# Machine Learning Based Autonomous Network Flow Identifying Method

Hongbo Shi[1,3], Tomoki Hamagami[1,3], and Haoyuan Xu[2,3]

[1] Division of Physics, Electrical and Computer Engineering,
Graduate School of Engineering, Yokohama National University
[2] Information Technology Service Center, Yokohama National University
[3] 79-1 Tokiwadai, Hodogaya-ku, Yokohama 240-8501 Japan
{shi,hamagami,haoyuan}@ynu.ac.jp

**Abstract.** Recently, various applications and services start to be used in the Internet. Load balancing the increasing network traffic in real time can affect the network quality. The flow control technologies become much more important than before. Our research project proposes an intelligent network flow identifying method, smart flow, which is based on the learning algorithm. In this paper, we suggest to utilize the SOM for learning the properties of packets, such as timestamp, source and destination. Based on our proposed normalization, IP network flows can be formed autonomously during the learning process. Furthermore, the combination use of the new normalization with the GHSOM can classify the sub-IP flows belongs to the same flow. This paper indicates that a flow shall consist of several sub-IP flows, and sub-IP flow shall consist of several IP packets.

**Keywords:** IPv6, SIP, IP flow, SOM, GHSOM, classification.

## 1 Introduction

Now days, the principal technology of the Next Generation Network (NGN), IP telephony, is used widely in the world. It also causes the network traffic to be increased much faster than before. Until now, the packet is used as the network traffic unit for routing control. Meanwhile, these years, a new unit called IP flow is started to be used for describing a stream of packets in the Internet. The network packets grouped in the same IP flow usually means the packets belong to the same source and destination. Thus, a large amount of packets generated by the web and P2P applications can be routed as a single IP flow. This kind of new flow routing can save much more transmission cost than the traditional packet routing. Because the new flow routers do not need to route every packet, but the first packet of a flow. Utilizing the IP flows for the network load balancing and network analysis becomes more important.[1] Meanwhile, these products and standards group the packets to a flow not just only based on the source, destination and ports, but also a very principle parameter, time interval, which can be configured to different values by the administrators or affected by the memory size of a router.

P. Ren et al. (Eds.): WICON 2011, LNICST 98, pp. 447–457, 2012.

There are several network management products, such as sFlow[2], NetFlow[3], Openflow[4] used popularly in the world. There is also an international standard called IPFIX (IP Flow Information Export) in the Internet Engineering Task Force (IETF). [5][6] Due to using these new technologies, the network management can watch much larger network scale and analysis network condition in more detail than the IP packet based tools. Furthermore, network intrusion caused by the DDoS and worm can be detected quickly by using the flow collectors.

However, these flow technologies are based on the packet sampling and filtering algorithms. These existing algorithms may limit the watching scale of a target network and affect the characteristics of the original IP flows generated in the network. Moreover, the time interval for the packet sampling or filtering is configured by the network operations individually. In other words, network management is based on the administrator's experiment. Different managements used in a network may generate different flows. It causes that IP flow modeling difficultly.

This paper suggests a new solution for generating IP flow autonomously by using the machine learning, Self-Organizing Maps (SOM).[7] We suggest to group the packets with similar characteristics in a flow based on learning the packet information included in the packet headers. The new suggestion can generate IP flows autonomously without using the time interval and without being affected by any kinds of network managements.

## 2 Self-Organizing Maps, SOM

Self-Organizing Maps[7] suggested by T. Kohonen, is a kind of neural network algorithms. It is known as a data visualization technology for those high-dimension data during the competitive learning of Euclidean distance. Furthermore, it is mainly used for unsupervised learning, clustering, classification and data mining. The SOM algorithm in detail is shown as follows.

