Automatic Construction of Mini Data Center and Wide Area Network Using Virtual Machines and Virtual Routers

Kenji FUJIKAWA[1], Yasunaga KOBARI[1], Hiroaki HARAI[1]
1: National Institute of Information and Communications Technology, 4–2–1 Nukui-kita, Koganei, Tokyo, 184–8795, Japan

***Abstract*—We redefine an address, which is location information within the Internet, as a locator, and propose a HANA protocol, which hierarchically and automatically allocates and assigns locators to network equipments. From the Internet core to stub networks, HANA can reduce manual configurations of network equipments related to locators as possible, and then can reduce network management costs. Even if a stub Autonomous System (AS) changes its subscribing ISP to another, manual configurations related to locators are not required. We construct a wide area network using virtual routers, and also construct a mini data center using 200 virtual machines. Under these environments, we execute the HANA protocol and show HANA enables automatic locator configurations in these networks.**

***Index Terms*—hierarchical automatic number allocation, virtual machine, virtual router, JGN-X**

I. Introduction

The current Internet is already an indispensable part of our social infrastructure, yet there are many many problems within it. These includes,

- The BGP routing table size of the Internet continues to increase.This prevents quick rerouting when a network failure occurs.
- Routers require manual address settings.

One of the reasons the global routing table is increasing is that a site that requires fault tolerance tends to multihome multiple upper Internet service providers (ISPs), and obtain multipaths to the Internet. Such a site is allocated an autonomous system (AS) number and a provider independent (PI) address space, and makes the upper ISPs advertise these using BGP.

It is well-known that introducing provider aggregatable (PA) addresses is effective in decreasing the BGP table size. Previous work [1] shows that introduction of PA addresses can reduce the BGP table size by 93.3%-96.3% when only stub ASes adopts the PA locator allocation. If the intermediate-level of ASes, which can be ISPs but not Tier1 ISPs, adopt hierarchical locator allocation, the BGP table size can be reduced by 99.1% for 90.2% of the ASes. If a site is allocated multiple PA addresses from different ISPs and utilizes them, the PA addresses do not influence the size of the BGP routing table. In reality, a site that desires fault tolerance does not actually introduce PA addresses, because there are still some issues concerning the method of of effective introduction, one of which is address allocation and assignment of multiple addresses.

Under the PA address scheme, if a site multihomes multiple ISPs, it is to be allocated multiple address spaces. If it changes its subscribing ISPs, then it has to be reallocated a new address space, and assigns new addresses to routers and hosts in the site. This is an extremely weighty task for network managers, and prevents the deployment of PA address multihoming.

We are working on a new-generation network design project called AKARI [2], [3], where we take a clean-slate approach that designs a new network architecture without being constrained by the legacy protocols or environments. We redefine an address, which is location information within the Internet, as a locator, and propose a Hierarchical/Automatic Number Allocation (HANA) protocol, which hierarchically and automatically allocates and assigns locators to network equipments.

We introduced HANA to commercial routers using virtual router (VR) functions and constructed a wide area network on nation-wide network testbed. We also constructed a mini data center using 200 virtual machines (VMs) and connected it to the wide area network based on HANA. We executed the HANA protocol on these environments, and verified that the HANA protocol is effective for constructing PA-address-based multihoming network environments.

The rest of this paper is organized as follows. In Section II we give an overview of HANA. In Section III we explain implementation of HANA. In Section IV, we describe automatic construction of a wide area network and a mini data center using VRs and VMs.

II. Overview of HANA

Here we show the locator structure and protocol of HANA. Refer to [4] for details.

A. Structure of a Locator

We define the upper part of a locator, the following part and the rest, as *prefix*, *midfix*, and *suffix*, respectively, and use the notation shown in Fig. 1 for indicating prefix/midfix/suffix length. A locator is N bits long. When the prefix length is n bits long, the length of the following midfix is $m - n$ bit long, and the length of the following suffix is N-m bit long. These are written as $/n$, $/n$-m, and $/m$-N, respectively. A prefix and the following midfix at a certain level of hierarchy can be combined into a new prefix for the level underneath.

