Cardinality-Based Abnormal Traffic Classification Using Packet Sampling

Kohei ITO[1], Akira SATO[1], Shuji SANNOMIYA[1,2], Kenichi YOSHIDA[1], and Yasushi SHINJO[1]
[1]University of Tsukuba, Ibaraki, Japan
[2]DDSNA, Inc., 1-1-1 Tennodai, Tsukuba, Ibaraki, Japan

Abstract—*Network administrators strive to detect abnormal traffic such as distributed denial of service (DDoS) attacks and Internet worms, as soon as it occurs. Cardinality analysis is a method to detect such abnormal traffic. Conventional cardinality analysis has a scalability problem as Internet traffic increases beyond the class of 100 Gbps. To address this problem, we propose in this paper cardinality analysis using sampled packets to reduce the amount of packets to be analyzed up to 1/1000. We compared the number of cardinality count reports using sampled Internet traffic and unsampled Internet traffic. The results show that several kinds of abnormal traffic disturbing important communication can be discovered using the sampled packets.*

Keywords: cardinality, packet sampling, traffic classification

1. Introduction

Abnormal traffic, such as distributed denial of service (DDoS) attacks and Internet worms, causes various problems in network systems, including large-scale Internet of Things (IoT) networks. For example, a large traffic flow amount due to a DDoS attack disturbs important network communication. Therefore, network administrators must detect such abnormal traffic. Shomura et al. [1] proposed cardinality analysis to detect abnormal traffic by using a counting routine to continue tracking the header field combinations that are frequently found in the packet streams of abnormal traffic and the variations of those combinations. This counting is called cardinality counting because of its similarity to mathematical cardinality, which denotes the number of elements in a given set. Moreover, Mori et al. [2] developed vicar general-purpose software, which implements the cardinality counting.

Meanwhile, Internet traffic has continued to explosively grow and its data rates have become 100 Gbps or higher in backbone networks. According to Mori et al., cardinality analysis by vicar using an Intel Core i7-3930k can handle up to an 1 Gbps line. To realize real-time cardinality analysis of 100 Gbps traffic, Sannomiya et al. [3] proposed a self-timed pipeline circuit implementation of cardinality counting by fully pipelining the cardinality analysis processes. Although the circuit implementation can achieve high throughput, its hardware resources for pipelining increase approximately in proportion to the required throughput. It is thus expected to become difficult to provide the throughput in the order of terabytes per second (Tbps) or greater for future network systems.

In this paper, a technique of cardinality analysis is proposed to realize the handing of a large amount of packets without using a massive amount of hardware or computing resources. The proposed technique introduces the sampling of target traffic in order to reduce the volume of input traffic. For example, if the volume is reduced to 1/100 by sampling, we can analyze the 100 Gbps traffic using the computing resources for 1 Gbps. Meanwhile, 10 Tbps can be analyzed by using the hardware resources for 100 Gbps.

Although use of sampling appears promising in cardinality analysis, we must verify whether using sampled packets can reproduce the same result as using all packets. To conduct this verification, the results of cardinality analysis with sampled packets are compared with those of cardinality analysis with all packets. Based on the comparison results, it is shown that various kinds of abnormal traffic can be detected by using the sampled packets.

2. Related Work

2.1 Cardinality Analysis

Shomura et al. [1] developed a simple frequent-itemset-mining method that uses only small amounts of memory and processing power to count cardinality. This method focuses on the itemset of every packet and simultaneously monitors its frequency and variation. The itemset refers to the combination of five packet's header fields that are the source IP address (SIP) , destination IP address (DIP) , source port number (SPT) , destination port number (DPT) , and protocol number (PRT).

The algorithm of the developed method is named the CPM algorithm and it is shown in Fig. 1. The itemsets function is recursively called to count the frequencies of itemsets. Fig. 2 shows an example of the function call process for itemsets consisting of SIP, DIP, and DPT. In Fig. 2, the numbers appended to the arrow show the recursion sequence. The mark "()" denotes the variable "items", and the mark "{}" represents the variable "rests."