1. Initialize the weights of the nodes in the map, $m_i$, with random numbers
2. Use Euclidean distance to find similarity between the input vector and the weight vector of a node in the map. Winning unit $m_c$ is most similar one, also called Best Matching Unit (BMU).

$$m_c(t) = min_i||x(t) - m_i(t)|| \tag{1}$$

3. Update the neighborhood of BMU to be closer to the input vector through the SOM learning. $h_{ci}$ is the neighborhood function.

$$m_i(t+1) = \begin{cases} m_i(t) + \alpha(t) \cdot h_{ci}(t) \cdot [x(t) - m_i(t)], \\ \qquad\qquad\qquad\qquad (i \in N_c) \\ m_i(t), \ (i \notin N_c) \end{cases} \tag{2}$$

4. The learning rate ($0 \leq \alpha(t) \leq 1$) and the neighborhood ($N_c = N_c(t)$) decrease with time. T means the number of learning times.

$$\alpha(t) = \alpha_0(1 - t/T) \tag{3}$$

$$N_c = N_c(0)(1 - t/T) \tag{4}$$

## 3 Growing Hierarchical Self-Organizing Map

The map size of SOM is required to be defined properly due to the users' experiment. Moreover, SOM uses the input data for learning and training the map. SOM doesn't have neither recursive mechanism for the coming data input in real time nor evolution mechanism for growing map. Thus, amount of data input may cause the difficult decision for the map size. The Growing Hierarchical Self-Organizing Map proposed by A. Rauber provides dynamic architecture for maps to grow in both hierarchical way and horizontal way. Thus, GHSOM can treat amounts of input data during an unsupervised training process due to its dynamically growing hierarchical map architecture. Fig. 1 shows that, Layer 0 only includes one unit and Layer 1 includes 2x3 units. There are 6 SOMs in the Layer 2. One for each unit in the Layer 1 map. Layer 3 has 2 SOMs which are expanded from the units in Layer 2.

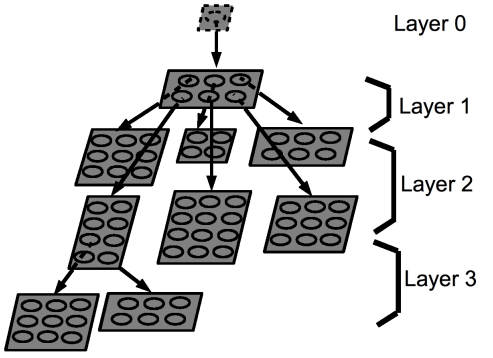


**Fig. 1.** Growing Hierarchical Mechanism of GHSOM

GHSOM growing process is started at a virtual Layer 0, which consists of only one single unit. Layer 0 is used for controlling the hierarchical growing process. The weight vector of this unit is initialized as the average of the entire input data. The unit is assigned a weight vector,

$$m_0 = [\mu_{0_1}, \mu_{0_2}, ..., \mu_{0_n}]^T \tag{5}$$

The expression,

$$mqe_0 = \frac{1}{d} \cdot ||m_0 - x|| \tag{6}$$

shows the deviation of the input data, the mean quantization error of the single unit. $d$ means the number of the input data $x$.

Training of the GHSOM is started at Layer 1 which is initialized with a grid of 2x2 units. Each unit is initialized with an n-dimensional weight vector,

$$m_i = [\mu_{i1}, \mu_{i2}, ..., \mu_{in}]^T, m_i \in \mathcal{R}^n \tag{7}$$

GHSOM uses the learning policy,

$$m_i(t) + \alpha(t) \cdot h_{ci}(t) \cdot [x(t) - m_i(t)] \tag{8}$$

The expression,

$$MQE_m = \frac{1}{u} \cdot \sum_i mqe_i \tag{9}$$

shows the mean quantization error of a map. $u$ means the number of unit $i$ in SOM $m$.

