Address-Translation-Based Network Virtualization

Yasusi Kanada, Toshiaki Tarui

Central Research Laboratory, Hitachi, Ltd.
Higashi-Koigakubo 1-280, Kokubunji, Tokyo 185-8601, Japan
{Yasusi.Kanada.yq, Toshiaki.Tarui.my}@hitachi.com

***Abstract* – Two network-virtualization architectures, namely, network segmentation and network paging, were investigated. They are analogical to two memory-virtualization architectures: segmentation and paging. Network paging, which is relatively new and is based on a type of network-address translation (NAT), is focused on. This architecture requires smaller packet size and has several more advantages over the conventional architecture (i.e., network segmentation). Intranet- and extranet-type communication methods based on this architecture are described. An address translators is placed at each edge router in the WAN and used to evaluate client-server communication under wide-area virtual-machine (VM) live migration as a case of extranet-type communication.**

***Keywords* – network virtualization; segmentation; paging; network address translation; NAT; extranet.**

I. Introduction

Network virtualization (NV) isolates multiple communities while using the same hardware, namely, computers, network nodes, and network links. It enables users to create their own wide-area networks. Virtual networks (VNs) are customizable and programmable. Because the developers of VNs can exclude the complicated and unnecessary features of conventional internet-protocol (IP)-based networks, the structure of a VN is much simpler. The developers can use simplified IP protocols such as IP−− [Oht 10] or can introduce non-IP protocols that are simpler, more powerful, and more efficient.

One of the complicated functions performed by real-world IP-based networks is network-address translation (NAT) [Zha 08] [Ege 94]. Using NAT is limited and complicated, so many engineers and scientists would prefer to avoid using it. However, it plays an important role in real-world networks. Conventional NAT [Sri 01] is useful when the number of available IP addresses is less than required. It is also useful when there are IP addresses that are only used locally or should be hidden from the global network.

Several types of address translation will play important roles in NV. Although all types of address translation can be called NAT, in this paper, the term "NAT" is not used for these types of address translation because NAT is usually used for conventional specific types of address translation, so it may cause misunderstanding. However, even in the case of conventional NAT, the localization and information hiding described above can be regarded as a virtualization function. Similar to dynamic address translation (DAT) used in memory virtualization, address translation is one of the two core functions that can be used for virtualization.

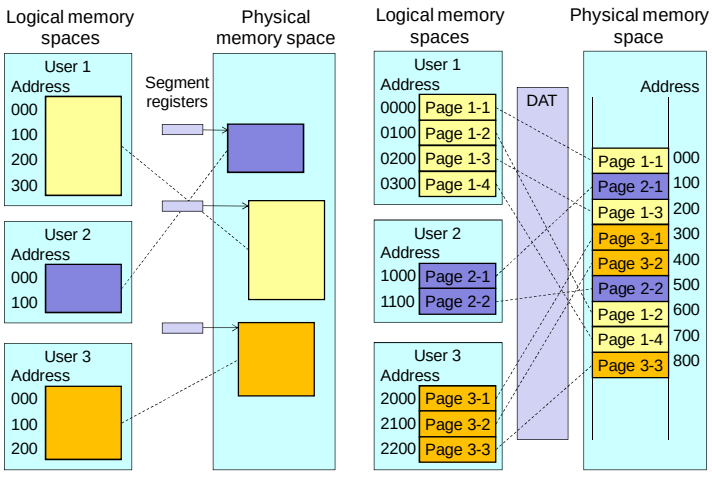
In the remainder of this paper, paging and segmentation in main-memory virtualization is explained and two NV architectures, namely, network-paging-based or address-translation-based architecture and network-segmentation-based architecture, are described in Section II. The former architecture is explained in detail in Section III, and a communication method using this architecture is described in Section IV. An application of address-translation-based virtualization, namely, wide-area VM live-migration, is presented in Section V. Related work is briefly reviewed in Section VI, and the paper is summarized in Section VII.