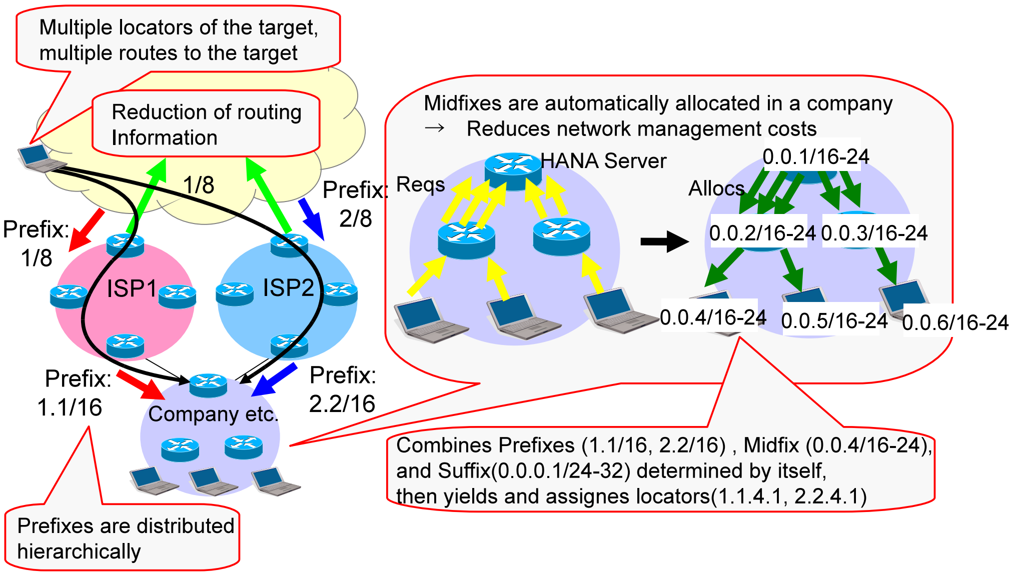

Fig. 2. Overview of the HANA protocol

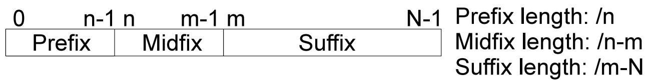

Fig. 1. Structure of a locator

We assume the locator space of about 64 bits. This is same as the number of bits in IPv6 address space excluding an interface ID. Thus, this can cover both of IPv4 and IPv6 address spaces. (For the sake of simplicity, we employ the dotted-decimal address notation of IPv4.)

B. *Overview of the HANA protocol*

Basic principles of the HANA protocol are as follows:

- Prefixes are distributed by the upper-level ISP. Each prefix is created by combining a midfix in the level and a prefix distributed from the upper-level.
- Midfixes are automatically allocated in a certain managed domain, even if the domain is disconnected from the upper ISP.
- Only routers in the borders of the area and HANA servers require manual settings of the prefix/midfix lengths. The other nodes are exempt from the address settings.

The HANA protocol has the functions of DHCP and IPv6 Router Renumbering (RR) [5], and integrates them. In the HANA protocol, midfixes are allocated in a manner similar to DHCP. As for differences from DHCP, ranges of midfixes can be varied according to each layer, and the midfix allocation can be adapted to routers in multiple layers. Prefixes are distributed with varying their values according to the situation of the midfix allocation. This is a different point from IPv6 RR.

Fig. 2 shows an overview of the HANA protocol and a system based on it. ISP1 and ISP2 are allocated address spaces of $1/8$ and $2/8$, respectively, and allocate and distribute address spaces of $1.1/16$ and $2.2/16$ to Company, etc. network, respectively.

In a Company network, the middle parts of locators are locally allocated and assigned. For example, in Fig. 2, the top of the routers becomes a HANA server and allocates 0.0.1/16-24, 0.0.2/16-24 and 0.0.3/16-24 to the routers, including itself. It also allocates 0.0.4/16-24, 0.0.5/16-24 and 0.0.6/16-24 to the terminal hosts.

As a result, for instance, one of the end hosts can utilize locators of 1.1.4.1 and 2.2.4.1, which are the combined locators of the distributed prefixes (1.1/16, 2.2/16), the allocated midfixes (0.0.4/16-24), and the suffixes it determines (e.g. 0.0.0.1/24-32).

III. Implementation

We implemented a HANA-based system on Linux OS (Ubuntu 10.04 LTS). The system consists of processes called *hanad* and *hanaroute*, are explained in the following sections.

A. *hanad*

A hanad is a main process of the HANA system. It runs in a HANA node, connects to the adjacent hanad in other HANA nodes, and exchanges HANA messages. It is controllable via Command Line Interface (CLI) that is established by TCP or telnet. The CLI is used by a network manager and the other HANA processes.

B. *hanaroute*

A hanaroute obtains locators and route information from hanad using the CLI of hanad, and reflects them to its HANA node. That is, it sets locators of the interfaces and next hop information of the forwarding table of the HANA node.

