Visualization of the Packet Flows using Self Organizing Maps

HIROSHI DOZONO
Saga University
Faculty of Science and Engineering
1 Honjyo Saga
Saga
JAPAN
hiro@dna.ec.saga-u.ac.jp

TAKERU KABASHIMA
Saga University
Faculty of Science and Engineering
1 Honjyo Saga
Saga
JAPAN
kabasima@dna.ec.saga-u.ac.jp

MASANORI NAKAKUNI
Information Technology Center, Fukuoka University
8-19-1, Nanakuma, Jonan-ku
Fukuoka
JAPAN
nak@fukuoka-u.ac.jp

SHIGEOMI HARA
Saga University
Faculty of Science and Engineering
1 Honjyo Saga
Saga
JAPAN
hara@dna.ec.saga-u.ac.jp

Abstract: Recently, the spread of the Internet makes familiar to the incident concerning the Internet, such as a DoS attack and a DDoS attack. Some methods which detect the abnormal traffics in the network using the information from headers and payloads of IP-packets transmitted in the networks are proposed. In this research, the method for the analysis of the flow of IP packet based on SOM (Self-Organizing Map) using the occurrence rate of the elements in IP-packets as input vectors is proposed. The flow of the packets can be visualized on the map and it will be available for detecting the abnormal flows of packets.

Key–Words: Packet Traffic Analysis, Self Organizing Map, IDS, Network incident, TCP/IP

1 Introduction

Recently, the environment that can use the Internet becomes familiar along with the spread of computers. However, the environment that can use internet easily made the incident that used the Internet familiar as the risk.

The incident in the Internet tends to be increasing, and the ways of the virus infection and the DoS attack etc. of the computer diversify and become clever now. It is difficult to discover those all respectively using an identical techniques because the method of attack is different[1]. In this paper, the detection method of illegal computer access that assumes the DoS/DDoS attack that obstructs, and stops the provided service is proposed. These attacks are the techniques of putting the load on other party's network by keeping chiefly sending a similar packet, and obstructing the communication. It is necessary and indispensable to analyze the flow and traffic on the network to defend the server from such an attack. Then, it is considered that the earlier detection of an illegal communication using the header information and the payload of IP-pakcets is possible from the captured packet with extracting the statistical feature of the packets. Such a statistical packet analysis has already been proposed. In this study, the construction of the system that can visually confirm the change in the statistical feature of the packet by using Self-Organizing map (SOM)[2] that can map multi-dimensional data to two dimensions.

In the analysis of illegal computer access, [3] reported that it is effective for detecting a lot of DoS/DDoS attacks for a long time using statistical feature of the packet. The distribution of each element, such that IP address and port numbers, were approximated by the χ^2 distribution, and the deviation from the distribution at usual operation was detected as abnormality. In our study, it proposes the traffic analysis on the network which uses the occurrence rate of the each element in the header of the packet and payload information as the input data of SOM. Using this method, it becomes possible to analyze statistical relations of IP-packets composed of the multiple elements by integrating those using SOM and their relations can be visualized on the map. In this paper, the experiments of packet capture is limited in the firewall, so the IP Packets concerning attacks can not be captured. Thus, we aimed to develop SOM which can organize the map which reflects the time changes of the flow of packets visually. SOMs are already applied to the analysis of IP Packets for detecting abnormal communications[4][5]. They uses the information of packet directly not using occurrence rate of IP Packets. For analyzing large scale data using SOM, occurrence rates are often used and shows good result for clustering[6][7]. For the analysis of large number of IP-Packets, our method will be effective.

This paper is composed as follows. First of all, section 2 describes the self-organizing map, section 3 describes the method of acquiring the packet and the method of processing the packet data, and section 4 shows the packet analysis and the result that uses SOM. And, describe the conclusion and the future works in section 5.

2 Self Organizing Maps:SOM

2.1 Features of SOM

Self-organizing map (Self-Organizing Maps:SOM) [2] is the model of the neurologic function of the cerebral cortex developed by T.Kohonen, and the neural net work of the feedforward type of two layers using the algorithms of unsupervised learning.

The feature of unsupervised learning is a point that it is possible to extract the features hidden in input data automatically. SOM can compose the map of input signal space in the arranged shape and the network that selectively reacts to input data be composed because it can approximate the density function of input vectors.

SOM converts a nonlinear and statistical relations that exist between higher dimension data into the map with a simple geometrical relation. They can usually be used to make a higher dimension space visible because it is generally displayed as the lattice of the neuron in two dimensions. Moreover, it becomes easy to be able to visualize higher dimension information because it is possible to cluster without the preliminary knowledge, and to understand the relations among these multi-dimensional data intuitively for human.

From the features mentioned above, SOM has many variations of applications. We have applied SOM to the analysis and implementation of biometric authentications[8] and control of the autonomous robot using visual information[9].

