2-6 Development and Evaluation of NIRVANA: Real Network Traffic Visualization System

SUZUKI Koei, ETO Masashi, and INOUE Daisuke

We have developed a real network traffic visualization system named “NIRVANA”. The NIRVANA is based on a visualization engine in the nicter called Atlas, which animates darknet traffic on a world map in a real-time manner. The NIRVANA visualizes livenet traffic, namely traffic in real operating networks, and significantly helps network administrators to promptly grasp comprehensive network status and network troubles such as bottlenecks, configuration errors and irregular communications.

Keywords
Network monitoring, Network administration, Visualization

1 Introduction

The development of virtualization technologies and the penetration of cloud computing are making our network environments more complex every day. Network administrators need to quickly understand traffic conditions to operate those complex networks. The situation requires more sophisticated skills of network administrators, making operational management more challenging.

The Network Security Incident Response Group of the Information Security Research Center has been engaged in the development of the nicter system[1][2] to study how to leverage large-scale network monitoring to speed up the detection of security incidents and rapidly identity their root causes. Multiple visualization systems developed by the project have been instrumental in enabling analysts to understand the status of cyber attacks in a real-time, instinctive manner[3][4].

Many visualization systems provided by the nicter visualize the traffic reaching darknets (i.e. unused IP address blocks). Applying them to real networks (i.e. networks where user terminals, servers, and other devices are connected) can lead to effective support tools enabling network administrators to act more rapidly. This paper describes in detail a real network traffic visualization system entitled NIRVANA, developed based on a global cyber attack visualization system called Atlas.

In this paper, Chapter **2** discusses the characteristics of Atlas, a global cyber attack visualization system. Chapter **3** summarizes the objectives and requirements of NIRVANA, followed by an overview of NIRVANA in Chapter **4** and the performance evaluation and effects to be observed based on actual operations of NIRVANA in Chapter **5**. Finally, Chapter **6** sums up the whole discussion and outlines future challenges.

2 Atlas, a global cyber attack visualization system

2.1 Characteristics of Atlas

The nicter comprises 1) the macro analysis system detecting and analyzing incidents based on large-scale network monitoring and event parsing, 2) the micro analysis system collecting and analyzing malware samples to extract their behaviors, and 3) the correlation analysis system analyzing the results of the first two systems to associate incidents with malware

behaviors.

Atlas, a global cyber attack visualization system is located within the macro analysis system, enabling analysts to visualize attack traffic reaching networks to be monitored in real-time and capture geographical trends of cyber attacks (see Fig. 1). Attack traffic is shown as a packet object (e.g. a 3D image like a circular cone), represented as an animation forming an arc and moving from a source country to a destination country. This allows us to instinctively capture how DDoS attacks are triggered across the world and botnets propagate malware infections (i.e. large-scale scans), leading to subsequent more in-depth analysis.

2.2 Challenges faced in applying Atlas to real networks

Atlas is optimized to monitor darknets and represent the regionality of incoming packets on a world map. Thus, it used to have the following four challenges as it tried to visualize real network traffic.

Challenge #1: Flexible switchover of background images

Atlas was designed to detect the regionality of cyber attacks triggered across the world and was not intended for the frequent switchover of background images. Considering that various network topologies exist in each system and those topologies change frequently in real networks, the application of Atlas to real networks has required the customization of background images and the rewriting of related programs.

Challenge #2: Elimination of noise traffic during failure monitoring

We do not have to verify traffic reaching darknets as authorized or unauthorized; we can construe most packets as unauthorized. On the other hand, most traffic reaching real networks is authorized. As real networks face some failures, network administrators are required to eliminate noise traffic (i.e. traffic not to be monitored) out of a vast volume of traffic. However, Atlas did not have the functionality to narrow down the traffic.

Challenge #3: Representation of traffic flow volume

One of the important missions to be done by network administrators is to investigate network utilization rates by monitoring traffic flow volume and create optimal networks. On the other hand, Atlas represents each packet in real-time images to capture traffic, capable of understanding traffic types in an instinctive manner but not really good at accurately iden-

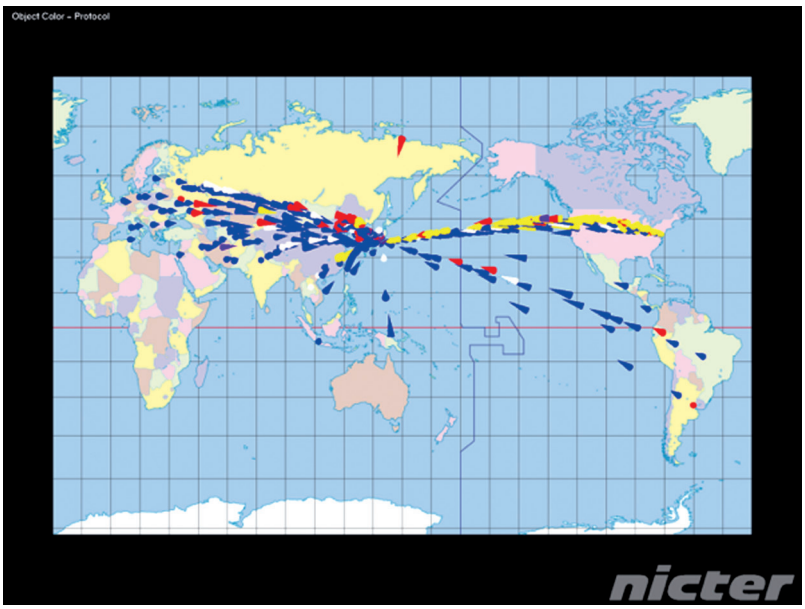

Fig.1 Atlas

tifying their volume.

Challenge #4: Real-time processing of large traffic

Real networks contain much larger traffic volume than darknets. Thus, the limitation of the graphic imaging capability would make it difficult to represent images per packet. This condition has required us to be creative in making the visualization process more lightweight and conduct accurate performance evaluation so that the system could meet the traffic processing performance required by actual operations.

3 Objectives and requirements of NIRVANA

Based on the challenges related to Atlas, a global cyber attack visualization system, discussed in Chapter **2**, this chapter describes the objectives and system requirements of NIRVANA, our traffic visualization system targeted for real networks.

3.1 Objectives of NIRVANA

NIRVANA visualizes real network traffic in real-time to enable us to instantly detect network failures (e.g. congestions and disruptions) and misconfigured devices and to reduce the workload of network administrators. The deployment of NIRVANA will be able to make the network administration by cloud service providers and telecom carriers quicker, more efficient, and dramatically reduce their administration costs.

3.2 System requirements

As we deploy a real network visualization system based on Atlas, the following four system requirements are defined based on the challenges discussed by Section **2.2**.

1) Switching over background images

To resolve challenge #1 (i.e. the lack of the background image switchover feature), NIRVANA offers the background image switchover capability. Network administrators can freely customize background images without rewriting programs to enable the visualization of various network topologies.

2) Narrowing down traffic

To address challenge #2 (i.e. the elimination of noise traffic), traffic filter feature will be deployed. In case of network failure, network administrators can highlight traffic to be monitored or eliminate noise traffic (i.e. traffic not to be monitored) out of a vast volume of traffic by highlighting or narrowing down traffic through the visualization interface.