To check whether each itemset is new or already found, the Hash2 memory management strategy is used. Hash2

ISBN: 1-60132-487-1, CSREA Press ©

```
Algorithm CPM
Variable
    Cache[]: fixed-size table
begin
    Create empty cache;
    while (input Transaction)
        for each item in Transaction
            Itemsets(item, rest of items in Transaction);
end


Function Itemsets(items, rests)
Variable
    items[]: items in the current itemsets
    rests[]: other items in the transaction
begin
    i = index of items in cache; (calculated by hash2)
    if (i is new index)
        increment cache_diff[index of original items] by 1;
    increment cache_cnt[i] by 1;
    for each item in rests
        Itemsets(items+item, rest of items in rests);
    if (cache_cnt[i] ≥ threshold)
        report statistics;
        cache_cnt[i] = 0;
        cache_diff[i] = 0;
end
```

Fig. 1: CPM algorithm.

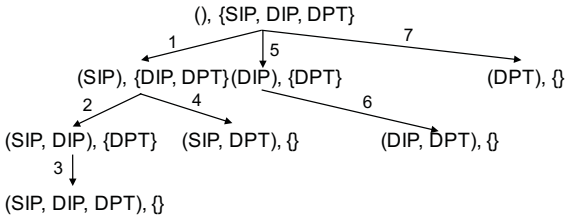

Fig. 2: Recursive call process

is shown in Fig. 3. It first calculates "n" hash values of the input itemset and then generates "n" indexes from the calculated values. If the itemset is new, the input itemset is not equal to any of the cache memory contents referenced by the indexes. In this case, Hash2 selects the index that refers to the entry with the least frequency from "n" indexes. Then, the selected itemset is replaced by the new itemset. According to Shomura et al., "n" is set to four based on the previous experimental evaluation result.

The itemset frequency and variation are stored in a fixed-size cache table. Table 1 presents an example of the cache table. The frequency and variation are represented by cache_cnt and cache_diff, respectively. Generally, the number of columns depends on the number of items to be considered. However, only five columns are shown in the table because only two items, SIP and SPT, are considered for providing a simplified description. Each entry of the

```
Function Hash2
Input
    Item: Data to be stored in Cache
Variable
    Hash[]: Table of Hash Values
    Idx[]: Table of Cache Index
begin
    Calculate "n" hash values from Item
                and store them into Hash[]
    Idx[] = Hash[] % Cache Size
    if (one of entry refereed by Idx[] stores Item)
    then    return Idx that refers the entry
    else    Select Idx that refers least frequent entry
            cache_cnt[Idx] = 0
            return Idx
end
```

Fig. 3: Hash function for quasi-association.

cache table corresponds to a distinct value of the itemset. When the itemset is already stored in the cache memory, the value of cache_cnt is incremented. The value of cache_diff is cleared to zero when the corresponding itemset is newly found. Moreover, it is incremented when the supersets of the corresponding itemset are newly found. For example, both cache_diff of the SPT column in the entry of the itemset (SIP) and cache_diff of the SIP column in the entry of the itemset (SPT) are incremented when a new itemset (SIP, SPT) is found. This is shown in Fig. 4. The subsets of the newly found itemset are called "original items." They are generated by substracting one item from the newly found itemset. The count for cache_diff is maintained by the single-underlined steps in the Fig. 1. Some examples are shown in Table 1. In one example, the first entry in the table (ID = 1) shows that there are 16 packets whose SIP is "192.0.2.153" and whose SPT is "80". The second entry in the table (ID = 2) shows that there are 152 packets whose SIP is "198.51.100.3." In addition, the number of packets with different SIPs is at least "8," i.e., there are at least eight packets whose SIPs are "198.51.100.3," as designated with distinct SPT's.

The values of the cache_cnt and cache_diff are checked in each iteration. Furthermore, the statistics of the corresponding itemset are reported when the values of cache_cnt exceed the threshold values that discriminate between abnormal and normal traffic. Then, these value are set to zero. This operation is realized by the double-underlined steps in the Fig. 1 . Using this algorithm, in a fixed size memory, only frequently found itemsets are enumerated along with the number of variations in the non-frequent parts of the itemsets. Shomura et al. analyzed the results of several measurements separately from TCP and UDP. In this paper, we only analyze results of TCP.

Table 1: Structure of cache table.

