Implementation of server management system by using VNIC and bridge-connection

Yusuke Goto
Department of Information Sciences
Faculty of Science and Technology
Tokyo University of Science
Yamazaki 2641, Noda-shi, Chiba, 278-8510 Japan
Email: 6314623@ed.tus.ac.jp

Satoshi Kodama
Department of Information Sciences
Faculty of Science and Technology
Tokyo University of Science
Yamazaki 2641, Noda-shi, Chiba, 278-8510 Japan
Email: kodama@is.noda.tus.ac.jp

***Abstract*—In recent years, a machine performance has been improved. Accordingly, we have become able to simultaneously use virtual machines in a single computer. A virtual machine is able to avoid increasing an initial cost, a running cost, and so on. However, it is not easy for this system to manage a lot of services because the inner structure becomes more complicated. In particular, it is also difficult to control IP addresses owing to layering. This paper proposes a solution which has functions to be independent of the layering problem and to control the systems based on IP addresses. This solution is implemented by VNICs(Virtual Network Interface Card) and bridge-connection. We show the effectiveness of this solution through the result from experiments with this method.**

I. Introduction

In recent years, we have been able to use more than one service with a single machine by improving machine performance. Accordingly, we can have managed many services such as virtual machines. It is able to avoid increasing the number of physical machines.

However, there are two problems with it. Firstly, it is demanded much resource such as memory and hard drive capacity since it is the same thing to run many physical machines. Secondly, the inner structure of the single machine becomes more complicated. In particular, the setting based on an IP address has been extremely complicated in network. As stated above, it is not the best way to use the virtual machine which is commonly used recently.

In general, we use a method like an IP-based virtual hosting on a single server in order to solve the resource problem. Specifically, it is possible to combined multiple IP addresses into a single server. This system allows one server to share its resources. However, there remains another problem with the management because the inner structure is complex like running a plurality of virtual machines

Therefore, in this study, we propose the system which makes use of VNICs(Virtual Network Interface Card) as one way to solve those problems.

For example, the network is divided into some groups in a company, a university and so on. We allocate network segments the groups. Thereby, it is not possible to communicate between the groups. By binding these IP addresses to the VNICs, we can control local networks and operate independently each service such as virtual hosting.

The rest of the paper is the followings. Section 2, we explain an overview of the solution. In section 3, we show detailed description of functions, Then, in section 4, we show the result with this method. Finally, we explain the prospect which apply our proposed method.

March 17, 2015

II. System overview

We construct a system like Fig-1 which consists of the server, ordinary router and PC1. The server is connected to WAN port of the router with wired LAN. PC1 is also connected to LAN port of the router with wired LAN. The server has two VNICs. These VNICs and a NIC(Network Interface Card) are bridge-connected by our program.

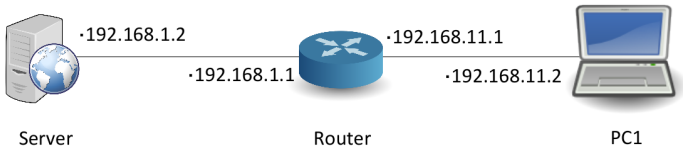


Fig-1 System Layout

III. Feature details

A. Implementation of VNIC

We implement Virtual Network Interface Cards and assign the IP addresses to VNICs. These VNICs are able to handle raw packets when bridge-connection with NIC is created.

In this paper, we assign any IP addresses in the different network segment of the router to VNICs and do not assign it to NIC like Fig-2. However, it cannot communicate with the router in this states. Accordingly, we rewrite the packet and let the server have a pseudo IP address. The server communicates with the router by using the IP address.

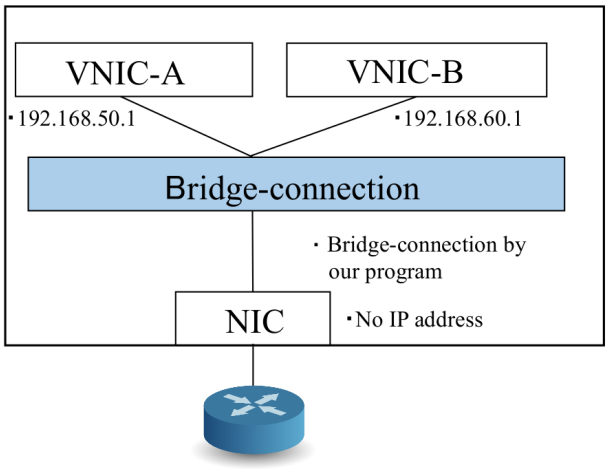


Fig-2 VNICs and NIC details

B. Segment separation

A static routing is often used to define exit points. In that case, a single PC has some NICs which are assigned IP address of the different network segment like Fig-3.