2.2 Structure of SOM

SOM is basically a network in the double-layered structure with the input layer and the competitive layer as shown in Fig. 1. However, there are not connections between neurons in the same layer. The first layer is input layer $\mathbf{x}$ of n dimension, and the second layer is called a competitive layer, and is generally two dimension array for visualizing output data. It is assumed that it has the neuron that input data is given by $(x_1, x_2, ..., x_n)$ n dimension real number vector $\mathbf{x}$ =and two dimensions SOM were arranged on the grid point of $m * m(= M)$ units. Input data $\mathbf{x}$is presented to all neurons. Moreover, the neuron located in (j, i)in two dimension lattice array has $\mathbf{w_{j,i}} = (w^1_{j,i}, w^2_{j,i}, ..., w^n_{j,i})$ weight vectors which are tunable corresponding to the input vectors. This $w_{j,i}$is called as reference vector. The algorithm can be divided into two phases in a competitive phase and the cooperative phase. First of all, the neuron of the closest ,in a word, the winner neuron, is selected at a competitive phase. Next, the winner's weight are adjusted at the cooperation phase as well as the lattice near the winner.

2.3 Algorithm of SOM

Euclidean distance is used for the learning of SOM. The learning steps are as follows [2].

1. STEP1: Initialization of network
   An initial value of the weight between the input layer and the map layer is set to a small value by using random numbers.

2. STEP2: input of the reference vector
   Input the reference vector $\mathbf{x} = (x_1, x_2, x_3, \ldots, x_n)$ to input layer.

3. STEP3: Calculation of distance between input vector and weight vectors on the map layer The distance between each weight vector of each neuron on the competitive layer and input vector is calculated.

   The distance between the j-th neuron on the competitive layer and the input vector is given as follows.

$$d_i = \sqrt{\sum_{i=0}^{n} (x_i - w_{j,i})^2} \qquad (1)$$

   where $w_{j,i}$is the weight value between i-th neuron of input layer and j-th neuron of competitive layer.

4. STEP4: Competitive phase
   The neuron (winner neuron) to which the distance of the input vector and the weight vector is

the closest in a word to minimizing $d_j$is selected, and it is assumed j^*.

5. STEP5: Cooperative phase
A winner neuron and neighboring neurons are updated and there weight vectors is updated based on the following expression.

$$\Delta w_{j,i} = \eta h(j, j^*)(x_i - w_{j,i}) \qquad (2)$$

where $h(j, j^*)$ is called as neighboring function, and shown by the following expression.

$$H(j, j^*) = \exp\left(\frac{-|j - j^*|}{\sigma^2}\right) \qquad (3)$$

Learning coefficient σ becomes small with the progress of learning. Therefore, the range of $h(j, j^*)$in figure 2 is wide in the first stage of learning and becomes narrower with the progress of learning as shown in figure. Thus, neighboring function $h(j, j^*)$plays the role to form the map effectively. Here, η is a positive constant and called as learning coefficient.

6. Return to STEP6. STEP2
Weights are learned by repeating the operation of STEP2-STEP5 for all the input vectors. As a result, a similar units to each input vector come to gather on the map.

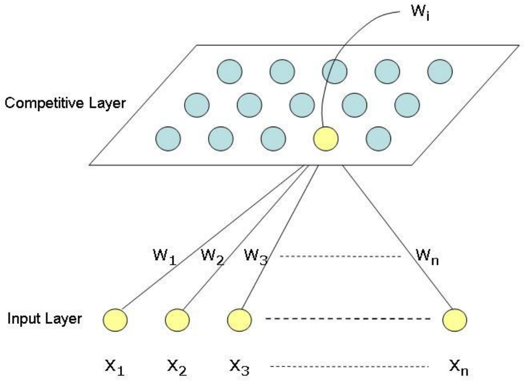


Figure 1: SOM structure

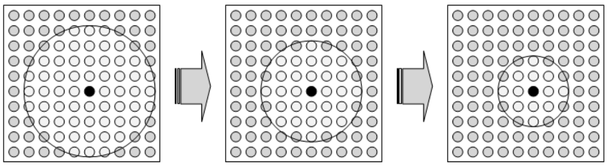


Figure 2: The neighboring radius of winner neuron

3 Packet Capture

3.1 The Basics of Packet Capture

The packet capture is to gather the IP packet that flows on an actual network. There are many commercial and free packet capture programs. In this study, the packet capture program was newly developed by using the library of the packet processing and captured packets are directly preprocessed to the input vectors of SOM. WinPcap is used as the packet capture library which are used in a lot of applications of Windows.

3.2 WinPcap

WinPcap is packet processing library for Windows, and used for purpose to record, to watch the communication data and to analyze them. WinPcap can be used directly in IP network under the Windows environment to access it. The application using WinPcap can be developed freely. In this study, the IP packets on the network are captured by using WinPcap.

3.3 Settings of Packet Capture

In this study, it experimented on all packets communicated in the bionics informatics laboratory of Saga University as a capture target. In addition, the targeted protocol was assumed to be TCP/IP.

The packet capture is done by the promiscuous mode. One of the packet or the packet of the broadcast address with a corresponding destination address to own MAC address of the network interface is read only in a usual mode. However, the promiscuous mode is one of the operational modes of the network card, and all packets that flow in the network are received and read in the promiscuous mode

Information on the address of transmission source and destination are recorded in the header of packet like figure 3. And, to which information that is called a header concerning the carrying data and the protocol has been added to the TCP Data. Figure 4 shows the actual packet data in capture. Such the numerical data keeps being consecutively flowing. The main body of data except the header part of the packet is called a payload, and the packet is composed of two parts(header and payload).