II. Paging and Segmentation

A. Paging and segmentation in main memory

Virtualization technology was first developed for virtualizing computer memory. In particular, data in memory was read and written by using virtual addresses. Two memory-virtualization architectures, segmentation and paging [Tan 08], were developed.

- *Segmentation*: A memory-virtualization architecture in which the memory space is divided into logically separated and variable-sized segments and each user uses a segment (see **Figure 1**(a)). Logical and physical memories are mapped to each other by using segment registers that point to the head of physical-memory segments. A memory address is represented by a pair consisting of a segment (register) number and a displacement in the segment.
- *Paging*: A memory-virtualization architecture in which the memory space is divided into fixed-size pages and the pages of all the users of a computer are mapped into a sin-



(a) Segmentation (b) Paging

Figure 1. Two memory-virtualization architectures

gle large address-space (see Figure 1(b)). Logical and physical memories are mapped to each other by using dynamic address translation (DAT). A memory address is represented by a number that indicates a point in the address space.

These architectures have advantages and disadvantages: segmentation is conceptually simpler, but paging is simpler to implement. They may also be used in combination.

B. Paging and segmentation in network

We can assume that there are two NV architectures that correspond to the above two memory-virtualization architectures because of the analogy between memory- and network-virtualization described below. In the case of memory virtualization, memory data are organized into a virtual-memory structure that differs from a real-memory structure. Similarly, in the case of NV, objects such as virtual machines (VMs) are organized into a virtual-network structure that differs from a real-network structure. Multiple memory-address spaces are created by memory virtualization. Similarly, multiple network-address spaces (or name spaces), in which virtual hosts, virtual nodes, and other virtual objects are identified, are created by NV. Both memory data and packets are read and written by using virtual addresses (or names), and the formats of data addresses and object addresses are similar too. The following two NV architectures are therefore assumed.

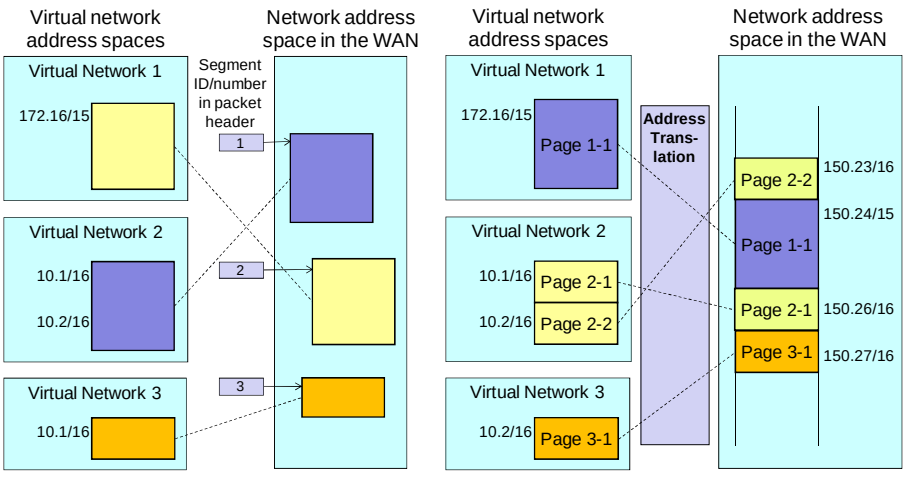
- *Network segmentation*: A NV architecture that distinguishes every network object by a pair, namely, a VN identifier (called a *segment identifier*) and a virtual address (or name), called an object *identifier* (OID) hereafter. VPN numbers or names, or VLAN identifiers, are used as segment identifiers (see Figure 2(a)). A real-network address is represented by a pair of these two identifiers, and each packet contains the sender's and the receiver's OIDs of this type. If identical OIDs are used in two VNs, they can be distinguished because their segment identifiers are different. This type of virtualization is widely used in VPNs and experimental virtual networks.
- *Network paging* (*address-translation-based virtualization*, *ATV*): A VN architecture that distinguishes every network object in all VNs by a single unique address. The OIDs are mapped into the address space of a wide-area (or global) network (WAN). This mapping is a type of NAT. A real-network address is represented by this single address, and each packet contains the sender's and the receiver's OIDs of this type. A virtual-address space may be divided into multiple pages and may be mapped to two or more non-contiguous subspaces in the WAN (see **Figure 2**(b)), although the page size may be varied because there is no hardware restriction. Each VN page must be mapped to a non-overlapping range of the WAN address space. If the same OIDs are used in two VNs, they are mapped to different addresses in the WAN.