3) Displaying the detailed information of packets

To expedite the traffic narrowing-down required by challenge #2, network administrators will need to be able to instantly access the detailed information of traffic to be monitored. To display the detailed information of packets on the visualization screen, the capability to view detailed information is provided.

4) Visualization of data flow volume

To address challenge #3 (the representation of traffic flow volume) and challenge #4 (the real-time processing of large traffic), network administrators will need to capture traffic volume (i.e. the number of packets and data volume) and process traffic visualization in a lightweight manner. To meet this goal, features need to be added and processing performance should be enhanced.

4 Deployment

Section **4.1** discusses the overall configuration of the NIRVANA system, followed by Section **4.2** and subsequent sections explaining various NIRVANA features deployed to meet the system requirements as described in Section **3.2**.

4.1 System configuration

- **Default system configuration**

NIRVANA is composed of its three subsystems: 1) the sensor systems collecting traffic from networks to be monitored, 2) the gate system aggregating collected traffic, and 3) the visualization system representing aggregated traffic on a network diagram in a 3D format. Network administrators use the visualization

interface (GUI) operating on the visualization system to understand network conditions. NIRVANA deploys sensors in multiple locations on networks to aggregate network traffic from multiple networks and monitor them.

Many components of NIRVANA (except for the visualization system) share features with the system developed based on the nicter project. Figure 2 shows the typical system configuration of NIRVANA.

- **Only visualization system structure**

The visualization system of NIRVANA can be operated as a standalone system by making it capture its own network interface card (NIC). This feature enables us to use small-size laptop PCs and other devices as the visualization system, creating environments for agile monitoring by directly monitoring the traffic of mirror ports of network devices with possible failures. Figure 3 shows the only visualization system structure of NIRVANA.

4.2 Switching over background images

This section explains how we deploy the system to meet system requirement (1) (i.e. switching over background screens) as mentioned in Section **3.2**.

To enable the visualization of various

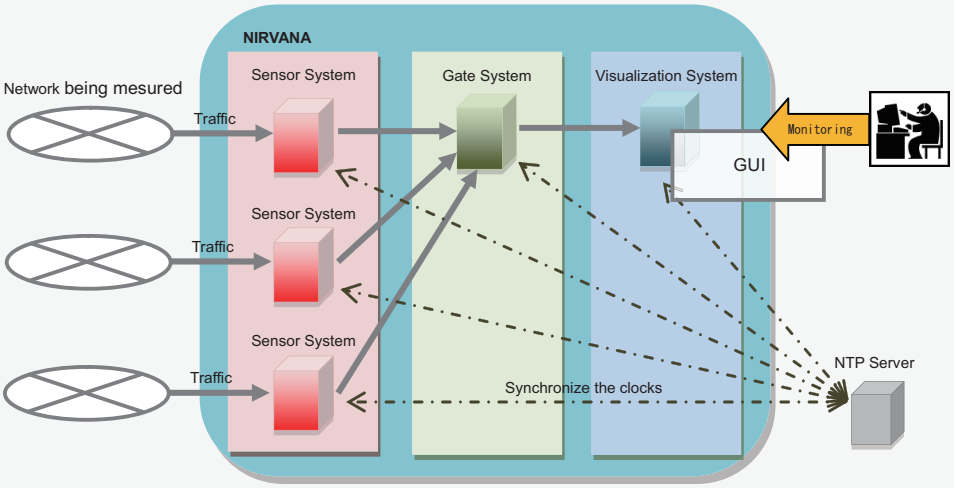

Fig.2 *Basic structure of NIRVANA*

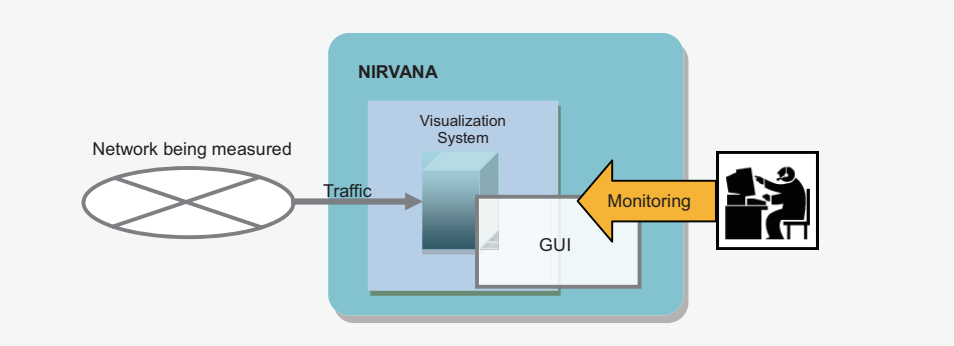

Fig.3 *Only visualization system structure of NIRVANA*

network topologies, we have deployed the NIRVANA system so that the visualization system can automatically load network images created by imaging tools. We have adopted Microsoft Visio 2007 as the imaging tool since it allows us to allocate data (e.g. IP address) as the object of network devices, offers ease of use for drawing images, and is widely used as a general tool.

To visualize traffic on NIRVANA, we need such information as background images of network topologies, the coordinates for the objects of source/destination network devices, and the IP addresses (or IP address ranges) corresponding to the coordinates.

We use Microsoft Visio 2007 to place the objects of network devices and define an IP address (or an IP address range) as a data field of the placed objects. The data field is a label giving a certain value to each object. Thus, defining the data field associates an object to an IP address. The visualization process draws an image corresponding to the traffic between the objects of network devices based on their source/destination IP addresses.

We can also define areas where each network device is located in the data field. For example, we can define an area to specify each sales office or each office floor. The visualization process can color-code or show/hide the traffic for each defined area, enabling us to distinguish traffic based on areas.

Image files created by Visio are output in the form of image files (png format) and coordinate information files (xml format), both of which are loaded onto the visualization interface of NIRVANA as background images and the information corresponding to the coordinates and IP address ranges (network ranges) of objects (see Fig. 4). This allows us to easily customize or manage background images used for traffic visualization and to visualize various network topologies.

Figures 5–7 show the background images created by the Information System Team (current Information System Office) of National Institute of Information and Communications Technology (NICT) to monitor its internal traffic. Figure 5 shows the image to visualize traffic between NICT and various countries. Figure 6 illustrates the image to visualize NICT's internal networks. Figure 7 is the image to visualize the traffic among IP address blocks assigned to NICT. As these example