Moreover, when the packet capture is done, extra data other than the header and the payload are included in the captured packet data. Therefore, this extra data should be cut out, and process to pull out only necessary data is needed.

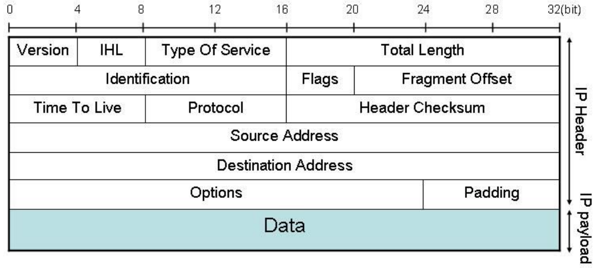
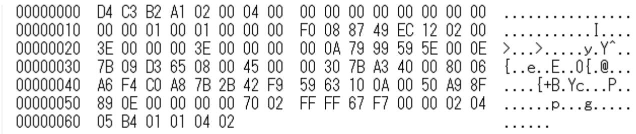


Figure 3: Format of IP header.



```
00000000  D4 C3 B2 A1 02 00 04 00  00 00 00 00 00 00 00 00  ................
00000010  00 00 01 00 01 00 00 00  F0 08 87 49 EC 12 02 00  ...........I....
00000020  3E 00 00 00 3E 00 00 00  00 0A 79 99 59 5E 00 0E  >...>.....y.Y^..
00000030  7B 09 D3 65 08 00 45 00  00 30 7B A3 40 00 80 06  {..e..E..0{.@...
00000040  A6 F4 C0 A8 7B 2B 42 F9  59 63 10 0A 00 50 A9 8F  ....{+B.Yc...P..
00000050  89 0E 00 00 00 00 70 02  FF FF 67 F7 00 00 02 04  ......p...g.....
00000060  05 B4 01 01 04 02                                 ......
```

Figure 4: Actual packet data.

3.4 Preprocessing of Packet Data

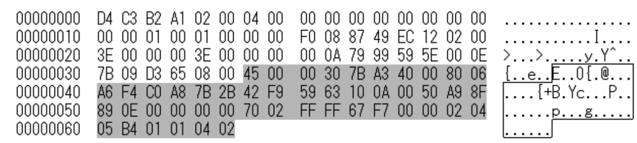
In addition to pull out necessary data from the packet data that does the capture previously described, they are preprocessed to use it for the input vectors of SOM .

The first step of the pre-processing is to find the beginning of the IP-header from captured packet data. The IP header is described in the format of figure 3, and the part of the packet are taken out based on it.

In this experiments, the packet capture was done on the assumption that ”4” the version of IP of LAN in the laboratory, and the service type were ”0”.

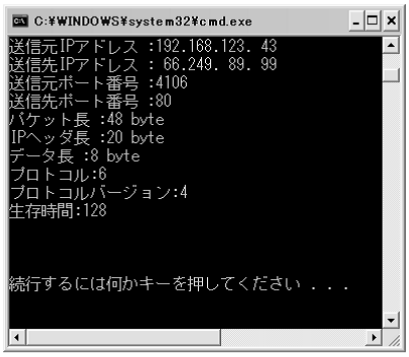
The many are ”4” or ”6. ” in the version of the IP header used now. In addition, it is thought that exclude ”6” can be excluded in this experiment when thinking about that IPv6 is not widespread yet. Moreover, the service type of almost all packets is ”0. ” usually. Therefore, the service type is assumed to be ”0”.

Therefore, when this condition is applied to the data of figure 4, the head of the packet is discovered and the part where the color reverses in figure 5 is target packet data.



```
00000000  D4 C3 B2 A1 02 00 04 00  00 00 00 00 00 00 00 00  ................
00000010  00 00 01 00 01 00 00 00  F0 08 87 49 EC 12 02 00  ...........I....
00000020  3E 00 00 00 3E 00 00 00  00 0A 79 99 59 5E 00 0E  >...>.....y.Y^..
00000030  7B 09 D3 65 08 00 45 00  00 30 7B A3 40 00 80 06  {..e..E..0{.@...
00000040  A6 F4 C0 A8 7B 2B 42 F9  59 63 10 0A 00 50 A9 8F  ....{+B.Yc...P..
00000050  89 0E 00 00 00 00 70 02  FF FF 67 F7 00 00 02 04  ......p...g.....
00000060  05 B4 01 01 04 02                                 ......
```

Figure 5: Data for one packet.



```
送信元IPアドレス :192.168.123. 43
送信先IPアドレス : 66.249. 89. 99
送信元ポート番号 :4106
送信先ポート番号 :80
パケット長 :48 byte
IPヘッダ長 :20 byte
データ長 :8 byte
プロトコル:6
プロトコルバージョン:4
生存時間:128


続行するには何かキーを押してください . . .
```

Figure 6: Extracted data for one packet

In addition, the numerical data which is pulled out and used as the input vector of SOM is shown in figure6. Information pulled out is IP address of source, IP address of destination, port number of source, port number of destination, packet length, a header length, a protocol type, and a protocol version.