There are two parameters, $\tau_m$ and $\tau_u$, used for controlling the growth of GHSOM. $\tau_m$ is used to manage the growth of each map, and $\tau_u$ is used to control the hierarchical growth. If $MQE_m \geq \tau_m \cdot mqe_0$, then a new row of units or a new column of units is inserted between the error unit, $e$ and the dissimilar unit, $d$. (Fig. 2) The error unit means the highest $mqe_e$ unit, after $\lambda$ training iterations. The dissimilar unit means the neighbor of error unit with the most dissimilar weight vector. The weight vectors of the new units are initialized simply with the averages of the weight vectors of the existing neighbors. After the insertion, the learning rate ($\alpha$) and neighborhood function ($h_{ci}$) are reset to their initial values and the training based on the traditional SOM process.

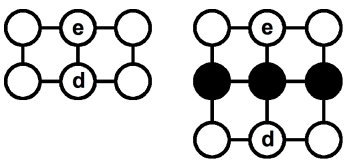


**Fig. 2.** Unit Insertion of GHSOM

After the growth of Layer 1 ($MQE_m < \tau_m \cdot mqe_0$), hierarchical control process in Layer 2 is started. The selection of unit depends on the $mqe$ of Layer 0. Each unit, $i$ begins the hierarchical expansion while the criterion, $mqe_i > \tau_u \cdot mqe_0$, is fulfilled.

## 4 New Proposal: Autonomous IP Flow Identifying

This paper focuses on the classification feature of SOM. We suggest to learn the characteristics included in the IP packet for identifying IP flow autonomously. The new proposal normalizes the timestamp , source and destination information included in IP packet as the input vector for the SOM learning.

### 4.1 Input Vector Normalization

In the Internet, hop or cost is used to express the distance between two nodes. The timestamp of a captured IP packet is usually affected by the distance between nodes, packet size, upper layer application and the network congestion. Therefore, this paper suggests to utilize the timestamp for estimating the network distance.

This paper also suggests to map the network peer communication to a vector. When there are $j$ nodes in a network, then the communication peers can be mapped to a $2j$-dimension vector space. (Fig. 3) In other words, $2j$-dimension vector space is correspondent to a mesh network in this paper. $Packet\_Header$ means the set of IP addresses included in an IP packet. In our proposal, $Packet\_Header(t)$ shows the source and destination IP addresses due to the time window ($0 \leq t \leq T$, $t \in \mathcal{N}$).

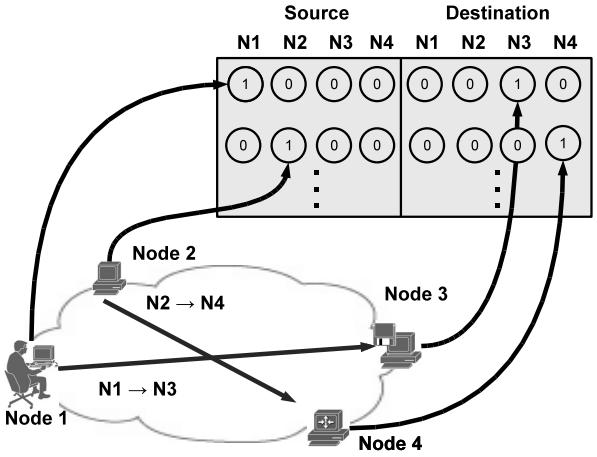


**Fig. 3.** Map Communication Peer to High Dimension Vector

$$Source_{init}(t)(j) = \begin{cases} 1, & (j \in Packet\_Header(t)) \\ 0 \end{cases} \tag{10}$$

$$Destination_{init}(t)(j) = \begin{cases} 1, & (j \in Packet\_Header(t)) \\ 0 \end{cases} \tag{11}$$

According to the timestamp, following formulas are used for normalization.

$$TIME(t) = \frac{(Timestamp(t) - Timestamp_{start})}{Counter(t)} \tag{12}$$

$Timestamp(t)$ means the record time of the current captured packet. $Timestamp_{start}$ is the start of the time window. $Counter(t)$ shows the order number of captured IP packets. This paper uses

$$\alpha(t) = \sum_{t} \frac{TIME(t)}{Counter(t)} \tag{13}$$

to normalize the timestamp, source and destination information included in IP packet as follows.

