowds and DoS attacks in the contexts of the general web and CDNs. Although the method used in this paper is closely related to IP aggregation, the analysis itself is performed in the com-

Manuscript received March 10, 2008.
Manuscript revised August 3, 2008.

[†]The author is with School of International Liberal Studies, Waseda University, Tokyo, 169–0051 Japan.
[††]The author is with Global Information and Telecommunication Institute, Waseda University, Tokyo, 169–0051 Japan.
[†††]The author is with Research Institute for Science and Engineering, Waseda University, Tokyo, 162–0044 Japan.
a) E-mail: maratishe@asagi.waseda.jp

DOI: 10.1587/transinf.E92.D.16

mon space where packet count, byte count and flow count form the dimensions.

Another research in [4] is also very close in the research target to this paper. The paper uses bit trees to represent traffic and analyze how traffic is distributed in the automatically formed tree. Instead of using bit trees, this paper proposed the use of variable-length prefixes to aggregate traffic. Since wide-area traffic in the Internet is significant in the range around 12-21 bits, the use of prefix is more efficient than generation of a bit tree. The two methods, however, achieve similar results in the end. Also, the above range of prefix significance could be different depending of where the traffic originates from and could be subject of change in each particular case. The point the authors are making is that only a small range of 32-bit IP addresses is significant for traffic analysis based on IP aggregation. This subject will be covered later in this paper.

The most prominent rival of IP aggregation-based traffic analysis methods is the analysis based on data mining within the common metric space. This space includes such common aggregate traffic metrics as byte count, packet count, and flow count, with several other less popular metrics. Such research is based on the notion that those three metrics are not in direct correlation. This is, in fact, true for some traffic anomalies as proved by the line of research in [5], [6], and [7]. The component analysis (PCA) is used by this research to mine data provided by the three above metrics in order to extract information on various anomalies. The reality has it, however, that the more anomalies are mixed in the traffic at the same time and the larger is the volume of traffic the more difficult it is to perform a reliable detection using PCA on a set of primitive metrics. IP aggregation is used exactly for this reason - to provide more low-level metrics for multi-dimensional analysis.

Another reason why IP aggregation is used is to alleviate the load imposed by online traffic analysis. Traditional packet-level analysis is so greedy for resources that it normally required offline analysis of data. In that case the data is collected by very simple and light packet capturing engine and is analyzed by another software entity later in offline mode.

At the flow level, the load is somewhat lighter since the number of entities simultaneously updated at the arrival of every packet is relatively small and contains flow-level metrics, such as packet counts, byte counts and others. So, basically, instead of storing each packet you only have to store a few integer numbers for each flow, normally defined uniquely by using the 5-tuple data, i.e. source IP, source port, destination IP, destination port, and protocol name.

Whenever there is load on one hand and performance requirement on the other, rulesets are used to overcome the difficulty of online analysis. Even such primitive software tools as *tcpdump* allow the use of packet filters based on a ruleset applied at the beginning of operation. Rules are normally compiled for yet better performance and are, therefore, normally static for the whole duration of operation.

The art of rulesets and online analysis was further perfected in the software tool called *SNORT* [8] which is the leading intrusion detection system in the public opinion today. It can operate both at the packet and flow level and has a complex system of defining rulesets for problems you might require *SNORT* to detect. The design of the tool itself and its overall effectiveness bring it very close to an authentically online anomaly detection tool if not for the native shortcomings of both packet and flow analysis - packet analysis is redundant and inevitably offline while flow analysis has to deal with the issue of finding a tradeoff between short flow timeouts and memory efficiency. Flow timeout is the upper threshold of the interarrival time between two successive packets in the flow above which the flow is split into two subsequent flows with a gap in between.

Finally, a similar multidimensional research is performed in [13]. Similarly to this paper, the authors in [13] also accept the fact that both *NetFlow* and *tcpdump* offer insufficient dimensionality in raw data. IP aggregation is thus considered a method to add more dimensions to data. However, there is a very tangible difference between the proposed and the research in [13]. Authors in [13] use IP aggregation in a very limited way, mostly to separate traffic into crude *clusters* of traffic. For example, traffic with low port numbers would be clustered first, and then separated further by a sequence of such classifications. In nature, this method is closer to general classification based on decision trees.