Most conventional NV methods are based on network segmentation. Each data frame in a VN is encapsulated by a packet header of the substrate network (i.e., underlying network), and the segment identifier is in the packet header. Typical NV methods use IP-based encapsulation such as generic routing encapsulation (GRE) [Far 00], use layer-2 methods such as VLAN, or use multi-protocol label switching (MPLS). With these methods, GRE keys, VLAN tags, or MPLS labels contain the segment identifiers or labels that correspond to the segment identifiers.

In contrast to segmentation, network paging (i.e., ATV) seems to have been seldom used for communication between two or more sites of a VN. Conventionally, each local network behind a NAT is an independent network site, so it is not regarded as a VN site.

The format for identifiers is the same as that for network segmentation and network paging; that is, the addresses (or names) can be structured as a pair, i.e., (PS, F). PS represents the segment or page, and F represents the sub-address or field identifier that distinguishes the object from other objects in the same page or segment. Both PS and F may be numbers or symbolic names such as a fully qualified domain name (FQDN). Examples of PS and F are given as follows.

- *Numerical example*: PS = 172.16/16 (the first 16 bits of an IPv4 address) and F = *.*.10.21 (the last 16 bits). PS represents a network page.
- *Symbolic example*: PS = example.com and F = www. PS represents a segment.
- *Compound example*: PS = Government (a segment name) and F = 02.01.043 (a sub-address that consists of a department, a division, and a host number).



(a) Network segmentation (b) Network paging

Figure 2. Two network-virtualization (NV) architectures

Three differences between network segmentation and network paging are explained here. The first is the overhead caused by segmentation and paging. In segmentation, a segment identifier is added or removed at LAN-WAN borders. The packet size becomes larger in the

WAN; on the other hand, it is smaller in paging (i.e., LAN). However, the processing overhead of adding or removing a segment identifier is not large. In contrast, the processing overhead of address translation may be large.

The second difference concerns the size and number of segments and pages. In the case of paging, the WAN address space can be divided into a collection of pages, where the size and number of pages can be altered. However, in the case of segmentation, the size and number of segments cannot be changed because each segment is a logically separated address-space.

The third difference is between the methods for handling non-IP protocols. In regard to network segmentation, the encapsulated payload may contain a data frame of an arbitrary format. It may contain a non-IP frame; namely, non-IP protocols can be handled. In regard to network paging, non-IP protocols can also be handled, but the translators must map non-IP addresses to IP addresses if the WAN uses IPs.

Network segmentation and network paging can be combined. For example, paging can be used for data packets, and segmentation can be used for packets of the ICMP and those of routing protocols.

III. Address-Translation-Based Virtualization (ATV)

A. Requirements of ATV

Two conditions are required to enable ATV.

- *Identity of addresses in VN sites*: Addresses (OIDs) used for the same object at each site of a VN must be identical in ordinary cases. This means the mapping at the exit of the WAN must be the inverse of that at the entrance.
- *Isolation of VNs*: The address translator at the entrance of the WAN may not let a packet with a disallowed address pass through. Multiple addresses used for a VN must be mapped into different addresses in the WAN. In addition, the address translator at the exit of the WAN may not let a packet with a disallowed address pass through. These conditions are met by dropping a packet when no translation rule matches either the source or destination address.

An important difference between memory paging and network paging is that, in the latter, the address (OID) in an incoming packet must be translated in the inverse direction. Such inverse translation is not required in the former because incoming data, namely, memory content, do not contain an address.