$$Source(t)(j) = \begin{cases} \alpha(j) \cdot Source_{init}(t)(j), \\ \qquad\qquad (j \in Packet\_Header) \\ 0 \end{cases} \tag{14}$$

$$Destination(t)(j) = \begin{cases} \alpha(j) \cdot Destination_{init}(t)(j), \\ \qquad\qquad (j \in Packet\_Header) \\ 0 \end{cases} \tag{15}$$

### 4.2 Experiment

This proposal uses the network traffic measured in the communication of remote video conference which uses IPv6 and SIP. (Fig. 4).

There are 11 hosts used in the remote video conference experiment. Thus, 22 dimensions are used for mapping the network topology. Due to our proposed normalization as introduced in Sec. 4.1, the network traffic measured by tcpdump (Fig. 5) can be normalized to 23-dimension vectors (Fig. 6) for SOM learning. This paper uses 150 normalized IP packets which are captured in above network. The 23 dimensions show the timestamp, source and destination of an IP packet.

### 4.3 Autonomous IP Flow Identifying

Fig. 9 shows the result of the SOM map training on the normalized characteristics of IP packet. The labels shown in the map consist of the captured order and parts of the source, destination IP addresses of an IP packet. For example, the label, 46.105.da8, means the 46th captured IP packet, the source is 2001:2f8:37::241:105 (omitted as 105) and the destination is 2001:da8:8005:1191:892:8385:1c85:5201 (omitted as da8).

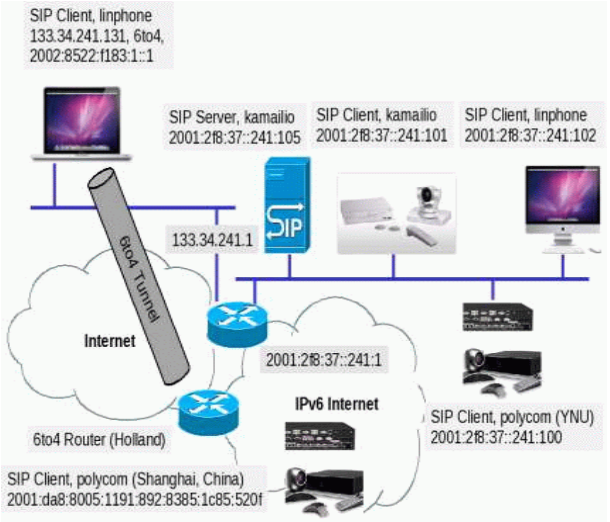


**Fig. 4.** SIP on IPv6 Testbed

```
12:41:56.289728 IP6 (hlim 128, next-header: ICMPv6 (58), length: 16) 2001:2f8:37::241:108 > 2001:2f8:37::241:102:
[icmp6 sum ok] ICMP6, echo reply, length 16, seq 1
12:41:59.669286 IP6 (hlim 64, next-header: UDP (17), length: 12) 2001:2f8:37::241:102.sip > 2001:2f8:37::241:105.sip:
[udp sum ok] SIP, length: 4
12:42:00.410380 IP6 (hlim 255, next-header: ICMPv6 (58), length: 32) 2001:2f8:37::241:102 > 2001:2f8:37::241:108:
[icmp6 sum ok] ICMP6, neighbor solicitation, length 32, who has 2001:2f8:37::241:108
12:42:00.410495 IP6 (hlim 255, next-header: ICMPv6 (58), length: 32) 2001:2f8:37::241:108 > 2001:2f8:37::241:102:
[icmp6 sum ok] ICMP6, neighbor advertisement, length 32, tgt is 2001:2f8:37::241:108, Flags [solicited, override]
12:42:01.638506 IP6 (hlim 255, next-header: ICMPv6 (58), length: 32) 2001:2f8:37::241:1 > 2001:2f8:37::241:105:
[icmp6 sum ok] ICMP6, neighbor solicitation, length 32, who has 2001:2f8:37::241:105
```