On the other hand, the proposed method does not use IP aggregation as a means of boosting dimensionality of data, but rather create a brand-new multidimensional space which can be used to easily reveal traffic anomalies. Attacks constitute only a subset of network anomalies while many other positive and negative traffic artifacts are recognized in traffic analysis research today. The flexibility of the proposed method allowes for detection of all possible traffic artifacts without the need to rerun traffic collection.

More importantly, the proposed method exploits differences in aggregation results with varying aggregation prefixes. Not only this is an additional dimension in analysis data, it considerably boosts the precision of detection results. Some practical cases when varying aggregation prefixes are used will be consdered further in this paper.

This paper argues that IP aggregation provides a metric-rich multidimensional space which can be easily subdued to multi-step analysis light enough to be conducted online. While the performance issues are not considered in this paper, in its full-fledged form, IP aggregation-based analysis will not require the notion of flows and will not use timeouts, instead generating the IP aggregation space directly from traffic. If compared on the scale of performance load, such space would be equal or even less a resource hog than flow-level aggregation.

2. IP Aggregation

Since any traffic analysis research is bound to have practical implications, the findings in this paper are verified on packet traces obtained from the repository provided freely

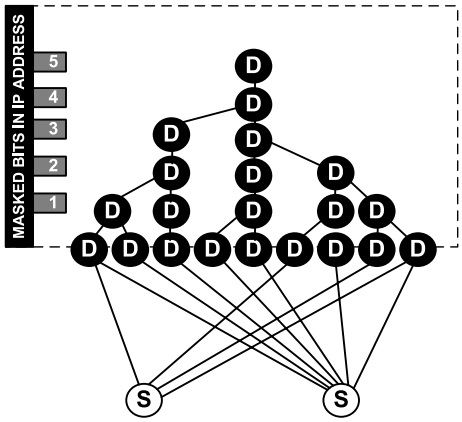

Fig. 1 Aggregation by destination.

by MAWI project in Japan [9]. A traffic analysis tool called *aguri* [10] is made available at the repository. In fact, *aguri* also uses IP aggregation to generate overall statistics of the traffic. More details in the repository can be found in [11]. Traffic was collected from a trans-Atlantic link connecting Japan and USA and a few other backbone links inside Japan.

2.1 Aggregation by Destination

Figure 1 displays the process of aggregation by destination IP. In this aggregation model, source IP addresses remain distinct while their destinations are aggregated and gradually merge in clusters with increasing depths of the mask. Destination aggregation can be useful to monitor network activity of a single source. Dynamics of changes in destinations contacted by the source can be used to infer user behavior.

Destination aggregation could also be used to merge multiple entries in a round-robin DNS. This kind of load balancing normally uses several servers with close IP addresses to reply to users that access a fixed URL. This application of destination, however, has small practical value in view of properties of network anomalies popular in research today.

Destination aggregation was displayed in this section for explanatory purposes only. The paper itself aggregates only sources, which is explained in detail in the next section.

2.2 Aggregation by Source

Figure 2 displays the process of aggregation by source. Opposite to the previous case, source aggregation is done for each destination separately. Sources would cluster with increasing depth of the mask, but all sources would always be attributed to a single destination.

Source aggregation is directly applicable to local network monitoring at ISP or corporate level as it automatically clusters the users into groups depending on their location in IP address space. Although neighbors in IP address space can still be separated by large distances, local network administrators would normally be aware of the design of their

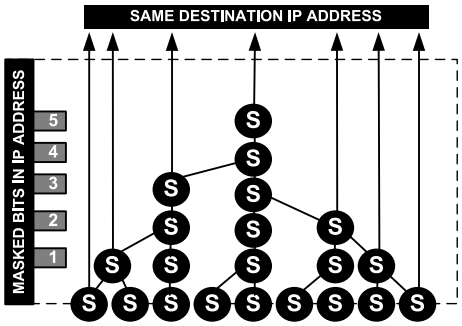

Fig. 2 Aggregation by source.

network and would interpret aggregation results correctly.