B. Ordered/unordered addresses

With memory virtualization, the addresses are ordered. With NV, however, the OIDs are not necessarily ordered; that is, there are at least two types of identifiers.

- *Ordered identifiers*: The relationship between two identifiers, i1 and i2, is given as i1 > i2, i1 = i2, or i1 < i2. IP addresses are ordered identifiers because address ranges, or subnets, are meaningful.
- *Unordered identifiers*: No order between two identifiers is defined. The MAC addresses are unordered identifiers.

In the case of ordered identifiers, addresses on a page (in a certain range) can be translated using the same method as in memory virtualization. For example, because IP addresses are ordered, ATV using IP-to-IP translation is very similar to memory paging. The virtual address space can also be divided into multiple pages and mapped to non-contiguous addresses (see Figure 2(b)). In the case of IP addresses, a subnet can naturally be regarded as a page.

In contrast, unordered identifiers may have to be handled in a different way; that is, an output address may have to be specified for each input address. In such a case, the translation-table size must equal the number of MAC addresses in the virtual space. For example, MAC addresses are represented by a 48-bit number, but the order is not significant. Each MAC address may therefore have to be handled separately.

C. Types and varieties of mapping

Address translation is not necessarily restricted to the page-to-page type; that is, a translator may map contiguous addresses to non-contiguous addresses. If the virtual space is symbolic, the mapping is also non-numerical. However, numerical mapping is focused on here, and it is categorized into three groups. It is assumed that the original address is *ai* and the converted address is *ao*.

- *Contiguous translation*: This type of translation maps contiguous addresses to contiguous addresses. For example, $ao = ai + 100$. Figure 2(b) depicts this type.
- *Striped translation*: This type of translation maps contiguous addresses to striped addresses with constant strides. For example, $ao = 3 * ai + 1$.
- *Randomized translation*: This type of translation maps contiguous addresses to randomized addresses. For example, *ao* can be generated by a pseudo-random-number generator or a mapping table.

Randomized translation may be useful for security purposes because it makes address scanning difficult. Striped translation may be useful when there is a need to assign two or more WAN addresses to each virtual address or vice versa.

Three miscellaneous issues concerning mapping are described below. First, if the address format used in the VN is structured, it is possible to map part of the address that is functional in the WAN to the WAN address, and to store the rest, or whole address, in the payload. For example, if an address consists of a locator and a host-identifier, and the WAN is an IP network, the former can be mapped to an IP address, and the latter can be stored in the payload.

Second, an address in the VN can be translated into combination of addresses in two or more layers in the WAN; namely, the WAN address may contain information of individual hosts (i.e., MAC addresses). This representation probably works well in small-scale WANs such as enterprise-wide networks, but it is difficult to use them in a large-scale network because this representation is not scalable.

Third, in network paging, if the same type of address space is used for the VN and the WAN, and there is no address conflict, address translation is unnecessary (but an access control may be required) between them. An example of this case is given in Section V; however, this is a special case, and address translation is usually required.

D. Advantages and disadvantages of ATV

The advantages and disadvantages of ATV compared to segmentation-based virtualization are listed here. The advantages are as follows.

- *No overhead and less redundancy in packets*: There is no overhead in terms of packet size and less redundancy in the WAN. For example, in the case of IP-to-IP translation, the packet is the normal IPv4 packet. In contrast, in the network-segmentation-based method, the packet must have a tunnel header, which contains the segment identifier, in the WAN.
- *Availability of WAN functions*: Virtualized packets may utilize WAN functions because the behavior of the packets depends on the WAN addresses; e.g., if the WAN is an IP network, the functions of ICMP or routing may be useful.
- *Availability of NAT implementations*: Although conventional NAT and address translation required for virtualization are different, implementations of the former may be enhanced to include functions required for the latter. In particular, because of the IPv4 address exhaustion problem, a high-performance carrier-grade (large-scale) NAT [Nis 09] will be deployed. It may be used for virtualization, and it will enable wire-rate translation performance.

The disadvantages of ATV are listed as follows.