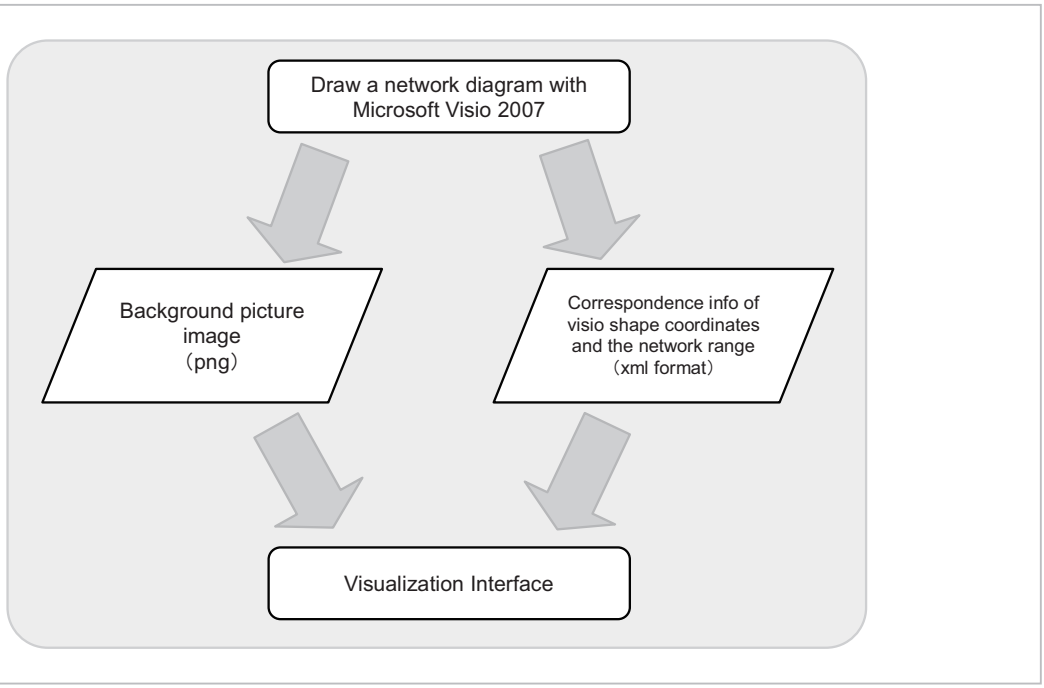

Fig.4 *Data flow of background picture configuration*

Fig.5 *Background picture for visualizing traffic between NICT and countries in the world*

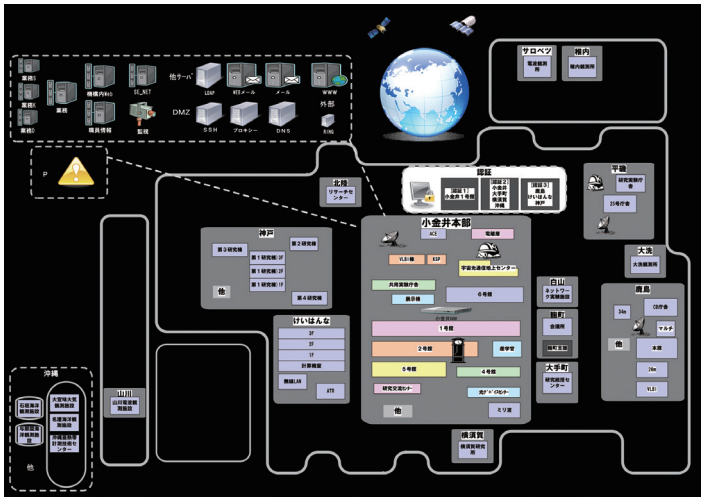

Fig.6 *Background picture for visualizing traffic between NICT branches*

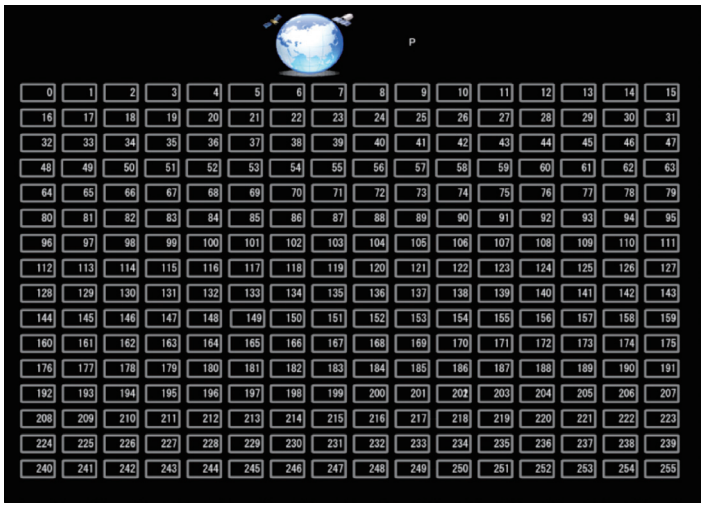

Fig.7 *Background picture for visualizing traffic between IP address block in NICT*

images show, we can visualize the same incoming traffic data from various perspectives (e.g. based on logical configuration and physical placement).

4.3 Narrowing down traffic

This section explains how we deploy the system to meet system requirement (2) (i.e. narrowing down traffic) as mentioned in Section **3.2**. NIRVANA provides the ability to narrow down traffic with the features to color-code packet objects and filtering feature.

4.3.1 Color-coding packet objects

This feature allows us to change the color of packet objects used to draw traffic based on various parameters. This enables us to highlight traffic from multiple perspectives. The following show the parameter types used for color-coding.

- Protocol

We can define a color of packet objects for each protocol type. We can use the following protocol/flag types: TCP/SYN, TCP/SYN-ACK, TCP/ACK, TCP/PUSH, TCP/RST, TCP/FIN, TCP/OTHER (other flags), UDP and ICMP.

- Area

We can define a color of packet objects for each destination area. Area means regional information we can define for each background image object as explained in Section **4.2**.

- Sensor ID

We can define a color of packet objects for each sensor ID. Sensor IDs are IDs uniquely allocated to each sensor system. Each sensor ID corresponds to the point where traffic is collected.

- IP address

We can define colors of packet objects based on the combination of source and destination IP addresses (i.e. IP address filters). IP address filters are created by the IP address filtering feature (see Section ***4.3.2***).

- Port number

We can define colors of packet objects based on the combination of source and destination ports (i.e. port number filters). Port number filters are created by the port number filtering feature (see Section ***4.3.2***).

4.3.2 Filtering

We can define filters (i.e. color-coding of packet objects, show/hide setup of images, and sampling) for each item to be filtered so that we can filter traffic displayed on screens. The following shows items to be filtered, followed by filter setups used for each item to be filtered.

The following are items to be filtered. Five items can be filtered.

- Protocol filter (Fig. 8)

The protocol filter allows us to set up filtering for each protocol type. As protocol types, we have adopted TCP (mainly used for real networks) and its flags, UDP, and ICMP. Table 1 shows the types of TCP flags and their filtering criteria. Figure 8 shows how we color-code traffic for each protocol type based on the protocol filter to show the protocols contained in the traffic.

- IP address filter (Fig. 9)

The IP address filter allows us to set up filtering for each combination of source and destination IP addresses. As we define the combination of source and destination IP addresses, we can select specific IP addresses or "ANY" (meaning all IP addresses). Figure 9 only shows the traffic between specific source and destination IP addresses based on the IP address filter.