4 Packet analysis using SOM

4.1 Direct learning of IP packet to SOM

The method of learning each one of the packet to SOM as an input vector is described here. Moreover, the average and the variance of the value of the payload were added to the input vector of SOM as information on the payload in addition to header information on figure 6. The dimension of the input vector becomes 19 dimensions as a whole. In this experiment, as for the input data used for learning, each input vector is made based on the data of one packet. Therefore, input data becomes a great number from the flow of the great number as for the packets of in one laboratory. Then, 10000 packets were captured for this experiment, and it used for learning.

In that case, one packet is learned once by SOM and the radius of neighborhood is converged from half of the map to 1.5. Using this learning method, even when the number of packets increased greatly, new input data was able to be reflected in the map. This becomes very important for analyzing the packet in real time.

The map for forward mapping of input data to SOM after learning is shown in figure 7 . This map is made by the gray scale for the color to approach the black from white according to the capture time of the packet. ”—” is displayed in the neuron on map which are not selected as winner for all input vectors. Ideally, the changes of the color on the map becomes continuously if the map reflects time changes of the packets. It is not arranged so that the similar color neurons are close on the resulting map, and it is thought that it is difficult to visualize a time change of the packets on the map in this method. It is considered that a packet is not significant and packet transmission becomes significant with the group of consecutive packets. Then, it is considered that learning the packet as a group is necessary.

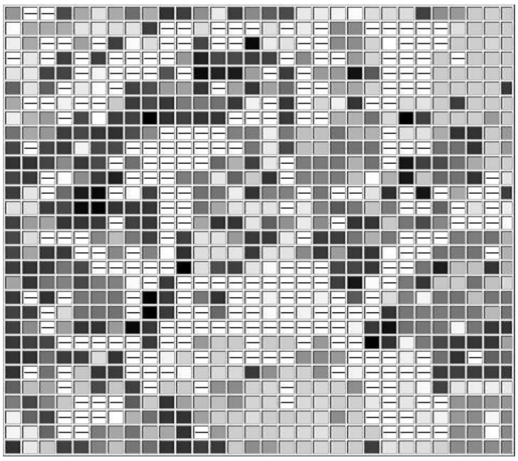


Figure 7: The map when learning a packet as a vector.

4.2 Learning by Occurrence Rate of Packets

It is considered that the feature of a time change in the packets (flow) is able to be extracted by bringing the packets together in the group and composing one input vector for SOM using the packets in one group which is included in a window. Then, the learning method which uses the occurrence rate of the packets in the window is proposed.

Moreover, in the illegal computer access of the DoS attack etc, detection is difficult only by checking the packet one by one because the attacking method is continuously sending the same packets. Then, using the window previously described becomes effective. It is necessary to analyze the packets in a windows of fixed time as a group to find the consecutive packets. Then, the flows of the communication of the packets are analyzed with the occurrence rate of the packets in the window that had been set by a pre-defined width in this experiment.

4.3 Packet Analysis using Statistical Information

In the paper ”Detection of unlawful computer access by a statistical analysis” [3], it was assumed that the value of IP address and the port number of the packet to be a random variable of χ^2 distribution , and illegal computer access was detected as abnormal distribution, and it succeeded in detecting the DoS attack. It can be considered that statistical information is efficient measures for detecting illegal computer access. The group of the packets was called a window in this research, and the best width was 500 packets. However, Internet Protocol address and the port number were handled as an independent random variable , and the analysis that integrates them was not done.

Here, we propose a method which uses the occurrence rate of the packet as input vector of SOM. In this method, it is considered that it is possible to analyze by integrating those information by composing the occurrence rate of two or more elements in the packet as an input vector on the map of SOM.

4.4 Composition of input Vector using Occurrence Rate

The calculation method of the occurrence rate of the packet in this study is shown below. First of all, the vector with the number of dimension shown in table 1 is prepared in each window. The source and destination IP-addresses are divided into four digit respectively, the each digit is 8 bits, and each address in IP is 32 bits in total . Then, the frequency of Internet Protocol address can be described by preparing the vector of 256 dimensions for each number of eight bits and counting to each element according to the value of IP address. Figure **??** shows an example of counting IP address and the configuration of input vector.