Major anomalies, albeit malicious attacks or signs of side effects of unusual use of the Internet, take extreme positions in terms of the number of destinations they target. For example, worms, Alpha Flows (large point-to-point data transfers), Flash Crowds (multipoint-to-point corrections to a popular location), DDOS, and SYN attacks target a single location, while network scanning would result in instantaneously increasing number of destinations. Both are easily captured by either source aggregation directly, or by monitoring the number of distinct destinations at any point of time. Specific cases of anomalies directly detectable by source aggregation are considered in the following sections.

Authors do not imply that only source aggregation is valuable for traffic analysis. In fact, for ample analysis both sources and destinations have to be studied in practice. However, this paper focuses mainly on the description of the detection framework rather than practical examples of its use. For simplicity reasons, only Flash Crowds and Alpha Flows are considered in this paper as examples. Both these anomalies focus on a single destination and therefore do not require analysis by destination aggregation.

2.3 Prefix-Safe Anonymization

The research of traffic aggregation would not be possible if not for the advanced anonymization of IP addresses in packet traces. To protect the privacy IP addresses have to be randomized before they access to them is granted to research community. This process is called anonymization. An algorithm allowing to randomize IP addresses in prefix-safe manner is described in [12]. This algorithm also exists in form of a tool that is applied by several large traffic repositories that grant free access to researchers.

3. Aggregation Process

Within the framework of this research, a software tool was developed to verify the findings on the proposed research. The software consists of two separate parts, the monitor and the browser, respectively. This section elaborates on the processes and data structures within the monitor.

3.1 Overall Process Flow

The process of collection is very similar to that of a NetFlow meter. In fact, the functionality of the monitor is also very similar to NetFlow as it stores traffic in flows. However, in the proposed process as certain regular time interval, IP aggregation is performed on the contents of stored flows after which the storage is flushed. Clearly, the shorter this interval the less flows are to be stored in the table simultaneously but at the same time the more difficult is would be to detect certain anomalies. However tempting, the scope of this paper does include the search for the optimal aggregation interval.

The aggregation process itself is displayed in Fig. 3. A separate list of flows is created for each distinct destination retrieved from all flows in the pool. Then, aggregation is performed for each of these lists by masking out a given number of bits at the end of source IP address. Variable-length prefix masks will generate different views of the same traffic in form of different sizes of traffic clusters and different distribution of traffic among them. Prefix length of masks does not have to be continuous and can jump several bits at a time within a given range normally much smaller than 32 bits.

3.2 Data Structures

In traffic analysis, underlying data collection system is very important. The proposed traffic analysis cannot be supported by either SNMP or NetFlow, which is why a separate data collection system was developed to facilitate analysis in this paper.

Figure 4 displays the overall structure of the system used for traffic analysis in this paper. Data collection is the integral part of the system and can work both with NetFlow and *tcpdump*. Naturally, each of these existing technologies requires a separate logic in aggregation process. In both cases, the module *Aggregator* is in charge of converting raw packet or flow data into aggregation data which is later used by aggregation-based analysis.

Although the performance of data collection subsystem is beyond the scope of this paper, it should be noted that the subsystem based on *tcpdump* is more efficient and much faster than the one based on *NetFlow*. In simple terms, the raw data and detail required by the proposed research is midway between the very simple *tcpdump* and the very elaborate *NetFlow* raw data.

The Storage module in Fig. 4 is implemented in form of a number of database tables. The database structure optimized for storage of aggregation results is displayed in Fig. 5. Results of aggregation performed at each regular interval are stored in 2 database tables, masks and flows. The main table is used to store general statistics stores only a single line per interval. The major performance optimization was performed on masks and flows tables, resulting in the structure in Fig. 5.

The masks table stores aggregation results down to masked addresses per destination per mask, thus resulting in masks * destinations * source clusters lines in the database table. The flows table is used to store actual flows, however, each line in this table retains relation to masked source address, mask depths and destination address and, therefore, can be easily accesses based on what destination, mask or masked source are under review at the moment. This structure proved to be optimal in performance for both storing and browsing.