- *Potentially large memory size and slow rate of processing*: Address translation requires rule memory (or translation-table memory) and long? processing time. The required memory size may be large. In that case, it may be difficult to process address translation at the wire rate.
- *Restriction on OID formats*: The OIDs of hosts or nodes in the VN must be mapped to addresses in the WAN. This may restrict the syntax and/or semantics of the OIDs.
- *Possible conflict with WAN function*: VN functions may cause conflict with WAN functions; e.g., if the WAN is an IP network, address translation may make routing work in an unexpected way on the VN.

Although the segmentation-based method seems to be easier and simpler in many cases, ATV has several advantages such as smaller packet size, flexible page size, and page-by-page processing. Several examples of network paging are presented in the next section.

IV. Communication Through A WAN

Two types of communication method between two VN sites through a WAN are shown in this sections. The first type involves intranet-type communication (i.e., two sites are in the same VN) and the second type involves extranet-type communication, where the two sites are in different VNs, but they are allowed to communicate with each other.

A. Intranet-type communication

A paging-based intranet-type communication (i.e., communication in a virtually closed network) is illustrated here. It is assumed that there is a VN with at least two sites (see **Figure 3**), which are connected through a WAN. It is also assumed that IPv4 is used in the WAN, but other protocols such as IPv6 or Ethernet (i.e., VLAN) can also be used. The sites can thus be connected through the WAN using IPv4.

If the WAN is a closed network, it is possible to put a translator at every external interface of the WAN edge routers to inhibit any unauthorized access to the VN. This set up is similar to a memory-paging architecture, because access control is a function of DAT, and usually no memory access can bypass the DAT. In contrast, if the WAN is an open network, such as the Internet, it is difficult to exclude unauthorized accesses. Because host addresses are mapped to the WAN address space, unauthorized users at a third site may illegally access the hosts. This can result in a security risk, so such access must be inhibited.

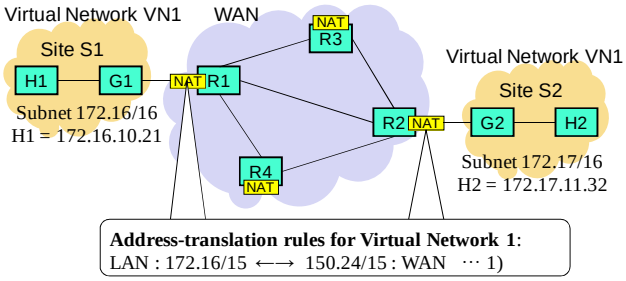


Figure 3. Intranet-type communication between two ATV-based VN sites

Figure 3 shows a translation rule that is required for communication between two sites, S1 and S2, through a VN. The same rule is used for both sites. It is applied to both the source and the destination addresses of a packet.

Only one page (a single rule) is used in this example. However, there may be multiple pages; for example, sites S1 and S2 may use different pages that map to non-contiguous pages in the WAN. If there are two or more rules, they are conceptually applied sequentially; namely, the first rule that matches the packet is used. However, the data structure of the rules can be optimized, and indexing or hashing may be used.

The meaning of the rule is described as follows.

- If a packet comes from the LAN, and if the source or destination subnet (the first 16 bits of the address) is 172.16/16 (or 172.17/16), it is translated to 150.24/16 (or 172.25/16); i.e., the rule is applied from left to right. The sub-address (the last 16 bits) is invariant (see Figure 2(b)).
- If a packet comes from the WAN, and if the source or destination subnet is 150.24/16, it is translated to 172.16/16; i.e., the same rule is applied inversely (from right to left), as described in Section III *A*. The sub-address is invariant.

Since there are no other rules, a packet with a source or destination address that is not specified in any rule is dropped.

Data packets are processed as follows. When host H1 at site S1 sends a packet to host H2 at site S2, the source address (172.16.10.21 in the figure) is translated into a WAN address (150.24.10.21) at ingress edge-router R1. The destination address (172.17.11.32) is also translated into a WAN address (150.25.11.32). This means Twice NAT [Sri 99] is applied to the packet. A Twice NAT is a type of NAT that modifies both the source address and the destination address. These addresses are IP addresses if the VN uses IP, but they may be another type of identifier if it uses a non-IP protocol.