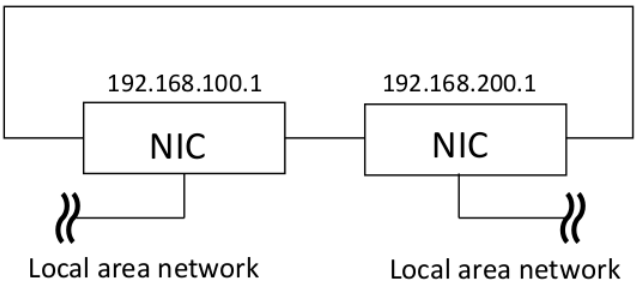


Fig-3 General static routing

In our proposed method, we use the static routing in order to separate the network segments of VNICs. In generally, It is impossible for other conventional devices to simultaneously communicate with multiple VNICs. Accordingly, we need the special settings because the server has not the multiple exit points but one exit point to VNIC-A and VNIC-B like Fig-2. It follows the procedures below.

1) We launch VNIC-A and VNIC-B in the server. The static routing can be added by using iproute2 commands that are `route add -net 192.168.50.0/24 gw 192.168.50.1 vnic-A` and `route add -net 192.168.60.0/24 gw 192.168.60.1 vnic-B`. These configurations are shown in Fig-4. (We named VNIC-A vnic-A and VNIC-B vnic-B in this experiment.)

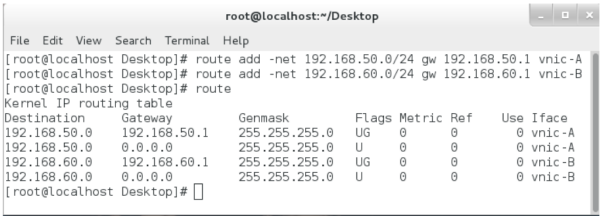


Fig-4 Commands and routing table

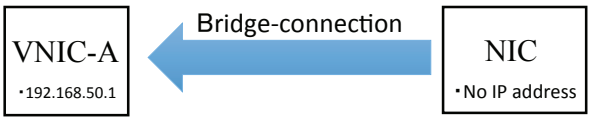
2) Thereby, the server has become able to ransmit the packet whose destination IP address is 192.168.50.* from VNIC-A(192.168.50.1). The same is true of 192.168.60.*. The server cannot generate the packet from both VNIC-A and VNIC-B without these settings.

However, it is not possible to communicate with the outside router and PCs. We described the communication system which uses the configurations in the following the next subsection.

C. Communication between VNICs and NIC

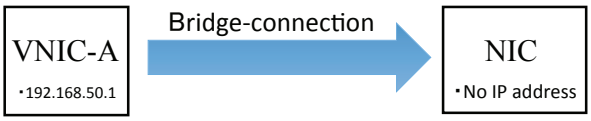
When the server transmits to the packet, we must rewrite its source IP address to 192.168.1.2 so that the server communicates with other ordinary devices. In this way, the server can communicate with the ordinary devices such as PC1 and the router. A server follows the procedure below when it transmit and receive data.

1) When the server receives the packet, our bridge-connection program distributes it to any VNICs.
2) If the server receives the packet at the VNIC-A, our bridge-connection program changes the destination IP address(192.168.1.2) to 192.168.50.1 and the source IP address(192.168.1.1) to 192.168.50.2 like Fig-5. When the server responds to the packet, our bridge-connection program changes the source IP address(192.168.50.1) to 192.168.1.2 and the destination IP address(192.168.50.2) to 192.168.1.1 like Fig-6.



・Source IP: 192.168.1.1 → 192.168.50.2
・Destination IP: 192.168.1.2 → 192.168.50.1

Fig-5 Transmission from NIC to VNIC-A

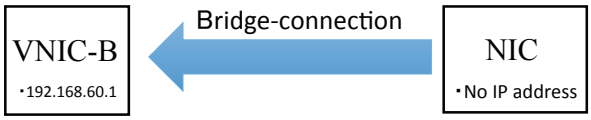


・Source IP: 192.168.50.1 → 192.168.1.2
・Destination IP: 192.168.50.2 → 192.168.1.1

Fig-6 Transmission from VNIC-A to NIC

3) If the server receives the packet at the VNIC-B, our bridge-connection program changes the destination IP address(192.168.1.2) to 192.168.60.1 and the source IP

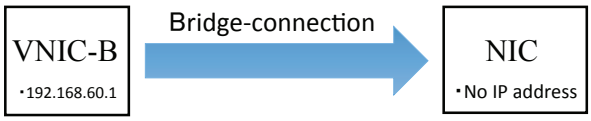
address(192.168.1.1) to 192.168.60.2 like Fig-7. When the server responds to the packet, our bridge-connection program changes the source IP address(192.168.60.1) to 192.168.1.2 and the destination IP address(192.168.60.2) to 192.168.1.1 like Fig-8.