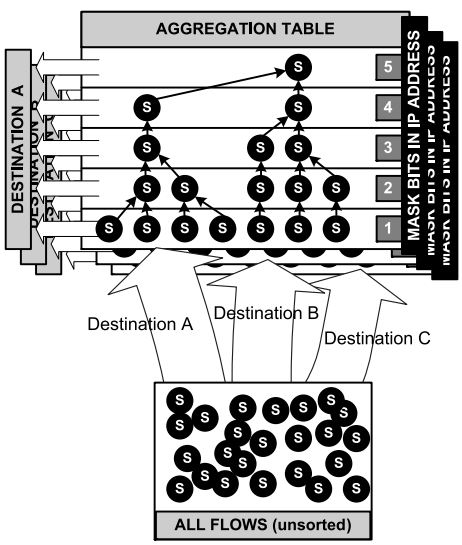

Fig. 3 Analysis of aggregated flows based on source IP prefix masks.

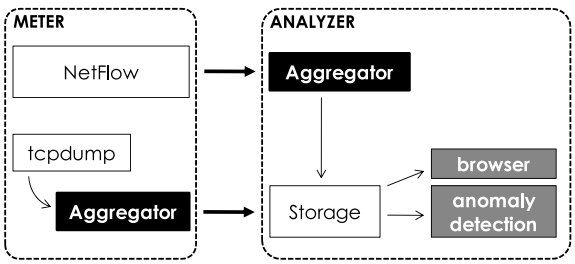

Fig. 4 Data collection subsystem that facilitates the aggregation-based analysis.

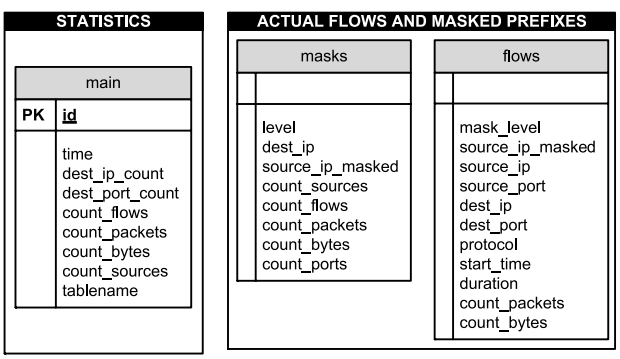

Fig. 5 Database structure used to store aggregation results.

4. Browsing Aggregation Results

This section elaborates on the second part of the developed software package, the browser. The browser is developed as a web application and can be accessed remotely over the network.

4.1 Hierarchical Design

Figure 6 displays the tree-like approached used in browsing. The displayed sequence is optimal in view of the database structure used to store aggregation results. The top list is the list of aggregations performed in time. Each timeslot unfolds into a list of distinct destinations. Each destination has a list of masks, each mask a list of source masks, each source mask a list of sources clustered under this mask, and finally each source has a number of flows originated from this source to the selected destination. Therefore, destinations exist at the top while flows form the bottom of the tree. Each of unfolded lists can be ordered by byte count, packet count, flow count, source count, and IP address in either ascending or descending order, if applicable. Each unfolded list in the web application contains a plot of samples of the selected metric in the context of the list. Comma-delimited text view of data is also offered for download and later use in spreadsheet or statistics applications.

4.2 Special Views

As the tree structure in Fig. 8 is limited to unfolding a single destination at a time, a number of tasks were developed to span time intervals, destinations, masks, and other nodes in the tree view. These special views are important for analyzing the behavior of traffic to a certain destination in time, or distributed of traffic mass between source clusters within a certain mask. The details of these special tasks are not included in the paper for space limitation, however, they are an important part of the monitoring task, as they provided sequences of samples that can be studied in multidimensional context discussed later in this paper. Tree view of the aggregation is a mere graphical interface to the database contents and can be used only on a one target at a time basis.

As a subset of special monitoring tasks we also developed a series of detection schemes along with a parsing engine that can apply these schemes to aggregation tables at runtime. These schemes are based on the multidimensional features of aggregated data discussed later in this paper, where each detection rule is applied to several metrics at once. At the present moment, all the rules are binary, i.e. are based on on-off states of each metric. Practically, for an example, this may translate to deciding whether there is only one or many sources under a certain source mask, or whether a certain mask depth resulted in major clustering of sources or not. The experience from using these rules proved that even binary conditions are very effective to detecting several anomalous conditions with relative ease.

