PAPER

Quickly Converging Renumbering in Network with Hierarchical Link-State Routing Protocol

Kenji FUJIKAWA[†a)], **Hiroaki HARAI**[†], **Motoyuki OHMORI**[††], *and* **Masataka OHTA**[†††], *Members*

SUMMARY We have developed an automatic network configuration technology for flexible and robust network construction. In this paper, we propose a two-or-more-level hierarchical link-state routing protocol in Hierarchical QoS Link Information Protocol (HQLIP). The hierarchical routing easily scales up the network by combining and stacking configured networks. HQLIP is designed not to recompute shortest-path trees from topology information in order to achieve a high-speed convergence of forwarding information base (FIB), especially when renumbering occurs in the network. In addition, we propose a fixed-midfix renumbering (FMR) method. FMR enables an even faster convergence when HQLIP is synchronized with Hierarchical/Automatic Number Allocation (HANA). Experiments demonstrate that HQLIP incorporating FMR achieves the convergence time within one second in the network where 22 switches and 800 server terminals are placed, and is superior to Open Shortest Path First (OSPF) in terms of a convergence time. This shows that a combination of HQLIP and HANA performs stable renumbering in link-state routing protocol networks.

key words: *HQLIP, HANA, renumbering, convergence time, link-state routing protocol*

1. Introduction

In future networks, not only human-operated terminals, but also a huge number of non-human-operated terminals will be connected. They autonomously collect, process and distribute information. In order to handle created huge data, a large number of storage servers and data processing servers must be placed in data centers. In addition, virtual machines and virtual networks are dynamically defined in the data centers. These make network management more complicated. In the network that consists of a huge number of nodes, hierarchicalization is effective for suppressing routing messages and shorten the convergence time of route calculation. The hierarchicalization also helps network managers with grasping an overview of the network, and mitigates network management costs.

Link-state routing protocols such as OSPF/OSPFv3 [1], [2] and IS-IS [3] are commonly employed in data-center Layer 3 (L3) networks, enterprise L3 networks, and internal networks of ISPs. The link-state routing protocols support hierarchical routing. Hierarchicalization reduces the number of nodes that construct the network topology, which is the calculation target. That is, introducing hierarchical routing improves the convergence time of the route calculation. However, OSPF/OSPFv3 and IS-IS support two-level hierarchy at a maximum. Two-level hierarchy is not sufficient for extending the network-scale by simply combining and stacking networks. For example, assuming that a set of data center racks is already constructed using two-level hierarchy of OSPF, a network manager cannot simply combine two or more sets of them and run OSPF because of hierarchy-level shortage. Therefore, a multi-level hierarchical routing protocol that deals with a huge number of nodes and dynamic construction and modification of networks are required.

A several approaches that do not use link-state routing protocols are proposed in order to construct data centers with a large number of servers. Al-Fares et al. [4] and Niranjan Mysore et al. [5] enable routing without a link-state routing protocol by restricting a network topology to the one called fat-tree. Mudigonda et al. [6] uses VLANs for providing multi-paths. However, these approaches cannot utilize autonomous and dynamic routing that quickly reflects link addition/deletion or a link failure. Autonomous and dynamic routing is the essential function of link-state routing protocols.

In this paper, we propose a two-or-more-level hierarchical link-state routing protocol in Hierarchical QoS Link Information Protocol (HQLIP). We call a link-state defined in HQLIP *link information*. It conveys a metric and QoS values such as bandwidth and delay. We focus on the metric for the hierarchical routing in this paper. A link-state routing protocol has an advantage that the convergence time of route calculation for creating forwarding information base (FIB) is faster than that of a distance-vector routing protocol. However, it has the disadvantage that each router needs more calculations to obtain shortest path trees [7], [8].

HQLIP realizes two-or-more-level hierarchy by introducing concepts of *area*s and *external links*. An *area* is an abstraction of a node or a managed domain. A network is divided into several layers by defining hierarchical *area*s. An external link is defined in order to notify the way to reach the external *area* from a certain *area*. The hierarchicalization of HQLIP can further suppress advertised routing information and accelerate the convergence time of route calculation. Therefore, HQLIP is adaptable to a large-scale network.

HQLIP is designed to construct topology information independently of address allocation and assignment.

Manuscript received May 22, 2015.
Manuscript revised January 21, 2016.
Manuscript publicized March 14, 2016.