ID	cache _cnt	1st item (e.g. SIP)		2nd item (e.g. SPT)		n-th item
		cache_diff	value	cache_diff	value	
1	16	-	192.0.2.153	-	80	...
2	152	-	198.51.100.3	8	-	...
3	2	2	-	-	443	...
4	40	-	203.0.113.55	-	110	...
5	20	-	192.0.2.153	2	-	...
6	20	2	-	-	80	...

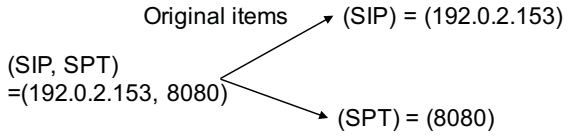

Fig. 4: Example of original items.

2.2 Packet Sampling Techniques

Packet sampling is a technique for reducing the number of packets to be processed. It can additionally be used in collecting packets flowing through the network for analysis or monitoring. IETF RFC 5475 [4] defines the packet sampling scheme. In that RFC, packet sampling methods are classified into two systems: 1) systematic sampling which obtains packets at regular intervals based on the number of packets or time, and 2) random sampling, which determines intervals obtained by random numbers.

2.3 Monitoring and Analyzing Traffic Using Packet Sampling

Several methods using packet sampling for traffic monitoring and analysis have been proposed. Davide et al. [5] implemented four of the packet sampling algorithms defined in [4]. They investigated the impact of packet sampling on various network monitoring tasks (mainly characterization and classification). The four implemented algorithms are outlined as follows.

- Systematic sampling
  When sampling rate k is set, the first packet of every k packets is obtained.
- Random sampling
  When sampling rate k is set, packets are randomly obtained with a probability of 1 / k.
- Stratified sampling
  When sampling rate k is set, one packet is randomly obtained from every k packets.
- Systematic SYN sampling
  In addition to the packet obtained by the systematic sampling, a packet whose TCP SYN flag value is one is obtained.

Davide et al. measured the effect of packet sampling on the characteristics of traffic by using two metrics: the HD distance and Fleiss Chi-Square. At low sampling rates, most features deteriorate regardless of the sampling algorithm. However, features that depend on analysis of specific protocols (such as TCP options) are less likely to degrade.

Valentín et al. [6] evaluated the accuracy of traffic classification based on machine learning using packets that are randomly sampled by NetFlow. When using sampling packets for traffic classification by C4.5, the accuracy was greatly reduced. However, by using sampling packets for the training process, Valentín et al. achieved 85% classification accuracy with 1/100 sampling.

Salama et al. [7] developed an adaptive sampling technique based on regression and a fuzzy inference system to assess the large network performance. In the developed method, traffic fluctuation is determined by a regression model using traffic parameters. The sampling interval is determined by fuzzy logic according to traffic fluctuation. This method increases the number of packets to be obtained when the traffic fluctuation is high, while it obtains fewer packets during the periods of low traffic fluctuation.

Mahmood et al. [8] proposed a scalable two-stage adaptive sampling method. This sampling method obtains traffic patterns whose sizes are small and preferentially rare by focusing on the tail-heavy distribution of the traffic pattern sizes. In the proposed method, a series of flow records extracted from the network stream is used as input. By lowering the priority of records with frequently occurring patterns in two stages of random sampling, records of small and rare patterns can be acquired even if the sampling rate is decreased.

Giotis et al. [9] aimed to detect anomalies in software defined networks (SDNs). Giotis et al. compared the results of an entropy-based analysis method with all packets collected by both Openflow, which is a technique used in SDN, and sFlow, which is a technique for performing random packet sampling in a switch. For communication at 100 Mbps and 500 Mbps, Giotis et al. conducted analysis with sampling at 1/64 and 1/256, respectively. Although the false-positive rate slightly deteriorated compared with the case of Openflow, it showed a similar tendency of the entropy value. Additionally, it demonstrated that DDoS attacks and scans included in the packet are detected at 100%.

These traffic monitoring and analyzing using packet sampling imply that we can obtain sufficient accuracy, even if the used packets are reduced. In this paper, we propose reduction of the packets used for cardinality analysis by sampling.