·Source IP: 192.168.1.1 → 192.168.60.2

·Destination IP: 192.168.1.2 → 192.168.60.1

Fig-7 Transmission from NIC to VNIC-B



·Source IP: 192.168.60.1 → 192.168.1.2

·Destination IP: 192.168.60.2 → 192.168.1.1

Fig-8 Transmission from VNIC-B to NIC

We use the IP addresses such as 192.168.50.2 and 192.168.60.2 so that the server can communicate with ordinary devices in the configuration shown in Fig-4. Thereby, it become possible to transmit a packet to the router(192.168.1.1), and the router recognizes the server as 192.168.1.2.

In this time, there are also some source MAC addresses because the source MAC addresses of VNICs are different. This state causes a duplication error of MAC address. Owing to this, the server needs to unify the source MAC address into only one. In order to prevent from conflicting the source MAC addresses on the arp-table of the router, we use a hardware address of NIC.

D. Communication with HTTP server

Generally, a HTTP server can only be assigned one IP address. Then, we are able to manage multiple HTTP servers by using virtual hosting. The HTTP server is configured with multiple NICs and VNICs on the multiple local IP addresses. At this time, each of local IP addresses belong to the same network segment. If we divided those virtual hosting into some groups, it is effective to divide in network segment.

In this proposed method, we assign any IP addresses to VNICs. Each of IP addresses do not belong to the same network segment. In addition, we bind each of IP addresses to HTTP servers(Fig-9) and independently manage these HTTP servers. The server can deal with the packet on the same port. We described below to the communication between PC1 and the HTTP servers launched inside the server.

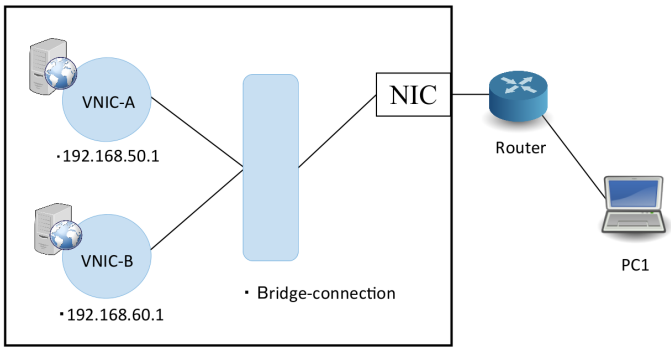


Fig-9 System layout using HTTP servers

1) We launch two HTTP servers and bind any IP addresses to each HTTP server-A and HTTP server-B. In other word, any IP addresses are bound to VNIC-A and VNIC-B.
2) An ordinary machine such as PC1 accesses a HTTP server in the server. PC1 sends the request packet to the server. The destination IP address of the packet is 192.168.1.2 in that time.
3) The server received the packet at NIC and distributes it to VNIC-A and VNIC-B by bridge-connection. The server arbitrarily distributes the packet to either VNIC-A or VNIC-B. At this time, the server communicates with the ordinary device to rewrite the packet by the method shown in the subsection C.
4) The HTTP server bound the VNICs respond to the client such as PC1. The client receives the HTTP reply packet from the server and can view the web pages. The web pages can be switched in response to the HTTP server bound any VNIC and IP address.
5) The client PC1 can see two types of web page, but the client needs one IP address and domain name to access the web page.

As a result, the server seems like a single system if we see it from outside. If we see the server from inside, it has multiple systems.

IV. The Verification Result

To confirm that it meets the demands performance specifics, we test operation using Fig-1 system and Fig-4 configuration.

A. Communication with each of VNICs

PC1 attempts to send a ping to the server. If our program lets VNIC-A respond to the packet, the results of experiment that are shown in Fig-10.