The simple contiguous translation described above was used. However, as described in Section III *C*, the address in VN1 can be separated into a subnet and a sub-address (host address), which are handled separately. Namely, the subnet can be mapped into a WAN address, and the sub-address can be put in the payload. This type of address translation is more like the conventional NAT that distinguishes local addresses by port numbers.

If dynamic routing is used in the VN, routing-protocol messages should be passed through the LAN-WAN borders. The messages will probably work well if dynamic routing is also used in the WAN. In this case, the edge routers of the WAN must translate the subnets in the messages when they pass through the address translation. When importing routes from the WAN to a VN site, they must be properly filtered. No routing-protocol extension is usually required to convey routes in the VN.

B. Extranet-type communication

A paging-based typical extranet-type communication (i.e., between intranets with access control) is illustrated here. There are assumed to be three VNs, i.e., VN1, 2, and 3 (see **Figure 4**). Each VN has only one site. Hosts at site S1 can communicate with hosts at the other two sites, S3 and S4. Hosts at site S3 can communicate only with hosts at S1 and S3, and hosts at site S4 can communicate only with hosts at S1 and S4. Figure 4 shows the translation rules, 1, 2, and 3 (three pages), required for communication between the three sites, S1, S3 and S4.

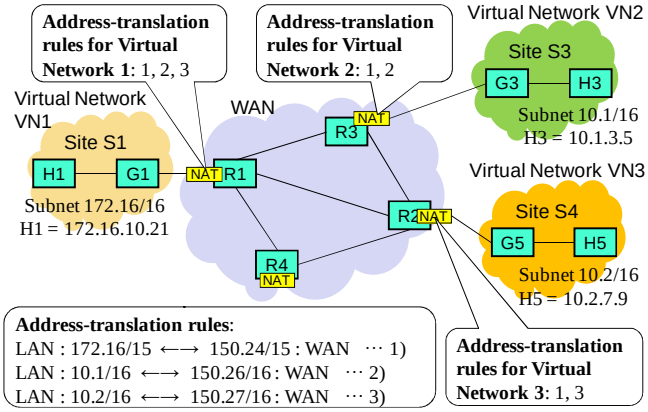


Figure 4. Extranet-type communication between two ATV-based VN sites

Hosts at S1 can communicate with hosts at S3 across the WAN. Rules 1 and 2 are used at edge routers R1 and R3 for this purpose. Data packets are processed using the following method (see Figure 2(b).) When host H1 at site S1 sends a packet to host H3 at site S3, the source address of the packet (172.16.10.21) is translated by using rule 1 for VN1 into a WAN address (150.24.10.21) at ingress edge-router R1. This translation is the same as in the intranet case. However, the destination address (10.1.3.5) is translated by using another rule, rule 2 for VN2, into a WAN address (150.26.3.5). The source- and destination-addresses are translated in the reverse direction at egress edge-router R3, so host H3 sees the original addresses. Hosts at S1 can also communicate with hosts at S4. Rule 1 and rule 3 for VN3 are used at edge routers R1 and R2.

On the contrary, hosts at S3 and S4 cannot communicate with each other because there are no rules for their communication in edge-routers R3 and R2. Namely, R3 does not have rule 3, and R2 does not have rule 2. For example, if a host address at S3 (10.1.3.5) is specified in a packet that comes from S4, the translator in the ingress edge-router R2 drops this packet because R2 does not have a rule that matches this address.

This type of access control can be specified page by page. This means that if a virtual-address space is divided into multiple pages, access to each page from outside the site can be controlled by the existence or non-existence of a rule for the page because each rule is defined for one page.

If there is no need to control access, rules 2 and 3 can be replaced by a single rule with doubled address ranges, which is similar to the rule used in the intranet example. However, they are separated for the purpose of access-control. If the WAN is the Internet, hosts in the VNs can communicate with hosts in the Internet if each router holds rules that map the hosts' addresses.