3. Cardinality-Based Abnormal Traffic Classification Using Packet Sampling

3.1 Analysis Target Traffic

In this paper, we verify whether the same analysis result can be obtained in the case of using all packets by cardinality analysis and using sampling packets. As the analysis target traffic, we use a part of the traffic trace provided by MAWI

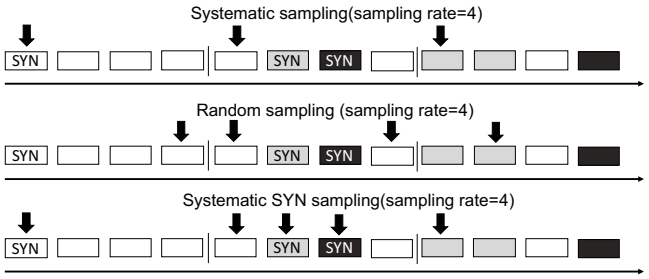

Fig. 5: Implemented sampling algorithms

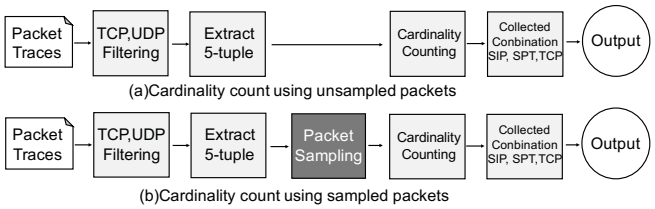

Fig. 6: Cardinality count process

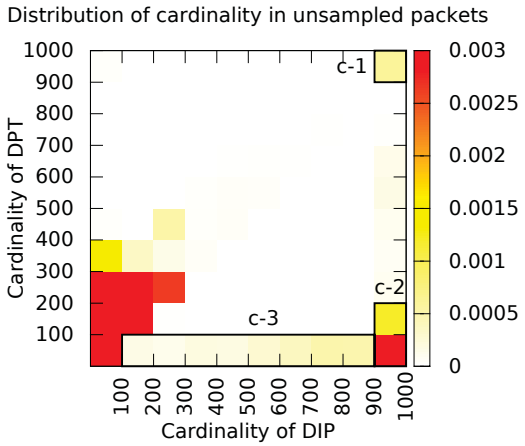

Fig. 7: Relative frequency of cardinality count using all unsampled packets

Working Group of the WIDE Project [10]. The used traffic trace was collected from 23:45 on April 12, 2017 to 23:45 on April 13, 2018 at an upstream ISP of MAWI Working Group. It contained IPv4 and IPv6 packets.

3.2 Implementation of Cardinality Analysis and Packet Sampling

In this paper, we sampled the packets from the traffic trace and input the sampled packets to the cardinality analysis algorithm. To check the difference of the sampling methods, we implemented three kinds of sampling algorithms proposed by Davide et al. [5]: systematic sampling, random sampling, and systematic SYN sampling. The overview of the implemented sampling algorithms is shown Fig. 5. Fig. 5 shows the operation of each sampling algorithm with a sampling rate set to four. The downward arrow points to the obtained packet. The packet with the same color is the packet included in the same TCP session. A packet denoted as "SYN" indicates a packet in which the value of the TCP SYN flag is set to one.

With respect to cardinality counting, we improved the program of Mori et al. [2] by changing the data structure used in the original program. The following improvements were made: 20% memory size reduction, and omission of the process of converting the input string values to hash values.

4. Evaluation

4.1 Evaluation Overview

As described in this paper, we compared the output of the cardinality count using unsampled packets and sampled packets. As mentioned earlier, Shomura et al. analyzed the results of several measurements separately from TCP and UDP. According to statistics of packets provided by MAWI Working Group [10], the number of packets of IPv4 TCP was the largest. We decided to first analyze IPv4 TCP packets with a large number of packets. For this paper, we only analyzed TCP output. We set the threshold as 1000, similar to Shomura et al. [1], to count the cardinality.

First, we extracted IPv4 TCP and UDP packets from MAWI Working Group. The 5-tuple itemset was extracted from the header information. The cardinality count procedure with all packets is shown in Fig. 6 (a), while that of the cardinality count with samplilng is shown in Fig. 6 (b). The difference exists only with respect to the sampling.

