Implementation of flexible network system with the grouping method by using VNIC and unique ID

Yuta Watanabe
Tokyo University of Science
Department of Information Sciences
Faculty of Science and Technology
Yamazaki 2641, Noda-shi, Chiba, 278-8510 Japan

Satoshi Kodama
Tokyo University of Science
Department of Information Sciences
Faculty of Science and Technology
Yamazaki 2641, Noda-shi, Chiba, 278-8510 Japan

***Abstract*—Local area network is separated into some groups to ensure the security in company, university, and so on. This is generally implemented by using the function of a networking device. However, it is not easy for this method to change the network layout. In addition, there is a problem with compatibility of the device. This paper proposes a network system which has functions to group clients and to conceal communication data by using virtual NIC (VNIC) and hardware ID. This system is implemented by the computer software. The implementation by the software is able to solve the problems of the conventional method. We verify the effectiveness of this system through the result from experiments with this method.**

I. Introduction

Nowadays most of the computers communicate with each other over a network. In other words, the network is connected by a lot of computers. This can be cause of network congestion and information leakage. We need to take measures against these problems. As a countermeasure, the network is separated into some groups in company, university, and so on. This is implemented by the technology called Virtual Local Area Network (VLAN). This technology constructs independent networks by creating multiple logical LAN segments in a physical switch. The networking devices are assigned to a logical LAN segment. In addition, we configure the permit among the LAN segments. Thus, the network traffic and communication permission are restricted. Although VLAN is very convenient technology, it has some problems. Some of the problems occur because VLAN is the function of the hardware.

This paper proposes a method for separating computers into some groups by the computer software. The implementation by the software is able to solve the problems of VLAN.

The rest of the paper is organized as follows. In section 2, we explain the conventional technology and our proposal. In section 3, we give the overview of our system. In section 4, we describe the details of our system. Finally, we show the experimental results in section 5.

II. Problems of conventional technology and our proposal

We explain the types and problems of VLAN in this section.

There are several types of VLAN. For example,

1) Assigning a logical ethernet segment to each port of a switch (called a Port-Based VLAN)
2) Inserting VLAN tag into ethernet frame and checking it in a switch (called a Tagged VLAN)
3) In addition to the feature of Port-Based VLAN, existing the port which enables to communicate all VLAN. (called a Multiple VLAN)

Those methods have some problems. The first method takes much time to change the network layout. The next method makes the packet size larger. Furthermore, each of the network devices needs to be compatible with the Tagged VLAN. The last one is based on the first one. Therefore, it makes a mess when the network layout is changed.

In addition, there is technology called VPN in order to enhanced the security. This technology makes tunnels among each clients and communicates encrypted data through the ones. However it uses VPN protocol, thereby communicating with the only inside of tunnels.

This paper proposes a method for solving these problems. The method has the functions described below.

- A client can communicate with other ones inside the same group.
- A client can communicate with other ones which belong to the permitted different groups.
- A client can communicate with conventional devices such as routers and printers.
- Data among communication path is concealed.

We use the ID based on the information of each clients in order to implement these functions. Moreover, a third person is not able to connect the network easily since the communication data is rewritten by using this ID. The implementation by the software can reduce the costs.

III. System overview

We give the overview of the system. We construct the environment such as Fig.1.

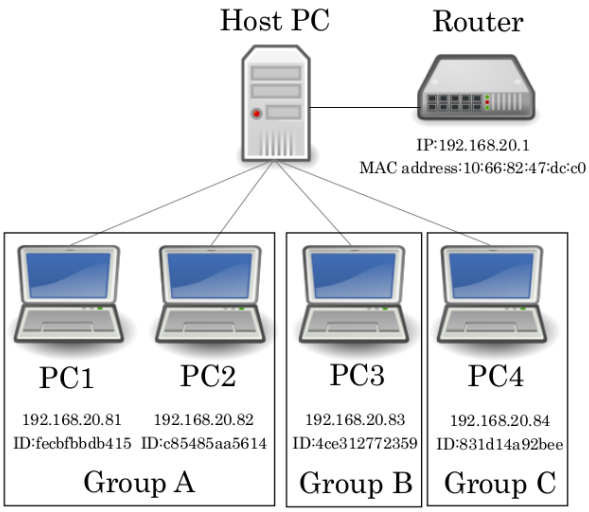

Fig.1 System layout

This is consisted of the 5 PCs (4 clients and 1 host) and general router. The system is built by executing our program on these devices except the router. The program calculates a unique ID. This ID is used to identify each of the clients. The unique ID is registered with Host PC beforehand. Host PC separates clients into some groups by using the unique ID. The network layout is able to be changed easily by the configuration of Host PC. In addition, the communication data among clients and Host PC is rewritten by the unique ID. However, the data is not rewritten if the communication partner is a conventional device. The data is rewritten between the virtual NIC (VNIC) of our own making and physical NIC. The client is able to communicate with other one without being aware of the configuration.