Table 1: Vector for frequency of appearance

Parameter	Dimension
Source IP-address	256 × 4
Destination IP-address	256 × 4
Source port number	5
Destination port number	5
Packet length	2
Payload data	256

When the input vector 192,168,56.18 is given, for the first 8 bit data 192, the element which represents the occurrence of 192 is incremented. For other 8 bit data, 168, 56 and 18, the elements are also incremented as shown in Figure **??**

As for port numbers, because there are a lot of uselessness as for the preparation for the vector for all of possible port numbers described in 16 bits(65536 combinations, and the port number used for illegal

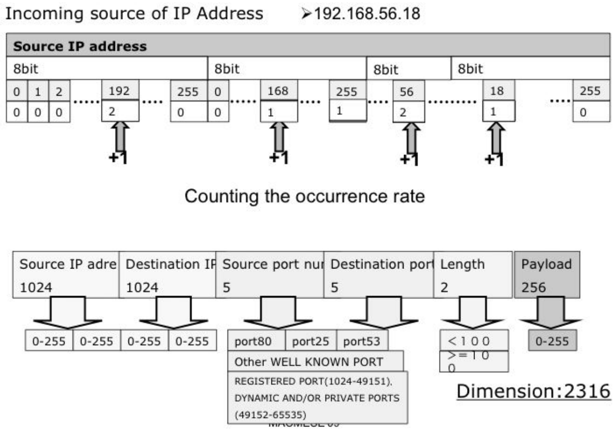


Figure 8: Input vector composed of occurrence rate.

computer access on the port is concentrated to the well-known port, the port number which is considered to be effective for detecting DoS attack is squeezed to five kinds. Then, frequency was measured by the source and destination port number in ten dimensions in total with dividing them into well-known ports to retrieve TCP/25 for mail communication, TCP/80 for WEB, UDP/53 for looking up host IP-address, well-known port except the three ports described before and other ports as same as the paper [3]. As for packet length, the threshold was assumed to be 100, and it is counted into two dimensions of vector according to the threshold. As the frequency of the value of every eight bits in payload, the vector of 256 dimensions was prepared, and counted by the value.

4.5 Experimental results using the Occurrence Rate as Input Vectors

The results of experiments with three kinds of window widths (100 packets, 500 packets, and 1000 packets) are shown. The other conditions are same for these experiments.

The condition is shown in table 2. The numbers of packets used for learning in the case of window size 100, 500 and 1000 are 10000, 50000 and 100000 respectively. The learning result of SOM when the width of the window is adjusted to 100,500 and 1000 is shown in figure 9 , figure 10 , and figure 11 respectively. In these figures, the flow of time is shown with the color of the node in the map as changing from white to the black with the time passing.

Table 2: The learning condition in SOM

Parameters	Conditions
Size of map	30×30
initial size of neighborhood	15
Initial learning coefficient	1
Learning iterations	300
Number of learned windows	100

The wider the width of the window is, the more continuously the color of the neurons is changing. It is considered that the wide one is able to cluster in shape mapping a near window by time adjacently, and is able to visualize the temporal changes of the packet on the map. In the case of width of the window is 100, the color of the neuron in the map is fragmented. It is considered that the number of packets is insufficient to find the feature of packet by time for this case. The mapping result of window width 500 is more consecutive compared with that of window width 1000. We use the window width 500 because smaller value is preferred considering the implementation of this method to real time analyzing tools considering the response time for incidence.

4.6 Learning with Weight for Each Element

In the method of counting described in the previous section, IP-address is 256 dimensions and the port number is 5 dimensions. So they greatly have the difference in the number of dimension in input vector , and have a big difference about one packet in the increasing rate of each element of input vectors. Moreover, the amount of the change of the size of norm is different in the payload depending on the packet data length. It is considered that the size of average norm of each element affects the learning. Then, the input vector is composed putting different weight for each element respectively to adjust the size of norm. The result of experiment is shown in figure **??**. Clustering is done in consecutive timewise as well as the result shown in previous section and the changes of the color becomes more consecutively.

4.7 Learning with Overlapped Window

Because of the setting of the width of the window depending on the number of packets etc, the communication might be divided to some windows while one communication is continuing. Then, one communication may be mapped independently by SOM. To solve this problem, Some overlaps are given between windows. Moreover, the similarity of the consecutive windows can be increased by this, and it is considered that the temporal changes become to be clustered more visually.