In this paper, we focus on the same cardinalities discussed in the paper of Shomura et al. [1]. The cardinalities of focus are the cache_diff's of DIP and DPT reported with the same SIP and SPT values. In addition, the PRT is TCP. Fig. 7 shows cardinalities of focus. The vertical axis denotes the cardinality of DPT. The horizontal axis denotes the cardinality of DIP. To clarify the number of reports with the same cardinalities, we divided the result graph into 10 × 10 compartments. In this figure, the frequency of the reports with the same cardinalities in each compartment is expressed in color, thereby denoting the relative value of the frequency.

The reports with the same SIP and SPT values and PRT as TCP seem to indicate that there may be the responses from the server. In the case of normal communication, the server communicates with a specific client, and the cardinality of DIP and DPT is considered to be approximately one. When the server frequently establishes short sessions with the same client, the cardinality of DPT becomes high. When the server communicates with multiple clients, the cardinalities of both DIP and DPT increase. Therefore, the reports that are considered to be concerned with such normal communication

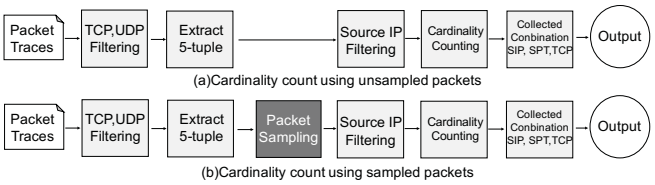

Fig. 8: Cardinality count process using each SIP

are concentrated on the origin of the vertical axis.

The purpose of cardinality counting is to detect abnormal communication. Therefore, those reports concerned with normal communication are excluded from the analysis by limiting the maximum value of the relative frequency to 0.003.

In Fig. 7, there are parts with many reports, although the values of the cardinality of DIP and DPT deviate from normal. It is considered that abnormal communication exists in this part. Among the remaining reports, we focus on three SIP values whose nodes are considered to be abnormal by using the three conditions below.

- **c-1** The cardinalities of both DIP and DPT are larger than 900.
- **c-2** The cardinality of DIP is larger than 900, and the cardinality of DPT is smaller than 200.
- **c-3** The cardinality of DIP is larger than 100 and smaller than 900. The cardinality of DPT is smaller than 100.

In addition, we set the sampling rate to 1/10, 1/100 and 1/1000. Fig. 8 (a) and Fig. 8 (b) show the procedure to conduct the cardinality analysis for all TCP and UDP packets and their sampling packets, respectively.

4.2 Results and Considerations

Table 2 shows the number of reports for each IP address. The three IP addresses in the reports satisfying condition c-1 are labeled as IP1, IP2 and IP3, respectively. All the TCP packets sent by IP1 have the ACK flag set, and 443 is used as the SIP. Most of the TCP packets sent by IP2 have the ACK flag set, and 80 is used as SIP. The ACK packets are sent from both the IP1 and IP2 to many destinations and their ports. The nodes of IP1 and IP2 apparently confirm the filtering status of the ports of the firewall by checking whether RST packets are returned.

All the TCP packets sent from IP3 have the SYN and ACK flags set, and 80 is used as SIP. The nodes with IP3 are considered to have received SYN packets from many nodes and returned packets with SYN and ACK flags.

As shown in column c-1 of the table. 2, in the cases of both random sampling and systematic sampling, the reports whose SIPs are IP1 and IP2 cannot be obtained if the sampling rate is 1000 since the total number of packets is insufficient. On the other hand, the number of reports decreases in proportion to the sampling rate. In this paper, we discuss about packet sampling when the total number of packets increases. If the traffic of such communication is also increases in proportion to the increase in the total amount of network traffic, it is considered that the reports of cardinality count can be obtained using sampled packets by random sampling and systematic sampling. If such traffic does not increase, it is considered to be negligible with respect to traffic capacity.

Since the SYN flag is set in the packets from IP3, the reports of cardinality count can be obtained by systematic SYN sampling even when the sampling rate is 1000. Systematic SYN sampling is considered effective for scans including many SYNs. On the other hand, in IP1 and IP2, the same effect as in systematic sampling is obtained.