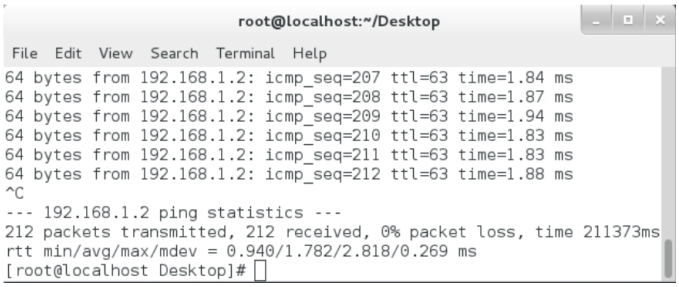


Fig-10 Communication between VNIC-A and NIC

If our program lets VNIC-B respond to the packet, the results of experiment that are shown in Fig-11.

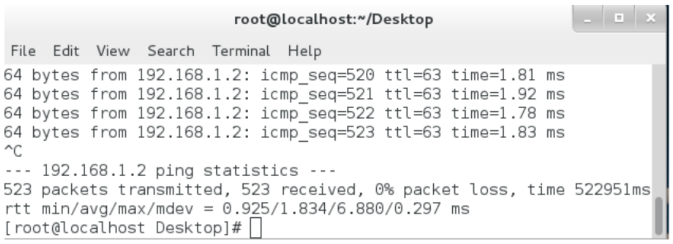


Fig-11 Communication between VNIC-B and NIC

According to the above results, we can ascertain the communication with each of VNICs in the same IP address(192.168.1.2).

B. The content in the packet on the path

We show the content in the packet on the path between VNIC-A and NIC. The communication that have been set forth in Fig-5 and Fig-6 is shown Fig-12 and Fig-13.

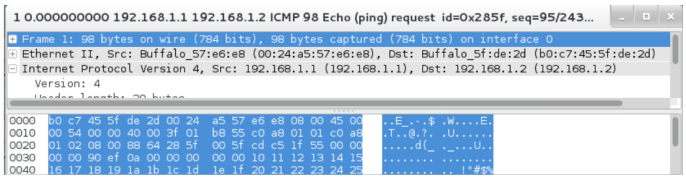


Fig-12 The content in the request packet on the NIC

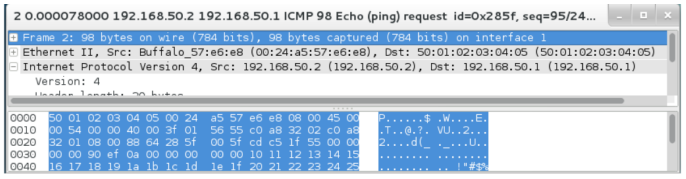


Fig-13 The content in the request packet on the VNIC-A

The communication that have been set forth in Fig-7 and Fig-8 is shown Fig-14 and Fig-15.

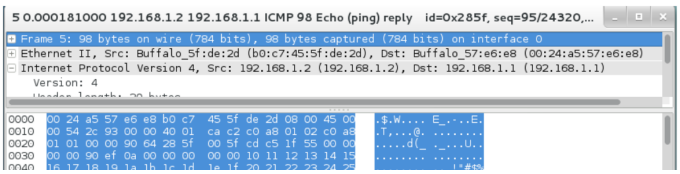


Fig-14 The content in the reply packet on the NIC

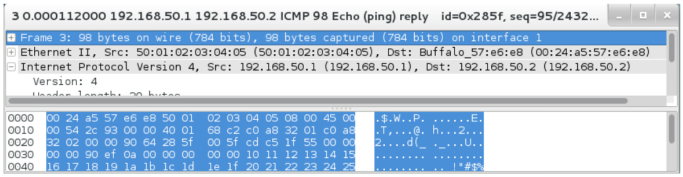


Fig-15 The content in the reply packet on the VNIC-A

C. Communication with each of HTTP servers

PC1 access to the HTTP server-A and the HTTP server-B . If our program lets the HTTP server-A respond to the request, the results of exciting that are shown in Fig-16.

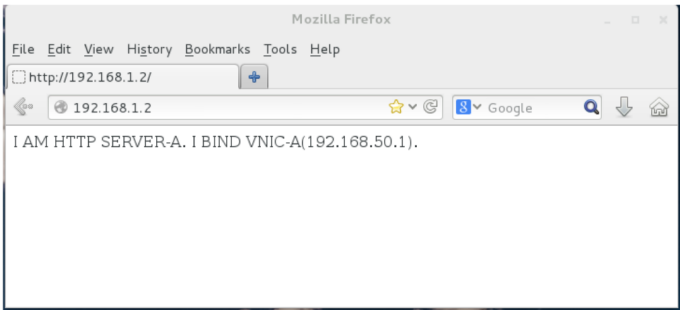


Fig-16 The web page in the server-A

If our program lets the HTTP server-B respond to the packet, the results of experiment that are shown in Fig-17.