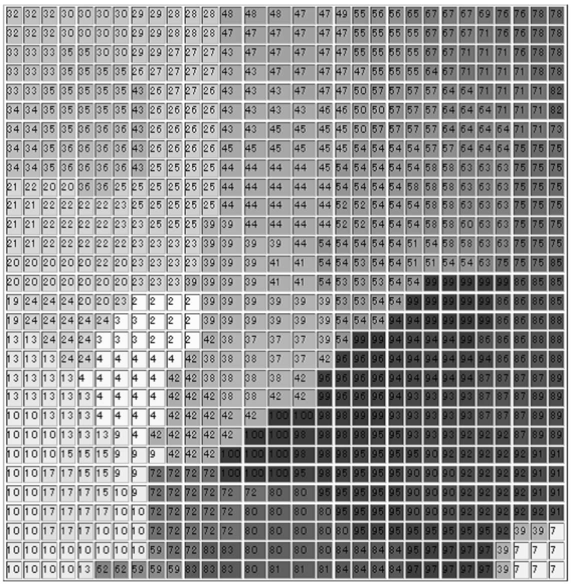


Figure 9: The map for dividing one window by 100 packets

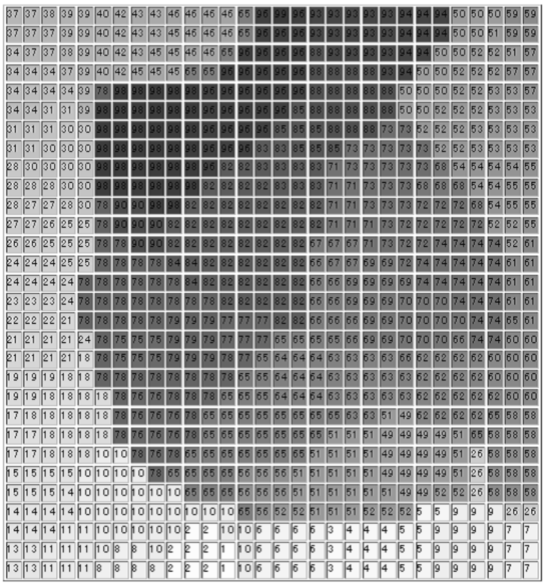


Figure 10: The map for dividing one window by 500 packets

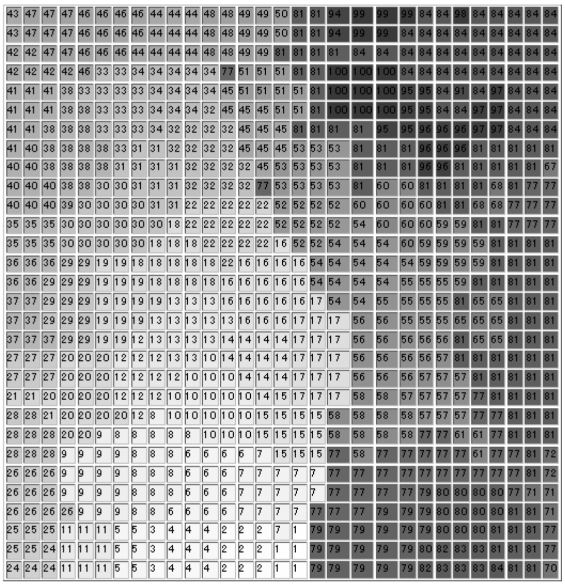


Figure 11: The map for dividing one window by 1000 packets

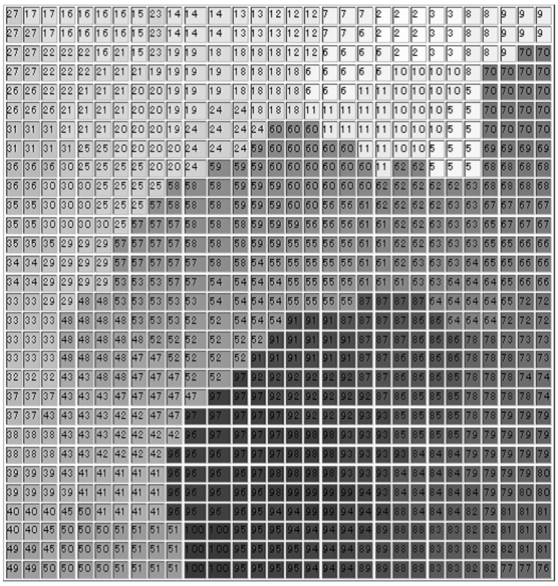


Figure 12: The map using weighted elements

Here, the width of overlap was adjusted to 10%, 20%, and 50% for the window width. Using 10% overlapped window, the changes of color becomes more consecutive. The larger overlapping window shows more consecutive mapping than the result shown in Figure 12 and with increasing the overlap to 20%, the map becomes more consecutive again. But, the map becomes a little fragmented for 50% overlapped window. Using larger size of overlapped window, the amount of calculation for learning becomes larger and it becomes more difficult to apply this method for real time analysis of packet flow. So, we selected the size of overlapping as 10%.

4.8 Visualization of Packet Flow on the Map

In the previous subsection, the flow of the packet is visualized as the changes of the grayscale levels labeled on the map. In this section, the packet flows are visualized more concretely as arrows on the map. Figure 16. 17 and 18 show the maps with the arrows which denote the packet flows by time of Figure 10, 12 and 13 respectively. The packet flow shown in Figure 16 ,which is organized without using weight values, is continuous at beginning, but becomes fragmented on the upper right side on the map. The packet flow shown in Figure 17 ,which is organized with using weight values, is more continuous, but it also becomes fragmented on the lower right side of the map. The packet flow shown in Figure 18 ,which is organized with using weight values and 10% overlapping windows, is also continuous and maps the packet flow from lower right side → upper right side → upper left side → lower left side and it is also reflected to the change of colors on the map. From, those results, the parameters which are used for the analysis are selected as follows.