Table 2 additionally shows the results satisfying condition c-2. The three IP addresses of those reports are labeled IP4, IP5 and IP6, respectively. In most packets of these IP addresses, the SYN flag is set. Apparently, these three IP addresses scan a port for several ranges in a long term for a large number of IP addresses. For example, the cardinality of DPT in the reports whose SIP is IP6 ranges from 1 to 100. Based on this fact, it is assumed that the scanning range for which the ports with IP6 are intended is from 1 to 100. Since the nodes with these three IP addresses send a large number of packets, the reports can be obtained, even if the sampling rate is 1000. However, the number of reports whose SIPs are IP5 and IP6 decreases in proportion to the sampling rate for the condition c-1. Therefore, if the number of packets decreases, such scanning cannot be detected. Conversely, it is considered that more reports can be obtained when the number of packets further increases. As for IP4, the number of reports becomes extremely small if the sampling rate is lower than 100, and the cardinality of DIP is slightly less than 200 when using unsampled packets. Since the cardinality of DPT increases slightly as the sampling rate decreases, the cardinality of DPT exceeds 200, and the number of reports within the range decreases. Especially for IP4, it seems difficult to obtain the reports from the cardinality count with the sampling packets even if the number of packets increases. Since most packets from these IP addresses are SYN packets, almost the same number of reports is obtained by using systematic SYN sampling compared with the case of all packets.

With regard to reports satisfying the condition c-3, the three IP addresses in those reports are labeled IP7, IP8, and IP9, respectively. Moreover, the cardinality of DPT in the reports is very low at one or two. The cardinality of DIP appears in a wide range. Column c-3 of Table 2 shows the number of the reports of the cardinality count for IP7, IP8, and IP9.

In most packets sent from IP7, IP8 and IP9, the SYN flag is set. All packets sent from the three IP addresses frequently use 23 or 2323 as DPT. The cardinality of DIP is distributed

Table 2: Number of outputs of cardinality count using each IP

	c-1			c-2			c-3		
	IP1	IP2	IP3	IP4	IP5	IP6	IP7	IP8	IP9
Unsampled Packet	486	481	265	5133	2855	2673	1122	767	350
Random Sampling Rate 10	48	48	26	513	285	266	118	76	47
Random Sampling Rate 100	4	4	2	12	28	26	11	7	2
Random Sampling Rate 1000	0	0	0	0	2	2	1	0	0
Systematic Sampling Rate 10	48	48	26	512	285	266	118	76	47
Systematic Sampling Rate 100	4	4	2	7	28	26	11	7	2
Systematic Sampling Rate 1000	0	0	0	0	2	2	1	0	0
Systematic SYN Sampling Rate 10	48	48	265	5093	2832	2651	1121	767	350
Systematic SYN Sampling Rate 100	4	4	265	5089	2830	2649	1121	767	350
Systematic SYN Sampling Rate 1000	0	0	265	5088	2829	2649	1121	767	350

in a wide range from 100 to 900. The nodes with these three IP addresses apparently exchanged data several times if there was a reply after sending the SYN packet to port 23 and 2323. Mirai [11] is known as a botnet which creates this type of communication. Mirai scans TCP's 23 and 2323 ports, and it attempts to connect by Telnet if the port is open. The cardinality of DIP does not become as extremely high as it does for the three IP addresses because the Mirai succeeds to scan several times and attempts to communicate with the same node. Since the nodes with IP8 or IP9 did not send as so many packets, the reports of the cardinality count cannot be obtained if the sampling rate is 1000. However, as with the case of the three IP addresses satisfying the condition c-1, if the number of packets increases so much that it disturbs important communication, it is considered that output can be obtained even if the sampling rate is decreased. Since most packets from these IP addresses are SYN packets, almost the same number of reports are obtained by using the systematic SYN sampling compared with the case of using all packets.

Furthermore, to verify the influence of sampling on the TCP report in all SIP's, the cardinality count was conducted with the packets sampled by three kinds of sampling algorithms without filtering by SIP in the procedure of Fig.6 (b). From the results of the cardinality count, we extracted SIP and SPT with a specific combination. The PRT is TCP. Fig. 9, Fig. 10 and 11 show the results of the relative frequency of the reports.