**Fig. 5.** tcpdump of SIP with IPv6

```
23
#time n1 n2 n3 n4 n5 n6 n7 n8 n9 n10 n11 n1 n2 n3 n4 n5 n6 n7 n8 n9 n10 n11
0041800001054 0 0 0 0 0 0.090096826667359 0 0 0 0 0 0.090096826667359 0 0 0 0 0 0 0 0 0 0 5.108.102
0.729961166666423 0.19674088333387 0 0 0 0 0 0 0 0 0 0 0 0.19674088333387 0 0 0 0 0 0 0 0 0 0 6.102.105
0.731551571429009 0.273142410204604 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0.273142410204604 0 0 0 0 0 7.102.108
0.640121999999792 0 0 0 0 0 0.319014858929002 0 0 0 0 0 0.319014858929002 0 0 0 0 0 0 0 0 0 0 8.108.102
0.705443000000539 0 0 0 0 0 0 0.361951319048062 0 0 0 0 0 0.361951319048062 0 0 0 0 0 0 0 0 0 0 9.1.105
```

**Fig. 6.** Normalized Input Data for SOM and GHSOM Training

As snown in Fig. 9, the IP packets sent by node da8 to node 105 are grouped in a flow autonomously. As the same, the network communication between the nodes (labelled with 102.105, 105.102, 100.105 and 105.100) in 2001:2f8:37::241::/112 network are identified to several IP flows properly. The IP packets (labelled with 2002.105) sent by using IPv6 over IPv4 tunneling technology are also grouped in a flow properly. Because of the incomplete normalization for the start of input data, some of the IP flow classification is treated correctly.