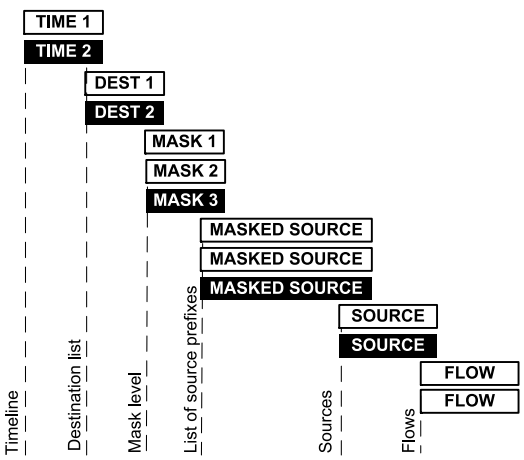

Fig. 6 Hierarchical design for browsing aggregation results.

5. Multidimensional Analysis Space

This section discusses multidimensional nature of metrics made accessible through aggregation. There are at least two 3-dimensional spaces that are applicable to detection of various anomalous conditions in traffic. The statistical significance of sequences of these metrics is considered to prove that the relations among metrics are not always linear.

5.1 Analysis Spaces

Figure 7 contains the view of at least two 3-dimensional metric spaces that can be used for monitoring and detection of anomalies. Capital letters are used to simplify addressing these spaces in discussion of relations among them.

The ABC space obtained directly from time sequence of aggregation tables can be applied to basic monitoring and detection of Flash Crowds (large number of connections simultaneous connections from multiple sources to the same destination). This is very similar to traditional traffic monitoring that normally focuses solely on utilization. ABC space allows viewing utilization distribution among destinations and study relation between sources, destinations and aggregated traffic between them.

The DEF space is applicable to a certain selected destination and represents dynamics of source clustering as a function of mask depth and total number of clusters. This space makes the task of detecting Alpha Flows very easy as

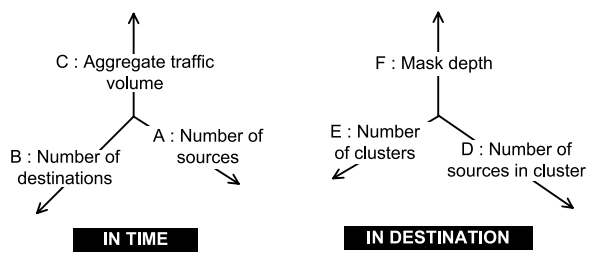

Fig. 7 Dimensions accessible directly from aggregation results.

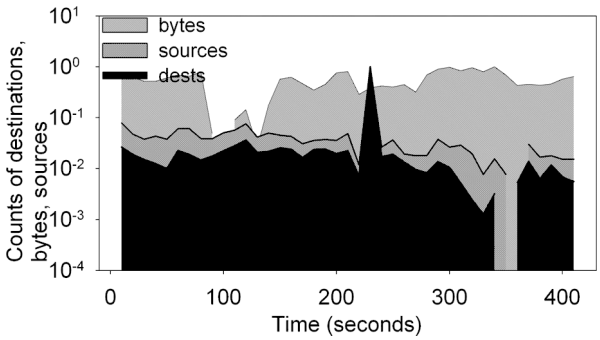

Fig. 8 Normalized destination and source counts, and total traffic volume in time.

Cell Contents:
Correlation Coefficient
P Value
Number of Samples

	Bytes	Sources
Destinations	-0.114 0.478 41	0.997 2.44E-45 41
Bytes		-0.116 0.469 41

Fig. 9 Results of correlation test for destination, source, and byte counts.

Cell Contents:
Correlation Coefficient
P Value
Number of Samples

	Max Sources	Mean Sources	Source Masks
Mask	0.936 0.0638 4	0.818 1.82E-01 4	-0.971 0.0295 4
Max Sources in Mask		0.928 0.0717 4	-0.988 0.0115 4
Mean Sources in Mask			-0.926 7.38E-02 4

Fig. 10 Source cluster dynamics in case of a popular destination.

in this case the traffic volume would be roughly the same at all mask depths and in any cluster size. In fact, increments of total traffic volumes from a single cluster at increasing mask depths is indicative of localized intensity of access to a certain location and can be used by administrators to construct traffic intensity maps within their domains.