Fig. 9 shows the result of the case with random sampling, and Fig. 10 shows the result of the case with systematic sampling. These results are different from the result of the case with unsampled packets in the range wherein the high-frequency part appears upward along the Y axis from the origin. This is believed to be due to the normal communication explained in Section 4.1. In the case of communication in which the server performs a short session several times with the same node, packets are taken from more sessions by sampling than the case of using all packets. It is thought that this causes the cardinality of DPT to

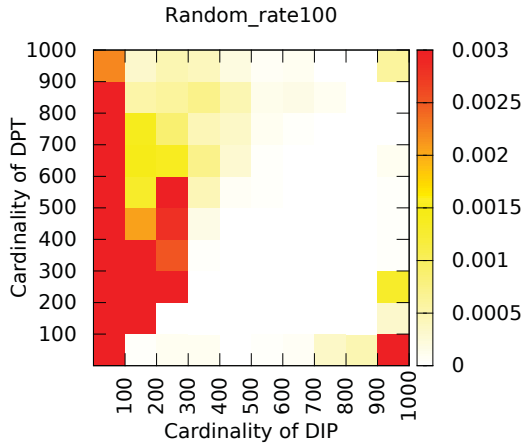

Fig. 9: Cardinality count using all random sampling packets by a sampling rate of 100

become high. In other cases, it is not notably different from the case without sampling. Therefore, there is a possibility that abnormal communication satisfying conditions c-1, c-2, and c-3 can be analyzed as producing the same result as in the case of using unsampled packets by random sampling and systematic sampling.

The result of systematic SYN sampling is shown in Fig. 11. With systematic SYN sampling, the relative frequency becomes high in a wide range. The number of reports in the same area as condition c-3 in the Fig. 7 is larger than that of the reports in the case without sampling. With systematic SYN sampling, there is a possibility that normal communication may be included in the range deemed abnormal in this paper. Therefore, to use systematic SYN sampling, it seems necessary to further analyze its usage.

5. Conclusion

In this study, we strived to verify whether the same analysis result can be obtained in the case of using all packets by cardinality analysis and using sampled packets.

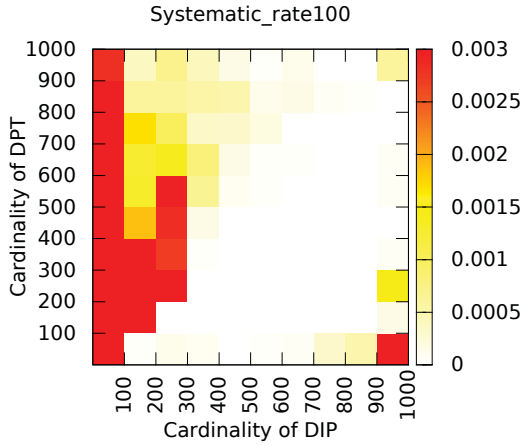

Fig. 10: Cardinality count using all systematic sampling packets by a sampling rate of 100

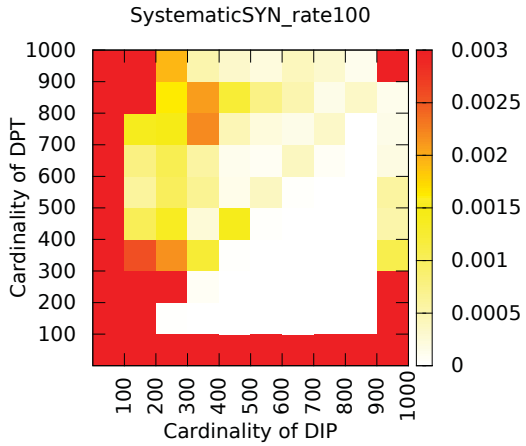

Fig. 11: Cardinality count using all systematic SYN sampling packets by a sampling rate of 100

We extracted TCP and UDP packets from MAWI Working Group. The 5-tuple itemset was extracted from the header information. First, we conducted a cardinality count of all TCP and UDP packets. We extracted from the reports of the cardinality count that SIP and SPT are the specific value combination, and PRT is TCP. From the extracted reports, we determined the condition of cardinality of DIP and DPT, which is considered to include abnormal communication. For each of the three IP addresses that showed a characteristic trend in the SIP of the reports satisfying the three conditions, we performed cardinality analysis for all TCP and UDP packets and its sampled packets individually. We extracted from the cardinality count reports that SIP and SPT are the specific value combination, and PRT is TCP. In each unsampled packet and sampled packet, we measured and compared the number of the reports satisfying the condition of DIP and DPT, which is considered to engage in abnormal communication.