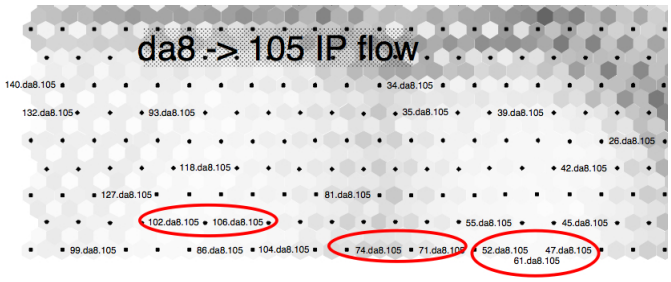


**Fig. 7.** IP Flow identified by SOM

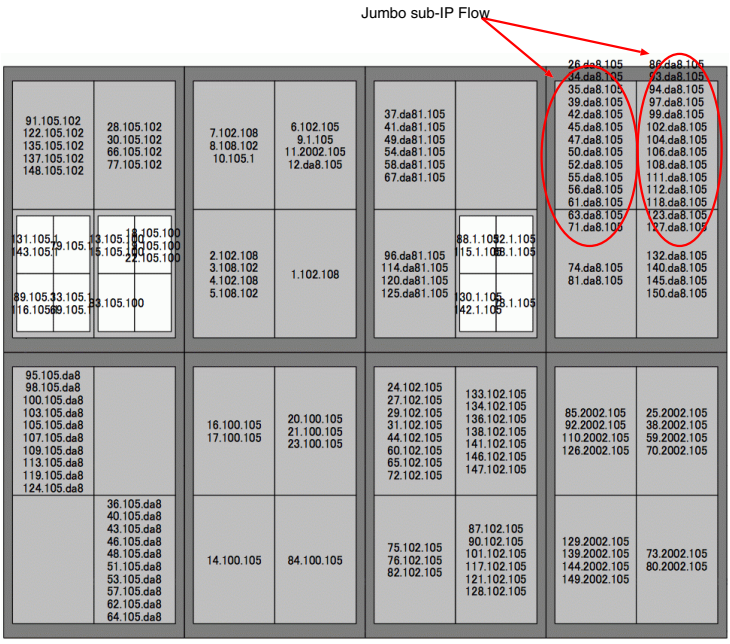


**Fig. 8.** IP Flow and sub-IP Flow generated by GHSOM Hierarchical Architecture

### 4.4 IP Flow in Hierarchical Classification with GHSOM

Usually, the network communications in the Internet may generated randomly. Communication between two nodes may consist of multiple IP flows. In other words, an IP flow may consist of multiple sub-IP flows in real Internet. Due to the growth of GHSOM map size, GHSOM shows the flow classification result more clearly than SOM. Because of the hierarchical architecture, there is more training on the characteristic, timestamp, in GHSOM than in SOM. These reasons cause that sub-IP flow and flow hierarchical architecture become visible in GHSOM map.

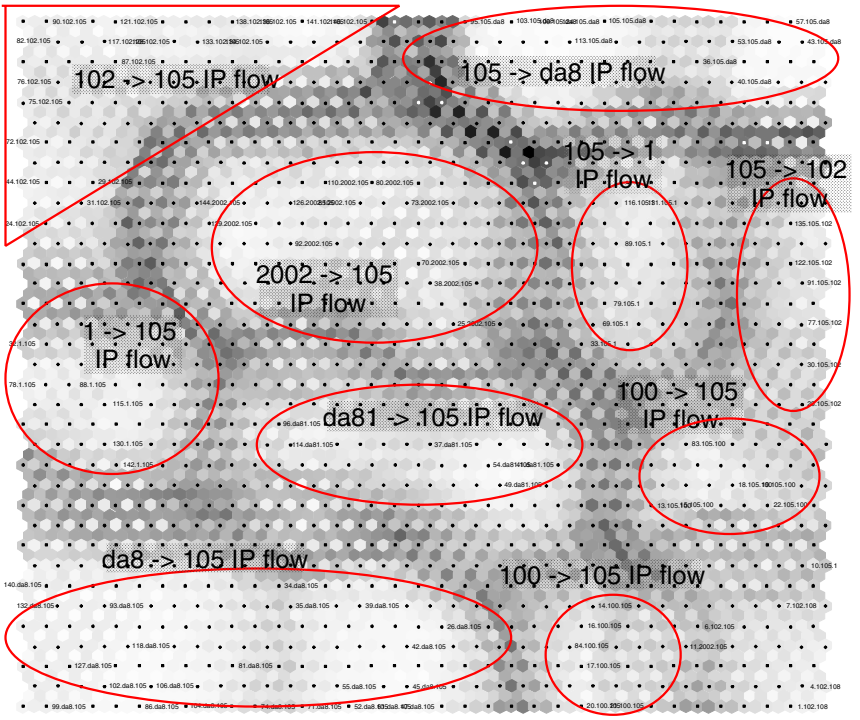


**Fig. 9.** Autonomous IP Flow Identifying

In this paper, we use GHSOM to learn the same normalized traffic data which is also used by SOM. Fig. 8 shows that IP packets can be grouped in different sub-IP flows (shown as different sizes of squares in the figure), according to the source and destination IP addresses. IP packets with the same direction are grouped to the same flow as well as the SOM. Moreover, the IP packets captured within a short interval time are classified to the same sub-IP flows correctly by GHSOM. In this experiment, there is only one session running between two nodes. Therefore, the packets grouped in the same flow or sub-IP flows shown in Fig. 9, are listed in order. Obviously, as shown in Fig. 8, IP packets sent from node da8 to node 105 can be identified to several sub-IP flows by GHSOM. Comparing with Fig. 7, it is clear that the original SOM learning can not identify the sub-IP flows as well as GHSOM. Affected by the incorrect normalization occurred at the start of the time window, there are still a few packets cannot be classified to flows or sub-IP flows correctly in GHSOM either. Thus, our proposal can classify IP packets to IP sub-IP flows correctly at a rate of 92%.