IV. Feature Details

A. Grouping clients

We register the unique ID of each of the clients with Host PC. This ID is used to identify each of the ones and separate the ones into some groups. In case of the conventional device such as the router and printer, the MAC address is registered with Host PC. Moreover, we configure permission among groups and devices. Fig.2 shows an example of the configuration.

B. Concealing communication data among clients and communicating with conventional devices.

1) Implementation of VNIC

We use VNIC of our own making in order to rewrite communication data. This VNIC is a virtual network device driver on the CentOS 6.4. We create a bridge connection between the VNIC and physical NIC by using our program. The bridge connection is shown in the following Fig.3. When executes the packet exchange between these NICs, the program rewrites the packet. This program rewrites the data field of ethernet frame. This data size doesn't change since the program exchange a byte for another byte. The communication data is not rewritten if the communication partner is a conventional device. As a result, the clients are able to communicate with a conventional device. In addition, the network system is able to prevent the lower of transmission speed.

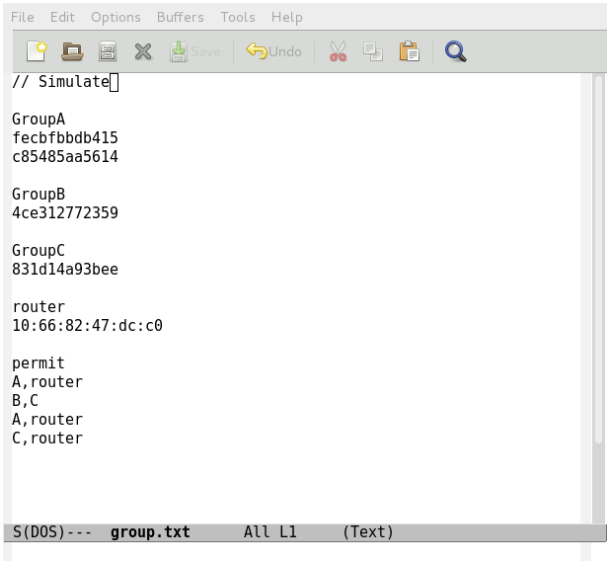

Fig.2 Example grouping configuration

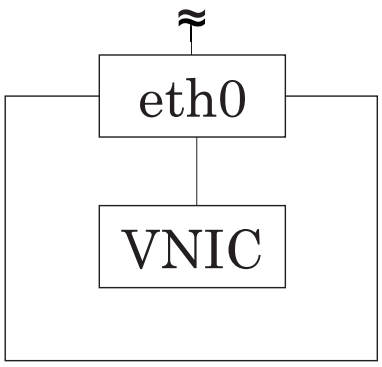

Fig.3 Bridge connection between VNIC and NIC

2) Renewal and synchronization of the random number

In the bridge connection, the communication data is rewritten by a generated value based on a random number and unique ID. If the generated value is not changed for a long time, the system has the risk of wiretapping by a third person. Therefore, the system conceals the data by changing the random number periodically. The client PC and Host PC need to be made the synchronization process. All clients and Host PC follow the procedures described below.

1) Host PC generates a random number and sends it to all clients. The number is stored in an ARP packet. From this point forward, the transmitting buffer stocks the received data until the synchronization process is finished.
2) Each of the clients retrieves the number from the ARP packet and calculates a hash value by this number and its unique ID. This hash value is defined as the new unique ID. The registered unique ID is defined as the original unique ID.
3) Each of the clients sends the new unique ID to all other clients. This new unique ID is stored in Gratuitous ARP packet.
4) Host PC checks that the received new unique ID is the same as a hash value calculated by the original unique ID of the client and the random number. If two values accord, the renewal of the

random number is success. The new unique ID is registered with Host PC. If the transmitting buffer has the data, this data is sent.