By splitting a hanad and a hanaroute, HANA is adaptable various node environments without modifying the hanad that

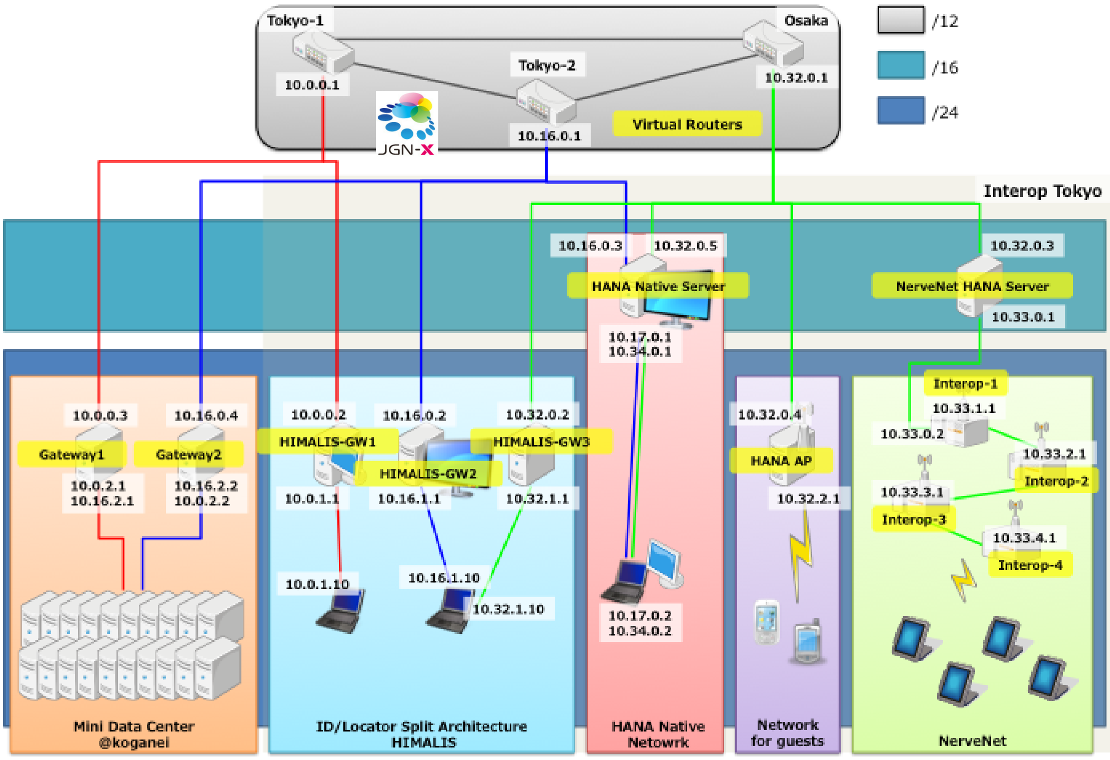

Fig. 3. Structure of all the networks

implements the HANA basic functions. The hanaroute currently supports Linux OS and Junos.

IV. Automatic Network Configuration by HANA

We automatically constructed a wide area network and a mini data center using virtual routers (VRs) and virtual machines (VMs). These are detailed in the following sections.

A. Structure of all the networks

We constructed the backbone of the wide area network by using VRs provided by JGN-X [6] (Fig. 3). Two VRs are placed in Tokyo, and one in Osaka. These are mutually connected via a VLAN, which is provided by JGN-X.

We constructed stub HANA networks under the backbone, and connected them to the backbone by JGN-X VLANs that are different from each other. One of the stub networks is a mini data center located in Koganei. This is describe in the next section. The others include HIMALIS [7] network, which supports ID/Locator split architecture, and a NerveNet [8]. HANA also automatically allocates locator spaces to these stub networks. When a stub network is connected to multiple VRs, it is allocated multiple locator spaces.

B. Construction of a wide area backbone network

JGN-X implements VRs by using VR functions provided by Juniper MX80 series. It is hard to directly install the HANA functions into VRs in our environment, so we implements a remote control function of VRs on a hanaroute, which was mentioned above.

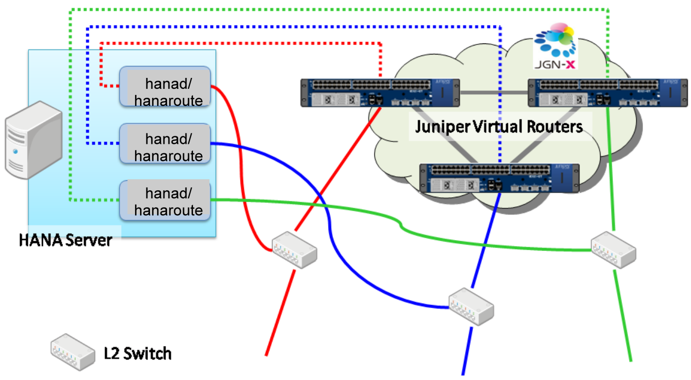

Fig. 4. Construction of a wide area backbone network

Fig. 4 shows the structure of a wide area backbone network on JGN-X. For simple management, we run three pairs of hanad and hanaroute on one PC located at Koganei, the site of our lab. Each pair of hanad/hanaroute controls one VR on JGN-X. For this, we prepared a dedicated VLAN for hanad/hanaroute to control VRs.