- Port number filter (Fig. 10)

The port number filter allows us to set up

Table 1 *Criteria for determining protocol types of TCP packet*

Protocol Type	Criteria
TCP_SYN	Only the TCP SYN flag is on
TCP_SYN_ACK	Only the TCP SYN and ACK flags are on
TCP_ACK	Only the TCP ACK flag is on
TCP_FIN	The TCP FIN flag is on
TCP_RST	The TCP RST flag is on
TCP_PUSH	The TCP PSH flag is on
TCP_OTHER	Other TCP packets

filtering for each combination of source and destination port numbers. As we define the combination of source and destination port numbers, we can select specific port numbers or "ANY" (meaning all port numbers). Figure 10 only shows the traffic transmitted to the

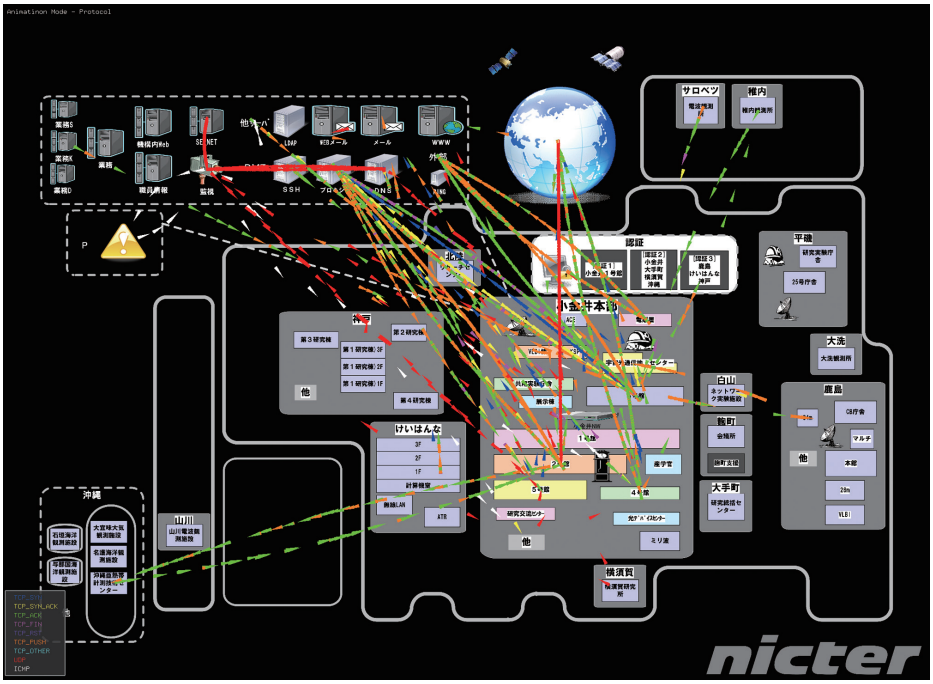

Fig.8 *Example of protocol filter*

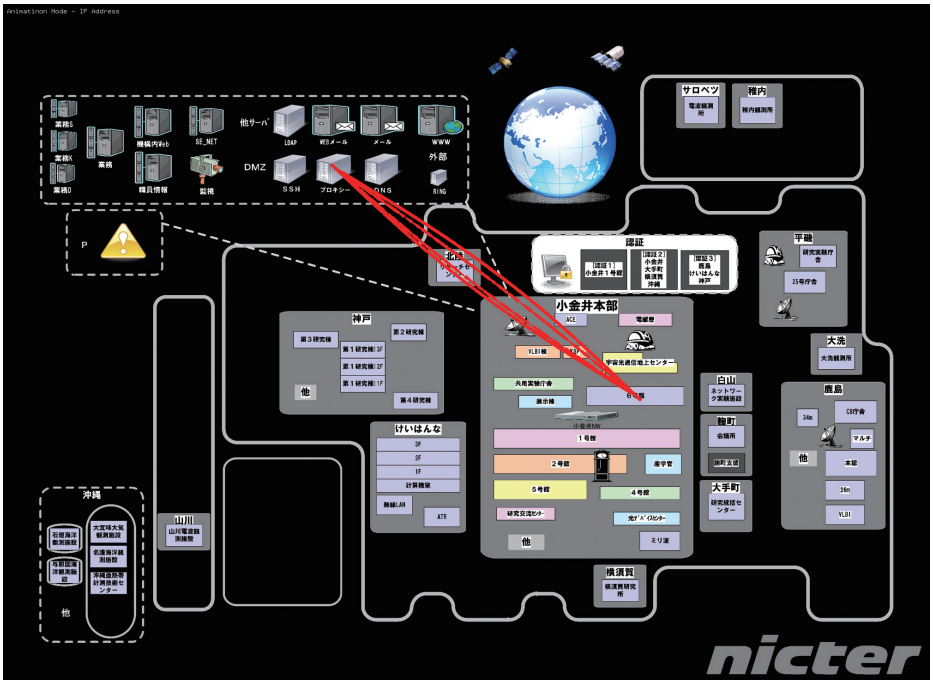

Fig.9 *Example of IP address filter*

port number 80.

- Sensor ID filter (Fig. 11)
  The sensor ID filter allows us to set up filtering based on sensor IDs, which are uniquely allocated to each sensor system. Figure 11 highlights the traffic collected by a sensor sys-