5) When receives the packet, each of the clients updates their ARP table.

The system executes this procedures periodically.

3) Process of rewriting data in the client PC

We register the unique ID of a client with Host PC. This unique ID is a hash value calculated by the HDD serial number, the OS version of the client PC and so on. The client follows the procedures described below.

1) The client calculates a hash value by the new unique ID.
2) In the data field of ethernet frame, the program adds this hash value to each of the bytes.
3) The client sends the rewritten packet to Host PC.

When the client receives the packet, the program subtracts the hash value from each of the bytes.

4) Process of rewriting data in the Host PC

We register the unique ID of clients and the MAC address of conventional devices with Host PC. We also register the group of each client and the permission of communication among groups. In addition, Host PC generates a random number and sends it to all clients.

1) When receives a packet, Host PC confirms sender. Host PC executes the process of a) or b) by the sender.
   a) In the case of a client PC. The program calculates a hash value by the new unique ID of sender. In the data field of ethernet frame, the program subtracts this hash value from each of the bytes.
   b) In the case of a conventional device, the program doesn't rewrite the packet.
2) Host PC confirms the groups of the sender and receiver.
3) Host PC checks the permission between the two groups. Host PC executes the process of a) or b) by the permission between two groups.
   a) In the case of permission, Host PC executes the process of i) or ii) by the receiver.
      i) If the receiver is a client PC, the program calculates a hash values by the new unique ID of the receiver. In the data field of ethernet frame, the program adds this hash value to each of the bytes.
      ii) If the receiver is a conventional device, the program doesn't rewrite the packet.
   b) In the case of no permission, the packets are discarded.
4) The Host PC sends the packet to receiver except 3)-b).

A conventional device is unable to recognize the rewritten packet. Therefore, Host PC converts this packet in order to communicate with a conventional device. In addition, this system prevents a third person from spoofing and wiretapping between a client and Host PC.

V. Verification result

We test operation using Fig.1 system and Fig.2 configuration. We show networking devices in PC1. PC1 load the VNIC driver of our own making. We name the driver "vn_our_making". We are able to confirm that the VNIC of our own making is functioning at Fig.4.

```
ファイル(F)  編集(E)  表示(V)  検索(S)  端末(T)  ヘルプ(H)
[root@localhost hostBridge]# ifconfig
enp0s4f1u1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.20.101  netmask 255.255.255.0  broadcast 192.168.20.255
        ether b0:c7:45:5f:61:38  txqueuelen 1000  (Ethernet)
        RX packets 64384  bytes 43319563 (41.3 MiB)
        RX errors 0  dropped 9198  overruns 0  frame 0
        TX packets 0  bytes 3695858 (3.5 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        loop  txqueuelen 0  (Local Loopback)
        RX packets 112  bytes 8304 (8.1 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 112  bytes 8304 (8.1 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

vn_our_making: flags=4291<UP,BROADCAST,RUNNING,NOARP,MULTICAST>  mtu 1500
        inet 192.168.20.200  netmask 255.255.255.0  broadcast 192.168.20.255
        ether 66:55:44:33:22:11  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 11  bytes 2078 (2.0 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

[root@localhost hostBridge]#
```

Fig.4 The VNIC of our making

References

[1] Mitsuyuki Komata, router jisaku de wakaru packet no nagare(Learn to understand the flow of a packet through the making of a router), Gijutsu-Hyohron Co., Ltd., 2011

[2] Yukio Murayama, Kiso kara wakaru TCP/IP network jikken programming(Learn TCP/IP network programming from the basics), Ohmsha, Ltd., 2004

[3] Odaka Tomohiro, Kiso kara wakaru TCP/IP analyzer sakusei to packet kaiseki (Learn to make TCP/IP analyzer from the basics), Ohmsha, Ltd., 2004

[4] Yasutaka Yamamoto, Satoshi Kodama, Implementation of flexible network system using VNIC, IEICE Technical Report, Internet Architecture, ISSN 0913-5685, 2014.

[5] Takahumi Takeshita, Yukio Murayama, Tooru Arai, Yukio Karita, Masutaringu TCP/IP nyumonhen(Mastering TCP/IP(basic)), Ohmsha, Ltd., 2012