On the other hand, the classification of flows is affected by the applications, network congestion and users. As shown in Fig. 8, the IP sub-IP flow from node da8 to node 105 consist of much more IP packets than other IP sub-IP flows belong to different nodes. These large sub-IP flows are caused by the SIP

authentication failures. Several retries were generated by application automatically. Therefore, GHSOM can detect the behavior of the packets generated by human being or by the programs properly.

Because of the retries caused by the authentication failures, the SIP server, node 105, becomes busy. Thus, the sub-IP flows from other clients, such as the node 2002 and the node 102 are affected. As shown in Fig. 8, the sub-IP flows which consist of the IP packets with the order numbers, from 74 to 82, only have a few IP packets comparing with other sub-IP flows.

## 5 Conclusion and Future Work

This paper suggests to use SOM and GHSOM for identifying the IP packets to IP flows by using the limited information included in IP header. Therefore, comparing with the tools which need to use the payload of an IP packet for network analysis, the user privacy can be protected well and the machine learning cost can be saved by our proposal.

In this paper, the vector dimension is utilized for describing the network topology and a new normalization is suggested for changing the raw network traffic to learning input vectors. The authors uses the SOM for learning the high-dimension vectors and the hierarchical architecture of the GHSOM for detail flow analysis.

Our experiment result shows that the IP packets can be identified correctly through using the new proposal. Furthermore, this paper discovered that a flow can consist of several sub-IP flows. According to the classified flows and sub-IP flows, we can detect abnormal behaviors during analyzing a few captured IP packets. As the future work, the proposal should be evaluated in a large scale network which has more network nodes and more network traffic. Also treating coming input data and classifying IP flows and sub-IP flows in real time are also remaining issues. Using AS number as the characteristic of network topology is also necessary in the normalization. Because of high-dimension vector space, our new suggestion is also supposed for being used to classify the packets used in multicast or multihoming environment. The evaluation test is required.

## References

1. Mori, T., Takine, T., Pan, J., Kawahara, R., Uchida, M., Goto, S.: Identifying Heavy-Hitter Flows from Sampled Flow Statistics. IEICE Transaction 90-B(11), 3061–3072 (2007)
2. sFlow, `http://www.sflow.org/`
3. NetFlow, `http://www.cisco.com`
4. Openflow, `http://www.openflowswitch.org/`
5. Claise, B.: Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of IP Traffic Flow Information, RFC 5401 (January 2008)
6. Sadasivan, G., Brownlee, N., Clasise, B., Quittek, J.: Architecture for IP Flow Information Export, RFC 5470 (March 2009)

7. Kohonen, T.: Self-Organizing Maps. Springer (2000)
8. SOM_PAK, http://www.cis.hut.fi/research/som_lvq_pak.shtml
9. IPv6 Specification, Internet Engineering Task Force, RFC 2460 (December 1998)
10. Rauber, A., Merkl, D., Dittenbach, M.: The growing hierarchical self-organizing map: exploratory analysis of high-dimensional data. IEEE Transactions on Neural Networks 13(6), 1331–1341 (2002)
11. Dittenbach, M., Merki, D., Rauber, A.: The Growing Hierarchical Self-Organizing Map. In: Proc. of the International Joint Conference on Neural Networks (IEEE IJCNN 2000) (July 2000)
12. Palomo, E.J., Domínguez, E., Luque, R.M., Muñoz, J.: A New GHSOM Model Applied to Network Security. In: Kůrková, V., Neruda, R., Koutník, J. (eds.) ICANN 2008, Part I. LNCS, vol. 5163, pp. 680–689. Springer, Heidelberg (2008)