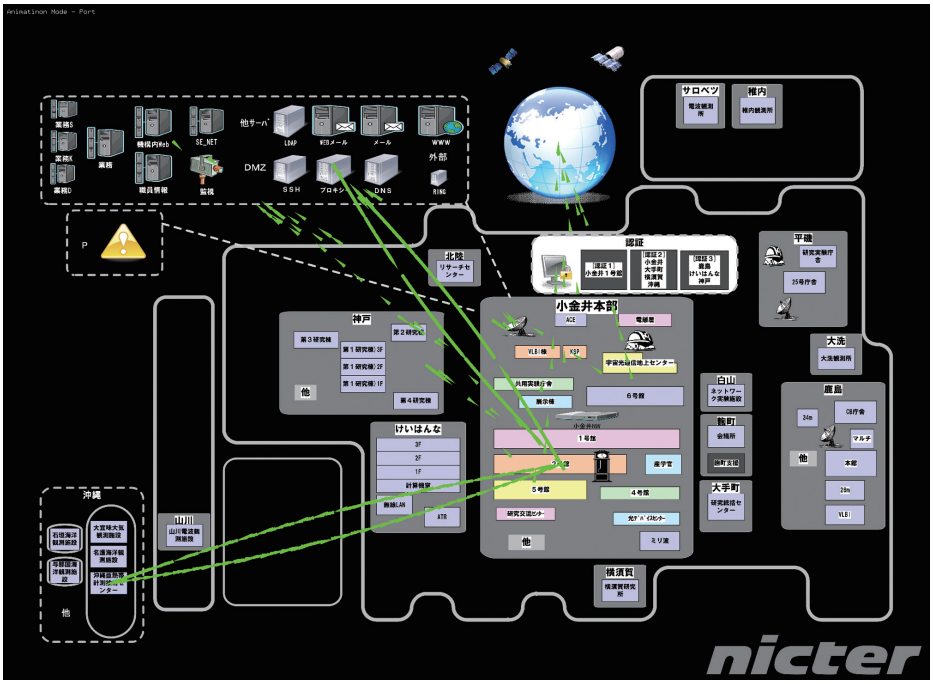

Fig.10 *Example of port number filter*

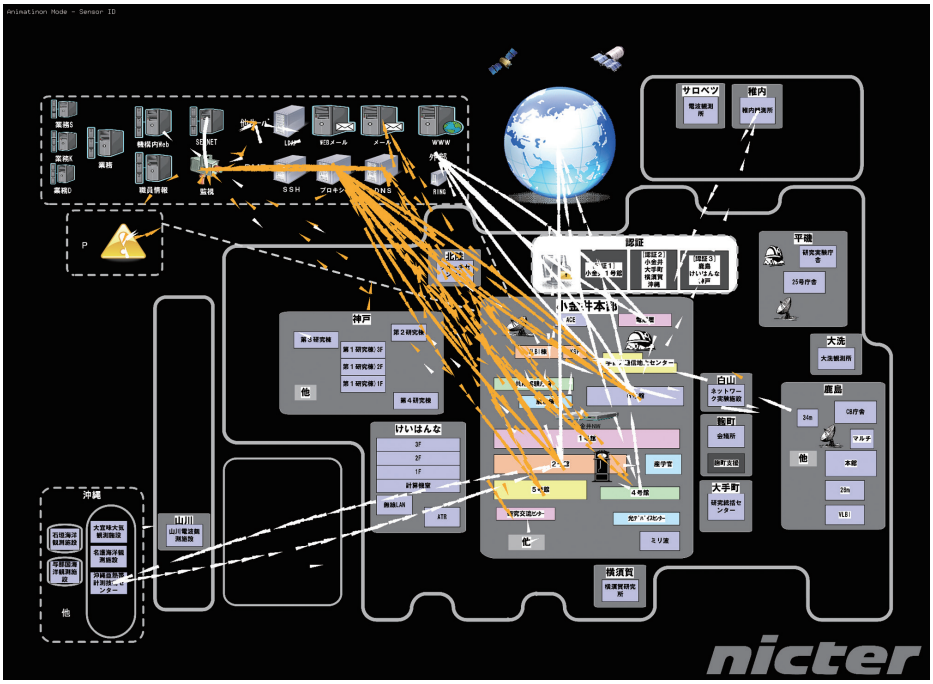

Fig.11 *Example of sensor ID filter*

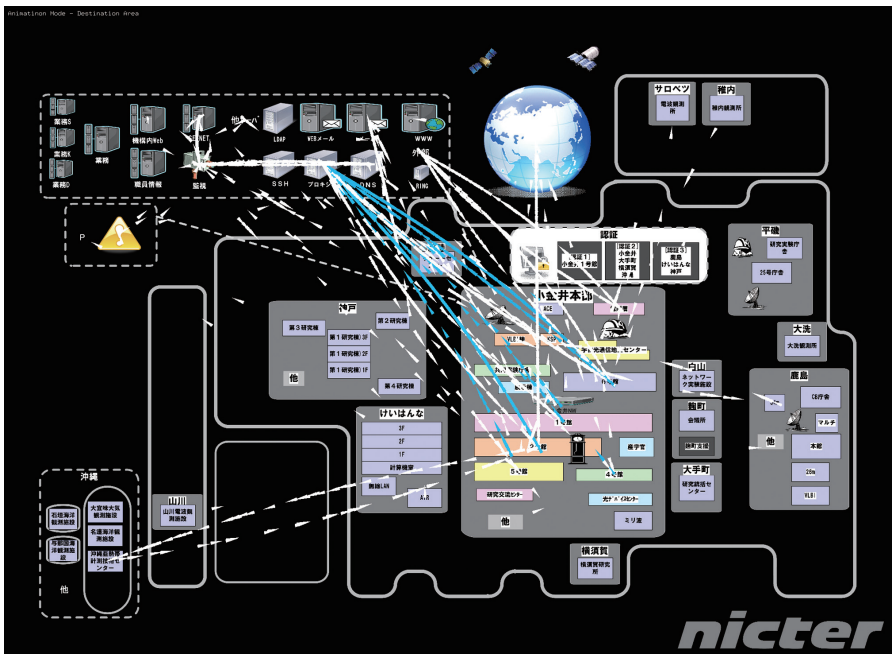

Fig.12 *Example of area filter*

tem with a specific sensor ID based on the sensor ID filter.

- Area filter (Fig. 12)

We can set up filtering for each destination area. Figure 12 only highlights the access toward the server defined as a specific area based on the area filter.

We have defined the following three filtering setups commonly used for each item to be filtered.

- Color-coding

We can define the color of packet objects for each item to be filtered so that we can highlight certain traffic.

- Show/hide setup for images

We can hide certain traffic out of the total traffic so that we can narrow down the traffic.

- Sampling

We can pick up certain traffic out of the total traffic (i.e. sampling process) so that we can make the remaining traffic less visible.

4.4 Displaying the detailed information of packets

This section explains how we deploy the system to meet system requirement (3) (i.e. displaying the detailed information of packets) as mentioned in Section **3.2**.

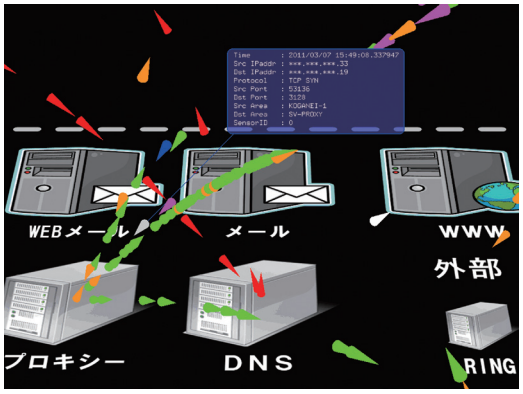

Fig.13 *Detailed packet information display window*

NIRVANA not only simply visualizes traffic but directly operates visualized 3D objects. This has enabled us to drill down information much more rapidly than the existing log-based network management. Figure 13 shows how clicking on a packet object with a mouse displays the window to show its detailed information. The window includes the timestamp

of the packet, source/destination IP addresses, a protocol, source/destination port numbers, source/destination areas, and a sensor ID.

4.5 Visualization of data flow volume

This section explains how we deploy the system to meet system requirement (4) (i.e. the visualization of data flow volume) as mentioned in Section **3.2**.

This feature has been enabled by the visualization of data flow volume between network segments (see Section ***4.5.1***) and data transmission volume of a network segment (see Section ***4.5.2***).

4.5.1 Data flow volume between network segments

To capture the data flow volume between network segments, we use an arc to represent the data flow volume (i.e. the number of packets and data volume) between shapes corresponding to network segments (see Fig. 14). The arc representing data flow volume stretches from a shape corresponding to a source IP address to a shape corresponding to a destination IP address, with its color, height, width, transparency, and gradation representing its data flow volume. A circular cone object is placed at the peak of the arc to show the direction of traffic. In addition, data flow volume counts are shown at the peak of the arc. Table 2 shows the types of data flow volume counts represented.

The following explain the items representing data flow volume.