5.2 Statistical Significance of Aggregation Metrics

This section studies the relations among metrics in the two multidimensional spaces mentioned above by applying the test of statistical significance often used in statistics to identify correlation in independent series. In our case, both the presence and the absence of correlation is acceptable, but the absence is preferred as this would mean that difference in metric dynamics is statistically significant, i.e. the multidimensional space offers more features. As was mentioned earlier, the packet traces from MAWI project in Japan [9] were used for this study.

Figure 11 displays sequences of ABC space metrics in time, in particular the raw destination count and traffic in the upper part and normalized plots of all three metrics in the lower part of the figure. It is visually detectable that traffic volume (byte count) is not directly related to dynamics in number of sources and destinations, while the latter exhibit correlated trends. This, however is more of a positive argument on the part of PCA-based data mining research. This paper, however, shifts attention to metrics generated directly by using IP aggregation.

Figure 9 contains the results of correlation test performed using Pearson Product Moment algorithm, which is good for noisy numerical data. Destination and source counts have 0.997 correlation coefficient with negligibly low probability of Error Type I, i.e. the probability that the correlation does not exist. Correlation between destination and source counts and aggregated traffic is negative (correlation of negative trends) and very low, with very high probability of no relation existing at all.

Although in Fig. 9 destination and source counts are perfectly correlated, this is not always the case. Unfortunately, we were unable to find a portion of traffic with a Flash Crowd, i.e. access to the same place from multiple sources at roughly the same time. This traffic artifact is quite rare and is often connected to major sport events or release of a new version of popular software, which makes it relatively hard to find. However, it is legitimate to assume that in the event of a Flash Crowd, the correlation of destination and source count would drop temporarily, which can be detected directly from ABC space. To facilitate this both destination and source counts were included in the space.

Results of correlation tests in DEF space can be found in Fig. 10 and consist of correlation results between aggregated statistics as a function of mask depth. The results show that max and mean of cluster size, i.e. number of sources in clusters are very well correlated with the depth of the mask. The dynamics of the number of cluster show the high negative correlations, which also can be expected of a popular destination. The higher the popularity of a particular destination, the higher should be the correlation of the above statistics to the depth of mask, as this is indicative of the smoothness of user distribution worldwide.

6. Practical Applications

IP aggregation can be applied in a wide range of tasks related to traffic engineering, network performance monitoring and troubleshooting, and detection of anomalous conditions in traffic. This paper cannot accommodate all possible case studies of practical applications of the proposed method. Instead, this section contains some examples of detection of two major anomalies: Flash Crowd and Alpha Flow. Both anomalies are similar in that their appearance causes increase in traffic volume, while they show com-

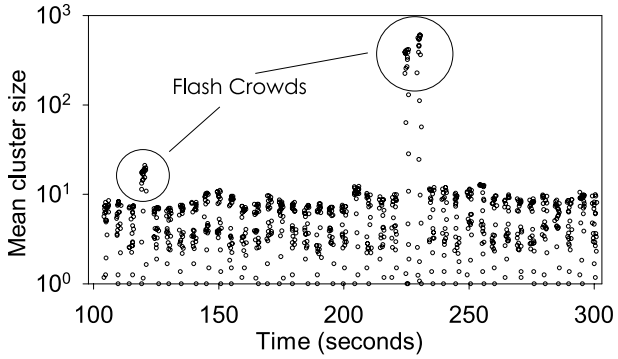

Fig. 11 Analysis based on mean cluster size of aggregated sources to a single destination with variable-length prefix.

pletely different patterns in IP aggregation results. Therefore, the detection of these anomalies using IP aggregation is a good way to prove the advantages of the proposed method versus conventional detection methods.

6.1 Flash Crowd Detection

To study properties of Flash Crowd anomaly obtained through source aggregation, the analysis was simplified by locating and isolating popular destinations from other traffic. The traffic from WIDE project contains several highly popular destinations at any point of time for which the number of sources range from hundreds to thousands of sources, or, in aggregation terms, from hundreds to thousands of clusters at lower bits.

For descriptive purposes, Fig. 11 displays the ease with which Flash Crowd anomalies can be detected using basic IP aggregation metrics. In Fig. 11 various prefix lengths were used to aggregate traffic sources of a single destination into clusters and the mean size (source count) of these clusters was plotted for each prefix. Prefix masks in the range of 1 through 18 were used to produce each vertical line which stands for aggregation results calculated each 5 seconds. To avoid overlapping, each sample was shifted randomly either right or left from the main axis of each line.