- Window size: 500
- Weight for each element: The weight which normalize each element to 1
- Size of overlapping: 10%

4.9 Experiments using Larger Scale Data

Next, we made experiments using larger scale data. To capture large number of packets, the capturing point was changed from inside of the laboratory to outside of the router of the laboratory. However, the incoming packets are filtered by the university firewall, so the packets concerning attacks can not be captured. Figure **??** shows the map using 1000 windows(500000 packets) for learning. The size of the map is changed to 50x50. Compared with the previous results, the

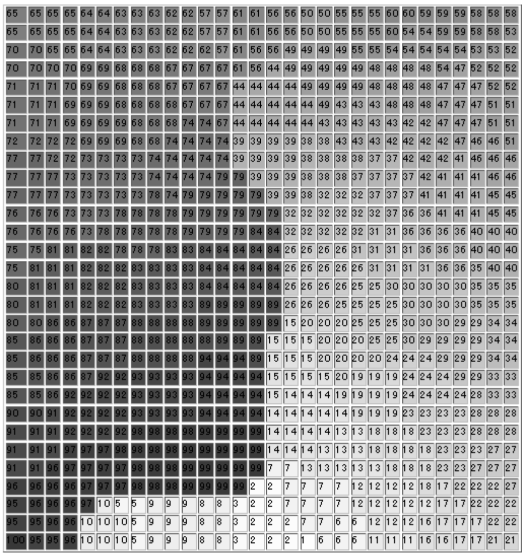


Figure 13: The map using 10% overlapped window

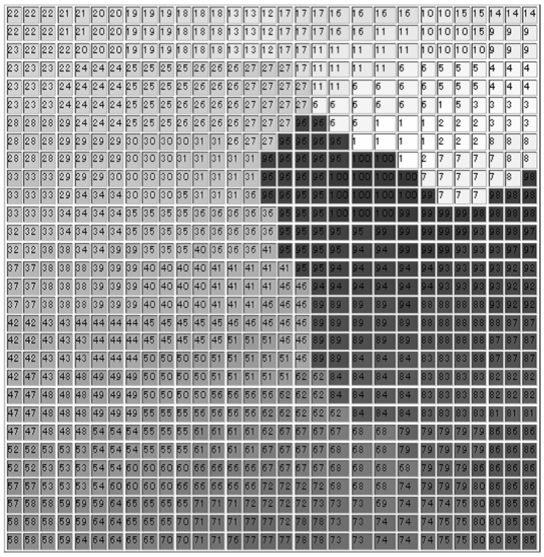


Figure 14: The map using 20% overlapped window

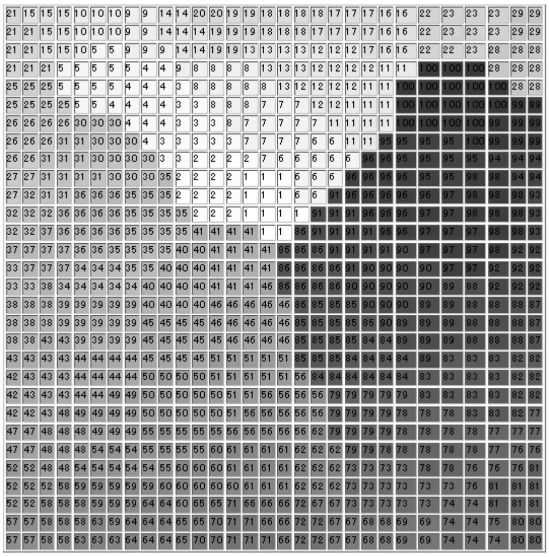


Figure 15: The map using 50% overlapped window

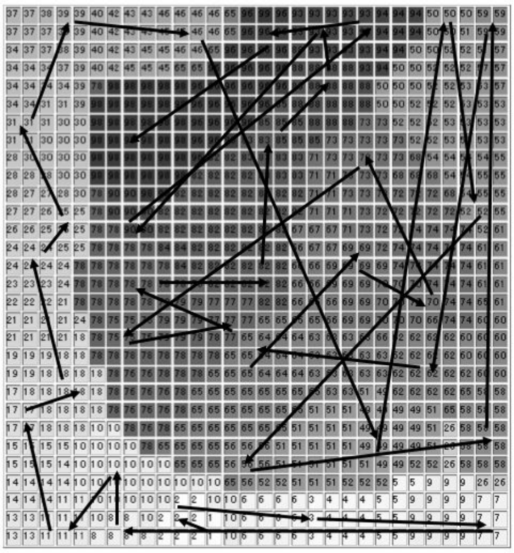


Figure 16: Packet flow of the map in Figure 10

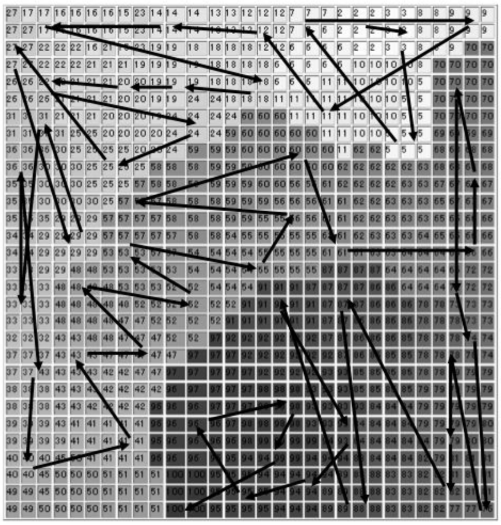


Figure 17: Packet flow of the map in Figure 12

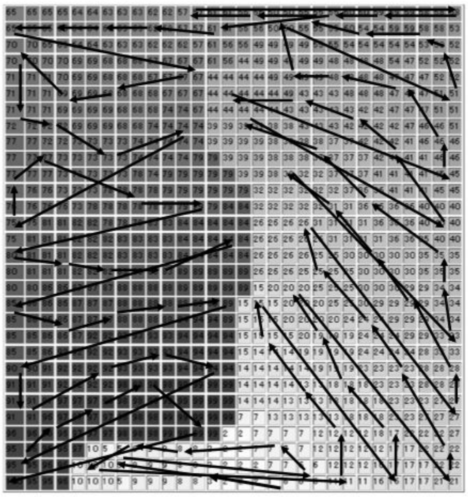


Figure 18: Packet flow of the map in Figure 13

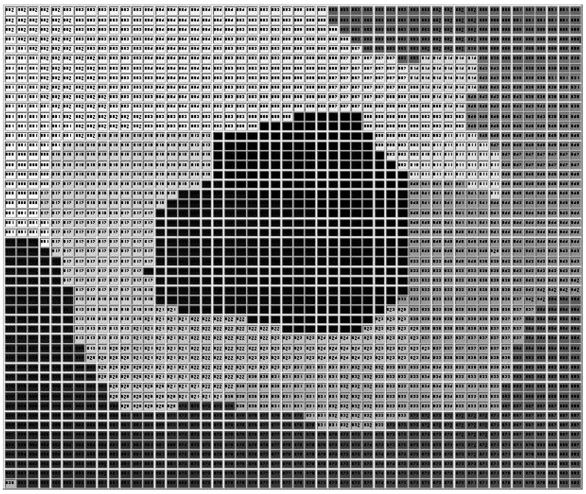


Figure 19: The map organized with 1000 windows of packets

changes of the greyscale levels becomes fragmented, however, it still continuous locally. It is considered that similar patterns of the packets flows are repeated while the larger number of packets are captured in longer time scale, and they are mapped to the same units on the map.

5 Conclusion

In this paper, the clustering method of frequencies of the packet traffics by using SOM is proposed as one of the statistical mode of analysis of the IP packet. Moreover, it experimented on clustering of the packet by SOM using the packets captured by the packet capture program using WinPcap.

When the packet was learned one by one as the input vector, the map of SOM did not become the one that a time change in the packet was reflected. On the other hand, it became possible to organize the map that reflected a time change in the communication situation by composing the input vector using the occurrence rate of the packet in the window composed of the group of the packet. In addition, in making a consecutive overlapped window, the map that reflects the continuousness of the packet transmission can be organized without cutting up the continuous communications. It is considered that the network traffic based on the occurrence rate of each packet elements can be analyzed by SOM.