- Color of an arc (Fig. 15)
  The color of an arc is decided based on the

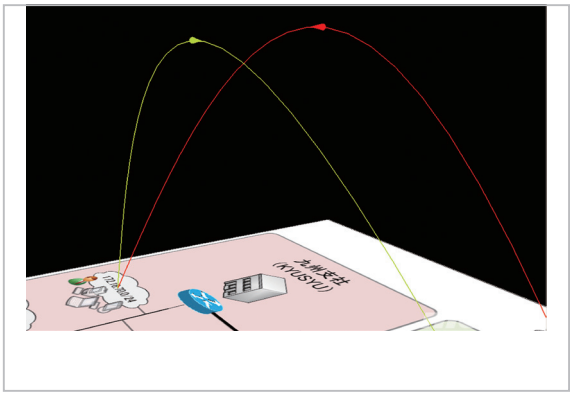

Fig.14 *The arcs representing data flow*

Fig.15 *Color legend of the arcs*

Table 2 *Types of data flow counter*

Type of data flow counter	Content of display
Packet count	Show the integrated value of the number of packets count between network objects.
Packet count rate	Show the packet flow rate between network objects in percentage. The value is displayed to 2 decimal places.
Data count	Show the integrated value(*) of the number of data count between network objects. The value is displayed in Mbit, and to 3 decimal places.
Data count rate	Show the data flow rate(*) between network objects in percentage. The value is displayed to 2 decimal places.

*The value is calculated by adding the length of IP packet.

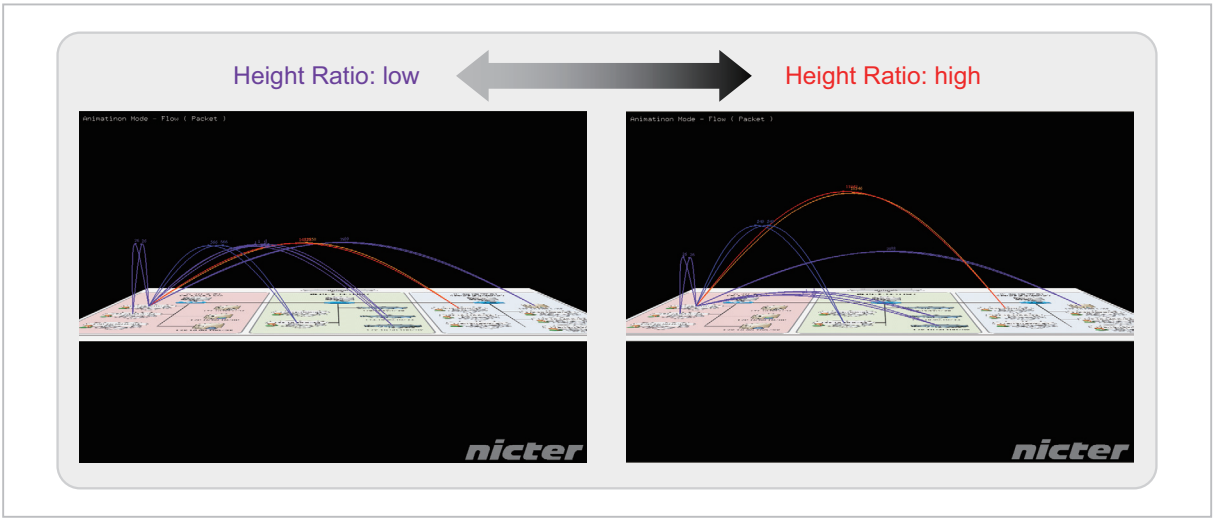

Fig.16 *Height configuration of the arcs*

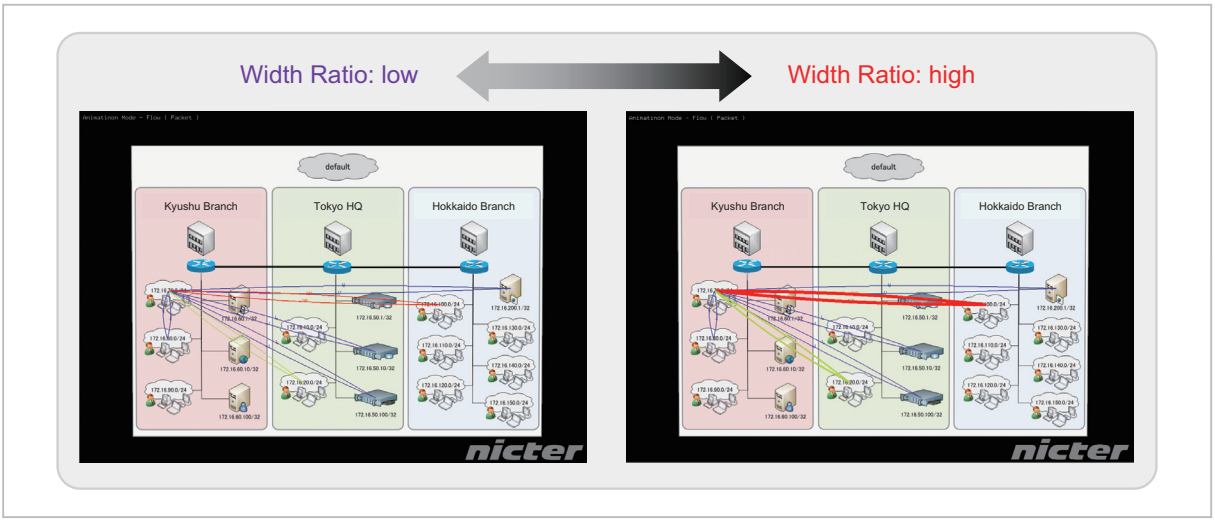

Fig.17 *Width configuration of the arcs*

related data flow volume, with an arc containing the largest data flow volume shown in red. Other colors are relatively decided based on the maximum data volume counters. The colors are represented in the gradation from red to purple based on the counter value, with red corresponding to the largest counter value.

- Height of an arc (Fig. 16)

The height of an arc is decided based on the related data flow volume, with an arc containing more data flow volume placed higher. To make it easier to detect spots with large data flow volume, the height ratio can be set up by users. The smaller the height ratio is, the smaller the height gap among all arcs will be. The larger the height ratio is, the larger the gap is between an arc with a larger counter value and an arc with a smaller counter value. This feature shows an arc with a larger counter value more clearly.

- Width of an arc (Fig. 17)

The width of an arc is decided based on the related data flow volume, with an arc containing more data flow represented by a wider arc. To make it easier to detect spots with large data volume, the width ratio can be set up by users. The smaller the width ratio is, the smaller the width gap among all arcs will be. The higher the width ratio is, the larger the gap is between an arc with a larger counter value and an arc