In conparing the number of cardinality count reports in random sampling and systematic sampling, as the sampling rate decreases, the number of reports decreases. When the number of packets is small, the report is lost. However, if abnormal communication increases to disturb important communication, it is considered that it is possible to obtain the reports by cardinality analysis using random sampling or systematic sampling. With systematic SYN sampling, in the case of communication, such as scanning, which sends a high proportion of packets with a SYN flag, it is possible to obtain the reports close to the time when using all packets. However, it is possible that normal communication is analyzed as abnormal, and it is necessary to further verify the effectiveness of systematic SYN sampling by future investigation.

When analyzing TCP packets by cardinality analysis using sampled packets, it is considered necessary to select a sampling algorithm and sampling rate in accordance with the communication to be analyzed. In future work, we will investigate the impact of packet sampling on the result of cardinality analysis in UDP.

References

[1] Y. Shomura, Y. Watanabe, and K. Yoshida, "Analyzing the Number of Varieties in Frequently Found Flows," *IEICE Transactions on Communications*, vol. E91.B, no. 6, pp. 1896–1905, 2008.

[2] S. Mori, Y. Kato, A. Sato, and K. Yoshida, "Vicar in Network," *IEICE technical report*, vol. 114, no. 286, pp. 21–26, 11 2014.

[3] S. Sannomiya, A. Sato, K. Yoshida, and H. Nishikawa, "Cardinality Counting Circuit for Real-Time Abnormal Traffic Detection," in *2017 IEEE 41st Annual Computer Software and Applications Conference (COMPSAC)*, vol. 1, 7 2017, pp. 505–510.

[4] N. D. T. Zseby, M. Molina, "RFC 5475 - Sampling and Filtering Techniques for IP Packet Selection."

[5] T. Davide, V. Silvio, R. Dario, and P. Antonio, "Exploiting Packet - sampling Measurements for Traffic Characterization and Classification," *International Journal of Network Management*, vol. 22, no. 6, pp. 451–476, 11 2012.

[6] V. Carela-Español, P. Barlet-Ros, A. Cabellos-Aparicio, and J. Solé-Pareta, "Analysis of the Impact of Sampling on NetFlow Traffic Classification," *Comput. Netw.*, vol. 55, pp. 1083–1099, 04 2011.

[7] A. Salama, R. Saatchi, and D. Burke, "Fuzzy Logic and Regression Approaches for Adaptive Sampling of Multimedia Traffic in Wireless Computer Networks," *Technologies*, vol. 6, no. 1, 02 2018.

[8] A. N. Mahmood, J. Hu, Z. Tari, and C. Leckie, "Critical Infrastructure Protection: Resource Efficient Sampling to Improve Detection of Less Frequent Patterns in Network Traffic," *Journal of Network and Computer Applications*, vol. 33, no. 4, pp. 491–502, 07 2010.

[9] K. Giotis, C. Argyropoulos, G. Androulidakis, D. Kalogeras, and V. Maglaris, "Combining OpenFlow and sFlow for an Effective and Scalable Anomaly Detection and Mitigation Mechanism on SDN Environments," *Computer networks (Amsterdam, Netherlands : 1999)*, vol. 62, pp. 122–136, 04 2014.

[10] "MAWI Working Group Traffic Archive," accessed May 9, 2018. [Online]. Available: http://mawi.wide.ad.jp/mawi/

[11] M. Antonakakis et al., "Understanding the Mirai Botnet," in *26th USENIX Security Symposium (USENIX Security 17)*. Vancouver, BC: USENIX Association, 2017, pp. 1093–1110.

[12] M. C. S. Joáo, C. Paulo, and S. R. Lima, "Inside Packet Sampling Techniques: Exploring Modularity to Enhance Network Measurements," *International Journal of Communication Systems*, vol. 30, no. 6, p. e3135, 03 2016.