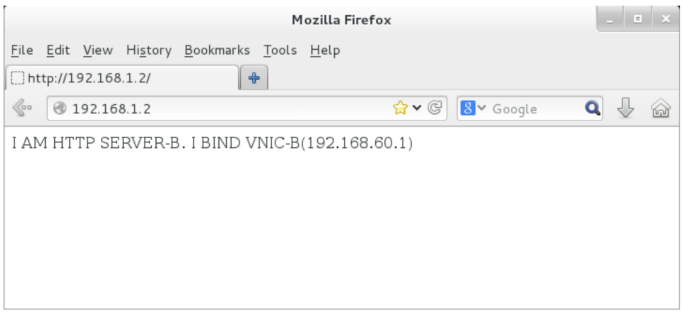


Fig-17 The web page in the server-B

According to the above results, we can access to the two HTTP servers in the same IP address(192.168.1.2) and the same port(80).

In this way, it becomes possible to connect multiple HTTP servers of different network segments without using a router in a single machine.

V. Future prospects

By applying this proposed method, we can assign IP address corresponding to URL(Uniform Resource Locator) to each of HTTP servers booted in a single of machine. For example, a client accesses to www.example.com/test1/index.html and www.example.com/test2/index.html in case of Fig-18. As shown in Fig-18, it is possible to separate two segments like 192.168.50.1 and 192.168.60.1 and to be operated separately the service itself. For that reason, it is useful to strengthen the security.

Additionally, for example, local networks such as companies and universities need to be separated into isolated categories to ensure the security. Then, we prepare two physical NICs and can makes local area networks forward to there like Fig-19 by applying this proposed method. Accordingly, PCs cannot communicate with other machines which are group in the other networks on this system. We can independently manage the groups and prevent a person in a network group from accessing to another network groups.

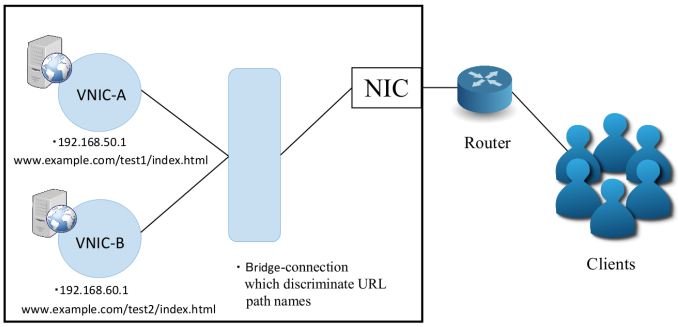


Fig-18 System layout using URL path name system

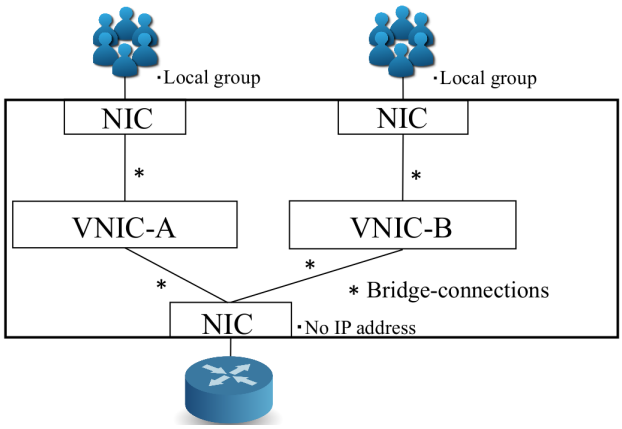


Fig-19 The grouping system

Acknowledgment

The authors would like to express their hearty thanks to Professor Shigeo Akashi and Mr. Yuta Watanabe who pointed out several errors and made a suggestive proposal to revise the manuscripts at this paper.

References

[1] Jonathan Corbet, Alessandro Rubini and Greg Kroah-Hartman, *Linux Device Drivers, 3rd Edition*, O'Reilly Media, 2005.

[2] Michael Kerrisk, *The Linux Programming Interface*, No Starch Press, 2010.

[3] Yasutaka Yamamoto and Satoshi Kodama, *Implementation of flexible network system using VNIC*, IEICE Tech. Rep., vol. 114, no. 286, IA2014-53, pp. 97-101, 2014.