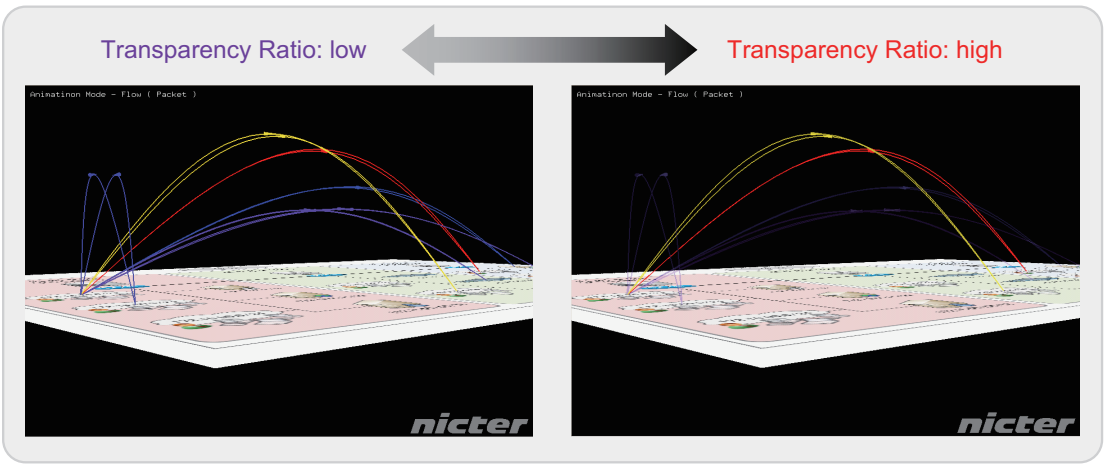

Fig.18 *Transparency configuration of the arcs*

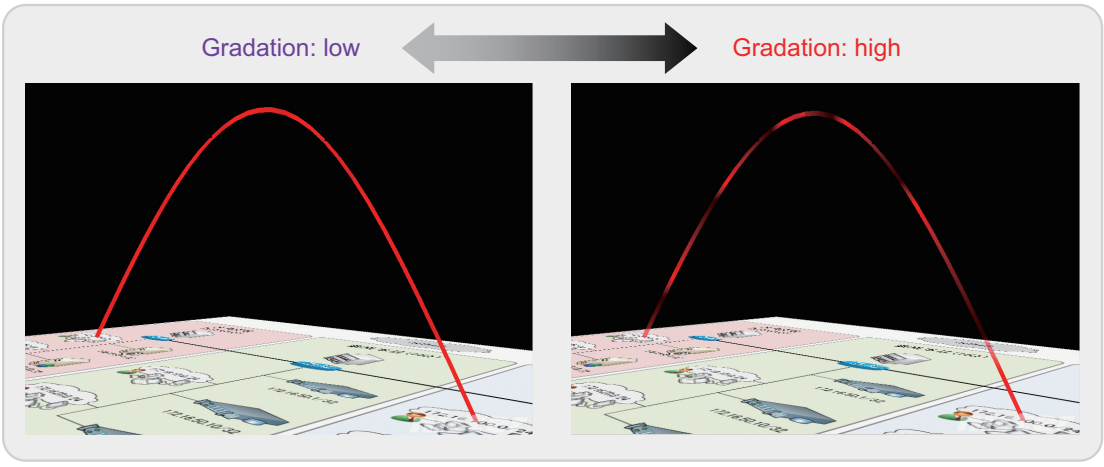

Fig.19 *Gradation configuration of the arcs*

with a smaller counter value. This feature shows an arc with a larger counter value more clearly.

- Transparency of an arc (Fig. 18)

The transparency of an arc is decided based on the related data flow volume, with an arc containing more data flow represented by a darker arc. To make it easier to detect spots with large data volume, the transparency ratio can be set up by users. The smaller the transparency ratio is, the smaller the transparency gap among all arcs will be. The higher the transparency ratio is, the more transparent an arc with a smaller counter value will be. This feature shows an arc with a larger counter value more clearly.

- Gradation of an arc (Fig. 19)

The gradation of an arc is represented so that an arc currently containing larger data flow volume is shown as an arc with larger traffic. No gradation is applied to the arc with no traffic counted.

4.5.2 Data transmission volume of a network segment

An image representing a network segment shows the data transmission volume (i.e. # of packets and data volume) for the network corresponding to its IP address range in the form of bar charts. Outgoing traffic is shown in a blue bar and incoming traffic is shown in a red

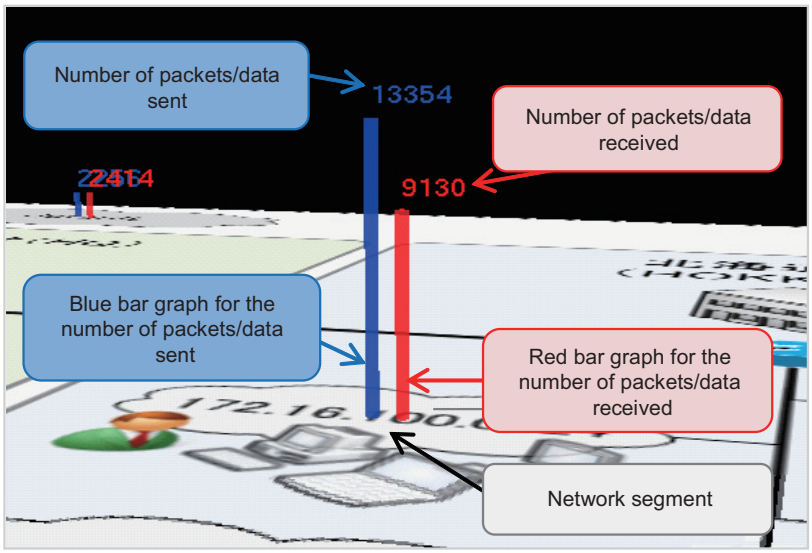

Fig.20 *Data count graphs*

Table 3 *Max traffic volume of each monitoring point in NICT*

Monitoring point	pps	bps
Point 1	26,378	427,534,624
Point 2	99	1,604,592
Point 3	4,642	75,237,536
Point 4	79	1,280,432
Point 5	4,117	1,280,432
Point 6	50	810,400
Point 7	3,407	55,220,656
Point 8	312	5,056,896
Point 9	42,712	692,276,096
Point 10	139	2,252,912
Point 11	24,945	404,308,560
Point 12	4,705	76,258,640
Point 13	58,599	949,772,592
Total	170,184	2,692,894,368

Table 4 *Performance requirements of each system*

System	pps	bps
Sensor	60,000 pps	950 Mbps
Gate	170,184 pps	205 Mbps
Visualization	170,184 pps	164 Mbps

bar. The peak of each bar shows the counter values. Table 2 describes the counter values displayed. The height of each bar shows the highest data volume detected. Other graphs are relatively adjusted based on the size of the highest counter value. Figure 20 illustrates how bar charts are represented.

5 Evaluation

5.1 Performance evaluation

NIRVANA captures and visualizes real network traffic. Thus, it is required to process a large volume of traffic in real-time. To evaluate whether the visualization system can process all the captured traffic, we have evaluated the processing performance of NIRVANA based on its operations on NICT's networks.

5.1.1 NIRVANA's system performance requirements for operations on NICT's networks

NICT operates Class B (16-bit netmask) networks internally. Monitoring all networks of NICT requires the monitoring of traffic at 13 points throughout its configuration. Table 3 shows the results of the largest number of packets detected per second (pps) and the maximum data volume per second (bps) for each point. Based on the monitoring results, we

have defined the performance requirements of each system as below, with Table 4 summing up the values of performance requirements.

- Performance requirements of the sensor systems

The sensor system should be able to regularly process traffic at 60,000 pps and 950 Mbps based on the results of Point 13 with the largest maximum values.

- Performance requirements of the gate system

The performance requirement of the gate system should correspond to the total of the maximum data flow volume (170,184 pps) from all collection points so that it can aggregate traffic detected by all sensor systems. The maximum data flow volume between the sensor systems and the gate system is set for 205 Mbps since the average data length of traffic between the sensor systems and the gate system is 150 bytes.