V. Application To VM Migration

To improve the performance of client-server communication under wide-area live migration of server virtual machines (VMs) (i.e., when server VMs are migrating between data centers [Kan 11]), the extranet-type communication method was applied. This method is briefly outlined in the following.

Wide-area VM live-migration between data centers can solve problems such as load balancing, disaster avoidance and recovery, and power saving. However, to enable migration between distant locations, other problems must be solved. One problem is "address warping". When a server VM is moved from one location to a more distant location, the IP and MAC addresses "warp" from the source server to the destination server. This confuses or complicates the status of both the WAN and LANs in a short time. This problem may cause a serious failure in the case of real-time traffic such as that involved in conferencing or on-line games.

This problem is solved by putting two data centers in different VNs, VN1 and VN3, as shown in **Figure 5**. This set up allows the VMs before and after migration that have identical IP and MAC addresses to briefly coexist. The IP addresses of the VMs are mapped to different addresses in the WAN, so no confusion occurs when the VM moves.

Users of the VM belong to another VN (VN2). When the VM is in source data center DC1, the address of the VM (172.16.10.21) in the users' VN is mapped to the VM before the migration from DC1. However, when the VM moves to destination data center DC2, a translation rule at the edge routers connected to the users' sites, R2 and R3, is switched, and the address of the VM in the users' VN is mapped to the VM after the motion in DC2.

The feasibility of ATV was tested by using IP-based simulated WAN and VNs. The WAN consisted of three layer-3 (L3) switches, translators (Linux PCs) connected to the L3

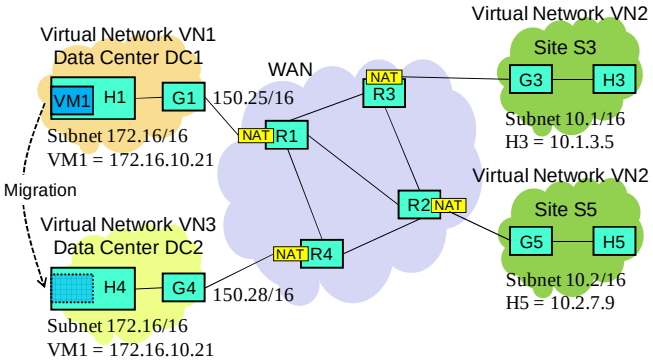


Figure 5. Wide-area VM migration using ATV-based VNs

switches, two sets of servers managed by VMware® at two simulated data centers, and a client PC at a simulated user site. The page size was 2^{16}. After the VM motion, RARP (reverse address resolution protocol) messages were generated by VMware and detected. The VN was then switched by a tool developed by an author [Kan 11].

In the current version of the translator, addresses in ICMP packets are not fully translated, so the function of the VNs is still limited. However, a UDP packet generator was embedded in a VM, and packets were captured at user sites. As a result, it was confirmed that a VM motion switched the VN and that the VNs worked correctly without any confusion.

VI. Related Work

NAT has been used to connect to the Internet by using only one IPv4 address or fewer addresses than the number of hosts. Because detailed standards are lacking, NAT has been developed in an ad hoc way. Much work has therefore been devoted to finding systematic methods for solving this problem. Among this work are IPNL [Fra 01] and IP4+4 [Tur 02], which enable communication between hosts behind the NATs. However, they were not intended to be used with virtualization applications.

Okada, et al. [Oka 02] described a method of deploying extranets using Twice NAT. This method is similar to our method described in Section V, but their major purpose was to deploy extranets without using multiple global addresses. They did not generalize their method for virtualizing networks.

Hasenstein [Has 97] mentioned the roles of NAT in NV and explained that he wanted to show where NAT might find or had already found its place in the entire virtualization scheme. Hara, et al. [Har 03] briefly mentioned the use of NATs in extranets. However, they did not discuss NAT as a virtualization medium.