It is obvious from Fig. 11 that with decreasing prefix length Flash Crowds stand out more vividly from the normal traffic. In normal traffic, access to any particular destination is distributed fairly smoothly over the entire IP space of the access to that particular destination. In the event of a Flash Crowd, some area of access is temporarily becoming more prominent over all the others, thus, resulting in much greater gaps in cluster source counts. This pattern can be successfully exploited using a fixed prefix length, but the use of a variable length prefix decreases the possibility of a false positive. In the proposed framework, the use of variable prefix lengths is native and can be easily applied to any destination.

6.2 Detection of Alpha Flows

The detection method used for Alpha Flows is fairly straightforward. We can use the same distribution plots as in the case of the previous Flash Crowd anomaly. Distribution of byte count alone is sufficient to separate Alpha Flow from the rest of the traffic. In fact, if by definition Alpha Flow is a point-to-point flow of an exceptional traffic volume, it would be found at the very top of the list of clusters ordered by byte count. As its traffic is coming from only one source, the value of traffic does not change much with decreasing prefix length which is why the traffic volume of this particular cluster remains relatively unchanged at any prefix.

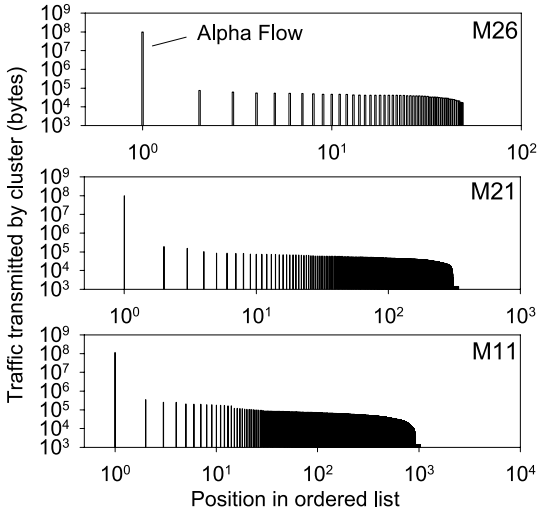

Fig. 12 Distribution of cluster traffic for each prefix length.

The result of detection is shown in Fig. 12, where the cluster with the anomalous Alpha Flow can be visually confirmed to be several orders of magnitude larger than its neighbors. Prefix lengths in the plot are denoted in Mxx form, where M stands for the word mask and xx is its length in bits. The mask is the opposite of the prefix as it stands for the number of bits at the end of IP address that are ignored. Visualizing aggregation results by the depth (length) of the mask is more straightforward than the traditional notation by the prefix length.

Optionally, a 2-dimensional plot of cluster size versus cluster traffic volume could be used as an alternative method of visualization. In this case, each new prefix length would shift cluster sizes of all traffic but the Alpha Flow, which would remain in relatively the same spot on the plot at any prefix length.

7. Conclusions

This paper proposed a novel approach to online traffic analysis that applies source address aggregation to traffic flows. Besides the aggregation process itself, the major difference from conventional flow-based monitoring is the use of a short time interval at the end of which the aggregation is performed and the list of flows is purged releasing the memory for the traffic in the next interval. This saves considerable amount of resources at runtime and allows for online

operation - a feature much valued by any anomaly detection method or a tool.

The aggregation process itself offers direct access to a number of new metrics that can diversify both network monitoring and anomaly detection tasks by introducing multidimensional features into traffic metrics. Two 3-dimensional spaces were proposed and verified by using the test of statistical significance, which proved that there are many variations with extremely high and low correlation coefficients. Both cases are good for analysis since they both signify the presence multiple patterns in data. Several examples of such detection were discussed in the paper.

Due to space limitations only two particular anomalies were considered in the paper, Flash Crowd and Alpha Flow, since the presence of both was confirmed by manual perusal of the traffic traces. The ease of detection was amply demonstrated and originated from the rich set of basic traffic metrics created automatically by applying IP aggregation to a traffic trace.