- Performance requirements of the visualization system

The visualization system needs to process data aggregated by the gate system, thus requiring processing performance of 170,184 pps just like the gate system. The maximum data flow volume between the gate system and visualization system is set for 164 Mbps since the average data length of traffic between the gate system and the visualization system is 120 bytes.

5.1.2 Evaluation environment

The operating systems and languages used are the following.

- Sensor system and gate system
  - OS: FreeBSD 7.2[7]
  - Language: C/C++
  - Library: libpcap-0.9 series[8] (packet capture)
- Visualization system
  - OS: CentOS 5.4[9]
  - Language: C/C++
  - Library: gtk-2.0[10] GUI), gtkglext-1.2 series[11], freeglut-2.4 series[12] (graphic)

The following show the hardware environment used for the evaluation.

- Traffic generating device
  Spirent TestCenter
- Sensor system and gate system
  - Model: DELL PowerEdge R210
  - CPU: Intel Xeon X3450 2.66GHz
  - Memory: 2GB
  - NIC: On-board Broadcom NetXtreme II BCM5716 1000Base-T
- Visualization system
  - CPU: Intel Core2Quad Q9650 3.00GHz
  - Memory: DDR2-667 8GB
  - NIC: Intel 82573 10/100/1000Base-T
  - Video card: nVidia GeForce GTX285 1024MB

5.1.3 Evaluation method

We have evaluated the sensor systems by generating packets by the traffic generating device and transmitting the traffic to the sensor systems with the highest possible load. We have used each byte length from 64 bytes to 1,500 bytes and IMIX packets for experiments. IMIX stands for Internet MIX, meaning a traffic mix containing typical types of traffic detected on the Internet. The traffic we have evaluated is composed of 64-byte packets (approx. 58.33%), 570-byte packets (approx. 33.33%), and 1,518-byte packets (approx. 8.33%).

In evaluating the gate system, the traffic generating device generated the data traffic between the sensor systems and the gate system and the traffic was transmitted to the gate system with the highest possible load.

In evaluating the visualization system, the traffic generating device generated the data traffic between the gate system and the visualization system and the traffic was transmitted to the visualization system with the highest possible load.

5.1.4 Evaluation results and discussion

Table 5 shows the evaluation results. We have confirmed that the sensor systems have met the performance requirement of 60,000 pps for each packet length. However, a further increase of the traffic volume will necessitate the deployment of high-speed packet capture technologies (e.g. Zero copy BPF[5] and ring-

Table 5 *Performance test results of each system*

System	Test Condition	Result
Sensor	64 byte	179,191 pps
	570 byte	187,864 pps
	1,518 byte	82,240 pps
	IMIX	187,645 pps
Gate	-	191,597 pps
Visualization	Flow	180,000 pps
	Packet Draw duration 1 sec	12,000 pps
	Packet Draw duration 3 sec	4,000 pps

map[6]) or the deployment of additional sensors to enable the load-balancing among sensors.

We have also confirmed that the gate system has met the performance requirement with its performance of 191,597 pps. In case the visualization system visualizes traffic per packet, we have found that image drawing deration for packet objects and the packet processing performance are inversely proportionate to each other. This is because more 3D objects are shown at the same time on the visualization screen as it takes longer to draw packet object images. A bottleneck occurs in NIRVANA with the video feature of the visualization system. Thus, we have deployed the sampling feature on the visualization system so that it can sample a certain ratio of packet objects and continue to display a certain ratio of packets even if the number of incoming packets exceeds the maximum packet processing performance. On the other hand, we have confirmed that NIRVANA can dramatically increase its packet processing performance when it visualizes data flow volume.

5.2 Effects to be observed based on actual operations of NIRVANA

NIRVANA is currently operated by the Information System Team (current Information System Office) of NICT to support the operations of NICT's internal networks. It allows us to understand current traffic status, confirm the behaviors of network devices when they are being replaced, or detect unauthorized traffic triggered by misconfigured devices. The mechanism enables network administrators to detect unauthorized traffic caused by misconfigured PCs and give instructions to PC users, proving that NIRVANA is a valid operation management tool.

However, NIRVANA is a tool dedicated to the real-time visualization of traffic information and the obtained information is valid only temporarily. In case unauthorized traffic is detected as explained above, network administrators cannot access past information in a simple manner. To enable the traceability of traffic, NIRVANA has to work with the traffic accumulation system as provided by the nicter so that it can reproduce past information.

6 Conclusion

This paper gives an overview of NIRVANA, a real network traffic visualization system based on the global cyber attack visualization system (nicter). It also summarizes system requirements of NIRVANA, with the description of multiple deployed features. Furthermore, the performance evaluation results of the system are also provided.

To bring the benefits of the nicter project back to the society, we will actively promote the technological transfer of NIRVANA going forward. Making NIRVANA a more practical system will require us to study various needs and challenges through actual operations of the system and improve its features on a continual basis.

References

1 Koji Nakao, Daisuke Inoue, Masashi Eto, and Katsunari Yoshioka, "Practical Correlation Analysis between Scan and Malware Profiles against Zero-Day Attacks based on Darknet Monitoring," IEICE Trans. Information and Systems, Vol. E92-D, No. 5, pp. 787–798, 2009.

2 Daisuke Inoue, Masashi Eto, Katsunari Yoshioka, Shunsuke Baba, Kazuya Suzuki, Junji Nakazato, Kazuhiro Ohtaka, and Koji Nakao, "nicter: An Incident Analysis System toward Binding Network Monitoring with Malware Analysis," WOMBAT Workshop on Information Security Threats Data Collection and Sharing (WISTDCS 2008), pp. 58–66, Apr. 2008.

3 Koji Nakao, Fumiko Matsumoto, Daisuke Inoue, Shunsuke Baba, Kazuya Suzuki, Masashi Eto, Katsunari Yoshioka, Kenji Rikitake, and Yoshiaki Hori, "Visualization Technologies of nicter Incident Analysis System," IEICE Technical Report, Vol. 106, No. 176, ISEC2006-51, pp. 83–89, July 2006. (in Japanese)

4 Kazuya Suzuki, Shunsuke Baba, Hidehiko Wada, Koji Nakao, Hiroki Takakura, and Yasuo Okabe "Development and Evaluation of a Traffic Visualization System to Support Swift Trouble Shooting," IEICE Trans. Commu. Vol. J92-B No. 10, Oct. 2009.

5 http://www.seccuris.com/documents/whitepapers/20070517-devsummit-zerocopybpf.pdf

6 http://code.google.com/p/ringmap/

7 http://freebsd.org

8 http://www.tcpdump.org

9 http://centos.org

10 http://www.gtk.org

11 http://gtkglext.sourceforge.net/

12 http://freeglut.sourceforge.net/

(Accepted June 15, 2011)

SUZUKI Koei

Technical Expert, Cybersecurity Laboratory, Network Security Research Institute

System Engineer, Network Engineer, Programmer

***ETO Masashi,** Ph.D.*

Senior Researcher, Cybersecurity Laboratory, Network Security Research Institute

Network Security, Malware Analysis, Network Operation

***INOUE Daisuke,** Ph.D.*

Director, Cybersecurity Laboratory, Network Security Research Institute

Network Security, Information Security