Using the occurrence rate as input vector, the amount of computation becomes large because of the size of input vector and it may be difficult to real time analysis of captured packet. In addition, bias are seen in the captured packets because the scale (place for capturing and amount of packets) on which the packets are captured is too small in the experimental environment, and the packets of the attacks such as actual DoS can not be captured because we can only make experiments inside the firewall now. Therefore, it is considered that the experiment in the environment without the restriction by the firewall etc. is necessary.

References:

[1] S. Malliga, A. Tamilarasi, A Proposal for New Marking Scheme with its Performance Evaluation for IP Traceback *WSEAS TRANSACTIONS on COMPUTER RESEARCH,Issue 4, Vol.3,2008* , pp.259-272,(2008)

[2] T. Kohonen, *Self Organizing Maps, Springer*, ISBN 3-540-67921-9

[3] S. Oshima,T. Nakashima,T. Sueyoshi, Extraction of Anormaly Access using Statistical Analysis *IPSJ SIG Notes*, pp.289-294,2009(20),(2009)

[4] K. Ohkouchi, K. Rikitake, K. Nakao, A Study on Network Incident Analysis Using Self Organizing Maps, *Proceedings of the 2006 Symposium on Cryptography and Information Security*,(2006)

[5] K. Kanenishi, S. Togawa, K. Matsuura,H. Mitsuhara, Y. Yano, Aberrant Detection from Behavior of Campus Network Traffic, *Journal of Academic Computing and Networking*, No.13, pp.74-83,(2009)

[6] K. Lagus,S. Kaski, T. Kohonen, Mining Massive Document Collections by the WEBSOM method, *Information Sciences*, Vol.163/1-3,pp.135-156,(2004)

[7] T. Abe, T. Ikemura,et.al, A Novel Bioinformatics Strategy for Phylogenetic Study of Genomic sequence Fragments: Self Organizing Map (SOM) of Oligonucleotide Frequencies, *Proceedings of 5th Workshop on Self Organizing Maps*, pp.669-676,(2005)

[8] Masanori Nakakuni, Hiroshi Dozon ,et.al, Application of Self Organizing Maps for the Integrated Authentication using Keystroke Timings and Handwritten Symbols, *WSEAS TRANSACTION on Information Science and Applications, Issue4, Volume 2, 2007*, pp.423-440,(2007)

[9] Hiroshi Dozono,et.al, Visual Reinforcement Learning Algorithm using Self Organizing Maps and Its Simulation in OpenGL Environment, *WSEAS TRANSACTIONS on Information Science and Applications,Issue 5, Volume 5, 2008*, pp.685-694,(2008)