Manual configurations are required for hanaroutes, since they control remote VRs. However, for locator settings, each of the hanads is only set up with an IPv4 locator space of /12. For locators, no other settings are required, including interfaces' locators. This proves that HANA can automatically configure locators of a wide area backbone network.

C. Construction of a mini data center

To prove that HANA is adaptable to a local area network consisting of hundreds of nodes, we constructed a mini data center in Koganei. This consists of two PC Gateways (GWs) and 20 host PCs for VMs. One of the GWs also runs as a HANA server. Each host PC executes 10 VMs, so 200 VMs run in the mini data center. One of the VMs in a PC runs as a router, and the other nine VMs run as HTTP servers. These are connected as shown in Fig. 5. Although the HTTP server VMs do not have a direct link to the HANA server, they can be allocated locator spaces by the HANA protocol. Only two GWs require for locator settings; that is, VMs including the other routers and HTTP servers do not require for locator settings.

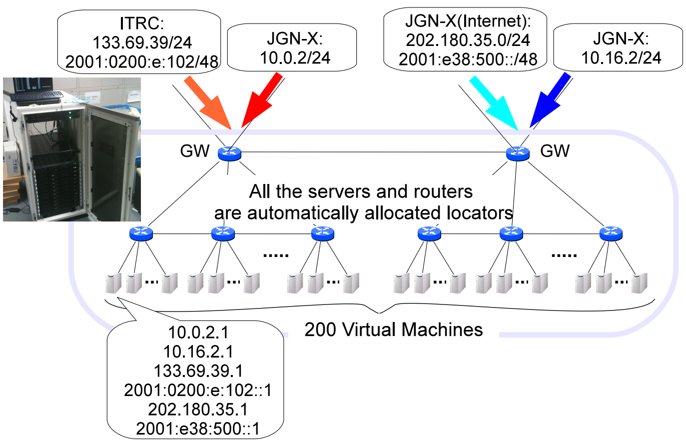

Fig. 5. Construction of a mini data center

GWs and VMs are allocated midfixes of $/24$-32 as IPv4, and $/56$-64 as IPv6. For prefixes, two IPv4 prefixes are distributed from VRs on JGN-X, and other two IPv4 prefixes and two IPv6 prefixes are distributed from the global Internet connections. Each node is resultantly allocated six locator spaces and assign its interfaces six locators from them.

Construction of the mini data center by HANA proves that the HANA enables automatic locator configuration of a LAN consisting of hundreds of routers and servers.

V. Conclusions

The HANA protocol hierarchically and automatically allocates and assigns locators to network equipments. To prove the HANA protocol is adaptable to the real networks, we constructed a wide area network using virtual routers, and also constructed a mini data center using 200 virtual machines. Under these environments, we executed the HANA protocol and showed it enables automatic locator configurations in these networks.

For future work, we will increase HANA-supported routers on JGN-X and HANA-supported stub networks, and verify and prove that HANA can be executed on wider area networks.

Acknowledgement

We would like to thank Youichi KOYAMA (Trans New Technology, Inc.) who assisted with this research.

References

[1] Y. Song, L. Gao, and K. Fujikawa, "Resilient routing under hierarchical automatic addressing," *Globecom 2011*, Dec. 2011.
[2] AKARI Project, "http://akari-project.nict.go.jp/."
[3] H. Harai, K. Fujikawa, V. P. Kafle, T. Miyazawa, M. Murata, M. Ohnishi, M. Ohta, and T. Umezawa, "Design guidelines for new generation network architecture," *IEICE TRANSACTIONS on Communications*, vol. E93-B, no. 3, pp. 462–465, 2010.
[4] K. Fujikawa, H. Harai, and M. Ohta, "The basic procedures of hierarchical automatic locator number allocation protocol HANA," *AWFIT2011*, Nov. 2011.
[5] M. Crawford, "Router renumbering for IPv6," *RFC 2894*, August 2000.
[6] JGN–X, "http://www.jgn.nict.go.jp/."
[7] V. P. Kafle and M. Inoue, "HIMALIS : Heterogeneity inclusion and mobility adaptation through locator ID separation in new generation network," *IEICE transactions on communications*, vol. 93, no. 3, pp. 478–489, 2010-03-01. [Online]. Available: http://ci.nii.ac.jp/naid/10026843953/
[8] M. Inoue, M. Ohnishi, C. Peng, R. Li, and Y. Owada, "NerveNet : A regional platform network for context-aware services with sensors and actuators," *IEICE transactions on communications*, vol. 94, no. 3, pp. 618–629, 2011-03-01. [Online]. Available: http://ci.nii.ac.jp/naid/10028230776/