VII. Conclusion

Two network virtualization (NV) architectures, namely, network paging and network segmentation, were described and compared. An address-translation-based virtualization (ATV, i.e., network-paging-based) method was investigated. Intranet- and extranet-type communication methods based on this architecture were proposed. An address translator was placed at each edge router in the WAN, and extranet-type communication during wide-area live migration of VMs was evaluated as a special case of extranet-type communication.

Segmentation has been widely used for NV, and segmentation-based methods would seem to be easier or more efficient in many cases. However, ATV-based methods have several advantages, such as less packet overhead, flexible page size, and page-by-page processing. As in the case of memory virtualization, segmentation and paging may also be used in combination to combine advantages of both architectures. Network paging is therefore a promising NV architecture. The authors will continue to develop and evaluate ATV-based methods, ways of utilizing the underlying network function in VMs, and a method for developing programmable networks based on ATV.

Acknowledgment

Part of the research results described in this paper is an outcome of the Eco-Internet Project (“R & D on Power-Saving Communication Technology − Realization of Eco-Internet −”), performed in fiscal year 2009, which was funded by the Ministry of Internal Affairs and Communications of the Japanese Government.

References

[Ege 94] Egevang, E. and Francis, P., “The IP Network Address Translator (NAT)”, RFC 1631, IETF, 1994.

[Far 00] Farinacci, D., Li, T., Hanks, S., Meyer, D., and Traina, P., “Generic Routing Encapsulation (GRE)”, RFC 2784, March 2000.

[Fra 01] Francis, P. and Gummadi, R., “IPNL: A NAT-Extended Internet Architecture”, *ACM SIGCOMM 2001*, pp. 69–80, August 2001.

[Har 03] Hara, Y., Ohsaki, H., Imase, M., Tajima, Y., Maruyoshi, M., Murayama, J., and Matsuda, K., “VPN Architecture Enabling Users to be Associated with Multiple VPNs”, *5th Asia-Pacific Symp. on Information and Telecomm. Tech.* (*APSITT 2003*), November 2003.

[Has 97] Hasenstein, M., “Diplomarbeit – IP Address Translation”, http://www.hasenstein.com/HyperNews/get/linux-ip-nat.html .

[Kan 11] Kanada, Y. and Tarui, T., “A “Network-Paging” Method for Wide-Area Live-Migration of VMs”, *25th International Conference on Information Networking* (*ICOIN 2011*), January 2011.

[Nis 09] Nishitani, T., Yamagata, I., Miyakawa, S., Nakagawa, A., and Ashida, H., “Common Functions of IP Address Sharing Schemes”, draft-nishitani-cgn-05, Internet Draft, IETF, July 2010.

[Oht 10] Ohta, M. and Fujikawa, K., “IP− −: A Reduced Internet Protocol for Optical Packet Networking”, *IEICE Transactions on Communications*, E93.B, No. 3, pp. 466-469, 2010.

[Oka 02] Okada, K., Chen, E. Y., Komiya, T., and Fuji, H., “Deploying User-based Extranet without Global Addresses”, *IPSJ SIG Notes*, CSEC 2002(43), pp. 7–12, Information Processing Society of Japan, May 2002.

[Sri 99] Srisuresh, P. and Holdrege, M., “IP Network Address Translator (NAT) Terminology and Considerations”, RFC 2663, IETF, August 1999.

[Sri 01] Srisuresh, P. and Egevang, K., “Traditional IP Network Address Translator (Traditional NAT)”, RFC 3022, IETF, January 2001.

[Tan 08] Tanenbaum, A. S., “Modern Operating Systems”, Third Edition, Pearson Prentice Hall, 2008.

[Tur 02] Turányi, Z. and Valkó, A., “IPv4+4”, *10th IEEE Int’l Conference on Network Protocols* (*ICNP’02*), November 2002.

[Zha 08] Zhang, L., “A Retrospective View of Network Address Translation”, *IEEE Network*, Vol. 22, No. 5, pp. 8–12, September/October 2008.