Although it could be argued that there are many other much simpler methods of detecting both Flash Crowds and Alpha Flows, it should be noted that the main purpose of this paper was not to detect Flash Crowds and Alpha Flows but to created an easily customizable framework for detection of a wide variety of traffic anomalies. It should also be stressed that network attacks covered by very rigorous research referred to as intrusion detection are only a subset of a broader area of network anomalies, which, in fact, can be perfectly benign and free of premeditation.

Also, the scope of this paper did not include the analysis of detection sensitivity which would normally collect all false positives and false negatives and calculate rates of false alarms, success ratio, etc. This kind of research would seek optimal parameters of detection schemes used for each anomaly and compare them using ROC curves - a well established verification tool for detection methods. This paper proposes a detection framework in very general terms, thus, allowing its use for detection of a wide variety of network anomalies. This research is, however, planned as the logical continuation of work presented in this paper.

References

[1] N. Brownlee, "Traffic flow measurement: Meter MIB," RFC2720, Oct. 1999.
[2] E.B. Claise, "Cisco systems NetFlow services export version 9," RFC 3954, Oct. 2004.
[3] J. Jung, B. Krishnamurthy, and M. Rabinovich, "Flash crowds and denial of service attacks: Characterization and implications for cdns and web sites," International World Wide Web Conference (WWW), pp.252–262, IEEE, May 2002.
[4] Y. Zhang, S. Singh, S. Sen, N.G. Duffield, and C. Lund, "Online identification of hierarchical heavy hitters: Algorithms, algorithms, evaluation, and applications," Internet Measurement Conference, pp.101–114, Oct. 2004.
[5] A. Lakhina, M. Crovella, and C. Diot, "Characterization of network-wide anomalies in traffic flows," 4th ACM SIGCOMM Conference on Internet Measurement, pp.201–206, New York, NY, USA, Oct. 2004.
[6] A. Lakhina, M. Crovella, and C. Diot, "Diagnosing network-wide traffic anomalies," SIGCOMM Computer Communications Review, vol.34, pp.219–230, Oct. 2004.
[7] A. Lakhina, K. Papagiannaki, M. Crovella, C. Diot, E.D. Kolaczyk, and N. Taft, "Structural analysis of network traffic flows," Performance Evaluation Review, vol.32, no.1, pp.61–72, ACM SIGMETRICS, June 2004.
[8] "Snort." Available at: http://www.snort.org/
[9] "MAWI working group traffic archive." Available at: http://tracer.csl.sony.co.jp/mawi/
[10] K. Cho, R. Kaizaki, and A. Kato, "Aguri: An aggregation-based traffic profiler," Proc. Second International Workshop on Quality of Future Internet Services (QofIS), pp.222–242, Sept. 2001.
[11] K. Cho, K. Mitsuya, and A. Kato, "Traffic data repository at the WIDE project," USENIX 2000 FREENIX Track, pp.263–270, June 2000.
[12] D. Koukis, S. Antonatos, D. Antoniades, P. Trimintzios, and E. Markatos, "A generic anonymization framework for network traffic," IEEE International Conference on Communications (ICC), pp.2302–2309, June 2006.
[13] C. Estan, S. Savage, and G. Varchese, "Automatically inferring patterns of resource consumption in network traffic," ACM SIGCOMM, pp.137–148, Aug. 2003.

Marat Zhanikeev received a B.S. degree in electrical and electronics engineering from Tashkent State Technical University, Tashkent, Uzbekistan, and a M.S. and a Doctor of Science in Global Information and Telecommunication Studies from Waseda University, Tokyo in 1997, 2003, and 2007, respectively. From 1997 to 2001, he worked as an engineer at Daewoo Telecom Tashkent on a number of telecommunications modernization projects for a governmental development program. His current research interests include network measurement, monitoring, and management. He is presently an assistant professor at the School of International Liberal Studies, Waseda University.

Yoshiaki Tanaka received B.E., M.E., and D.E. degrees in electrical engineering from the University of Tokyo, Tokyo in 1974, 1976, and 1979, respectively. In 1979 he joined the Department of Electrical Engineering, the University of Tokyo, where he has been teaching and researching in the fields of telecommunication networks, switching systems, and network security. He is presently a professor at the Global Information and Telecommunication Institute, Waseda University. He received the IEICE Achievements Award in 1980, the Okawa Publication Prize in 1994, and the IEICE Best Paper Award in 2005.