[†]The authors are with National Institute of Information and Communications Technology, Koganei-shi, 184–8795 Japan.
[††]The author is with Tottori University, Tottori-shi, 680–8550 Japan.
[†††]The author is with Tokyo Institute of Technology, Tokyo, 152–8552 Japan.
a) E-mail: hudikaha@nict.go.jp

DOI: 10.1587/transinf.2015EDP7197

First, each node sends a message that includes the link information of its directly-connected links in flooding manner. Then, it constructs topology information from the received link information. If a topology is changed by adding/deleting a link or by a link failure, then each node receives new link information or notices that the link is not available by expiration of the link information. Therefore, it reconstructs the topology information. In HQLIP, address information is advertised separately from link information. Advertisement of address information does not affect the topology information that each node holds. Thus, HQLIP does not have to recompute shortest-path trees from topology information at the time of the occurrence of network address allocation/assignment including renumbering. For this reason, HQLIP achieves a fast routing convergence when renumbering occurs. P-NNI [9] is another link-state routing protocol that supports multi-level hierarchicalization. However it is designed for asynchronous transfer mode (ATM), and does not support IP networks or provide fast convergence in renumbering.

Two-or-more-level hierarchicalization provided by HQLIP reduces human operation when combining and stacking networks. Additionally, we design HQLIP to work with Hierarchical Automatic Number Allocation (HANA) protocol [10], [11]. This combination makes network reconfiguration flexible and fast. Renumbering of HANA is directly reflected to a fast convergence of HQLIP.

Furthermore, we propose a fixed-midfix renumbering (FMR) method to be employed to achieve a faster convergence. FMR is available only when HQLIP is synchronized with HANA. FMR updates the FIB in the kernel according to the address information obtained from HANA, before receiving the link-state information obtained from HQLIP. We show that the HQLIP and HQLIP-FMR is superior to OSPF and OSPFv3. Therefore, the combination of HQLIP and HANA realizes quickly-converging renumbering in a link-state protocol network.

In Sect. 2, we describe basic functions of HANA. In Sect. 3, we propose HQLIP for hierarchical routing. In Sect. 4, we propose an additional use of FMR in order to achieve a faster convergence. In Sect. 5, we focus on the implementations of HQLIP. In Sect. 6, we show a comparison of convergence time among HQLIP, HQLIP-FMR, OSPF and OSPFv3 at the time of the occurrence of renumbering. This proves that HQLIP and HQLIP-FMR are superior to OSPF and OSPFv3.

2. Basic Functions of HANA

We describe the locator structure and basic functions of HANA [11], which tightly work with HQLIP and HQLIP-FMR. HANA has the functions of DHCP/DHCPv6 [12], [13] and IPv6 router renumbering [14], and integrates them in a hierarchical manner. Renumbering generally requires much labor of network managers [15]. HANA's automated renumbering gets rid of such labor.

We also have developed a mechanism of automatically updating IPv4/IPv6 addresses allocated by HANA to the DNS system [16], but this is out of scope of this paper.

2.1 Locator Structure

We re-define an IP address that indicates a location as "a locator." Here, we define the upper, the middle, and the lower parts of the locator, as *prefix*, *midfix*, and *suffix*, respectively. Figure 1 shows the notations for indicating prefix length, midfix length and suffix length. Presume that a locator is N bit long. When the prefix length is n bit long, the length of the following midfix is $m - n$ bit long, and the length of the following suffix is N-m bit long. The ranges of them are described as $/n$, $/n$-m, and $/m$-N in the order of upper to lower. We employ the dotted-decimal address notation of IPv4.

2.2 Overview of the HANA Protocol

As shown in Fig. 2, basic principles of the HANA protocol are as follows:

1. A locator space is assigned to each of top-level ISPs by

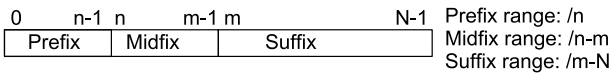


Fig. 1 Structure of a locator

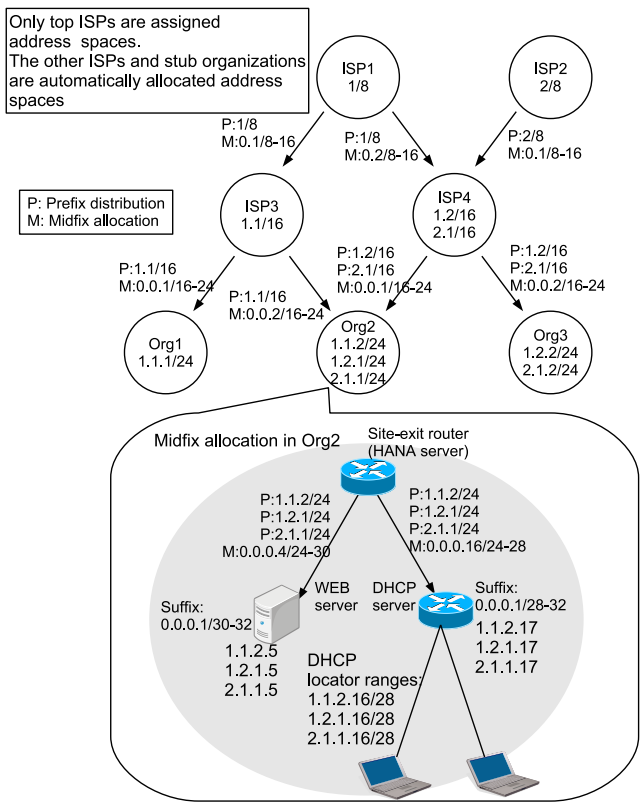


Fig. 2 Hierarchical locator allocation

a network manager. A locator space is a range of contiguous locators, and is described by the same notation of the prefix. Each of the top-level ISPs allocates parts of the assigned locator space to the lower-level ISPs directly-connected to it. The other ISPs except the top-level ISPs are free from setting actually-used locator spaces.
2. The upper-level ISP allocates a different midfix to each of the lower-level ISPs.
3. An upper-level ISP distributes the same prefix to all the lower-level ISPs directly-connected to it. Each of ISPs except top-level ISPs distributes the prefixes that are created by combining the distributed prefixes and the allocated midfix by the upper-level ISPs.
4. Locators of nodes in a stub organization are created by combining prefixes and the allocated midfix, and a suffix that is determined by the node itself.

We provide an example of hierarchical locator allocation by the HANA protocol. HANA can support unlimited multi-level hierarchical topology, but here we show the three-level topology as an example. In Fig. 2, seven ISPs and organizations are depicted. ISP1 and ISP2 are at the top level. ISP3 and ISP4 are at the second level. Org1, Org2, and Org3 are stub organizations at the third level. P: denotes a prefix that a prefixinfo message conveys. M: denotes a midfix that a midfixack message conveys.

As shown in Fig. 2, HANA allows an ISP as well as a stub organization to multihome ISPs at the upper level. For example, ISP4 is allocated two locator spaces of 1.2/16 and 2.1/16 by ISP1 and ISP2 that it is multihomed to. ISP4 distributes two prefixes of 1.2/16 and 2.1/16. Org2 receives the prefixes from ISP4, and also receives a prefix of 1.1/16 from ISP3. It is also allocated a midfix of 0.0.2/16-24 by ISP3 and a midfix of 0.0.1/16-24 by ISP4. As a result, Org2 are allocated three locator spaces of 1.1.2/24, 1.2.1/24 and 2.1.1/24. In this manner, ISPs and stub organizations are allocated multiple locator spaces.

On an internal network in each organization, midfixes and suffixes are locally allocated and assigned. In HANA, one node is defined as a HANA server by a network manager and the others become HANA clients in a certain managed domain.

For example, in Fig. 2, the site-exit router is defined as a HANA server in the Org2 network. It allocates midfixes of 0.0.0.4/24-30 and 0.0.0.16/24-28 to the WEB server and the DHCP server, respectively. In HANA, each client can request a different size of the locator space. The WEB server requests the midfix range of 24-30, and the DHCP server requests the midfix range of 24-28. The HANA server also distributes the prefixes of 1.1.2/24, 1.2.1/24 and 2.1.1/24 to the WEB server and DHCP server.

Suppose that the WEB server determine to use a suffix of 0.0.0.1/30-32. It can use locators of 1.1.2.5, 1.2.1.5, and 2.1.1.5, which are created by combining the prefixes, the midfix and the suffix. In the same way, if the DHCP server determines to use a suffix of 0.0.0.1/28-32, and uses

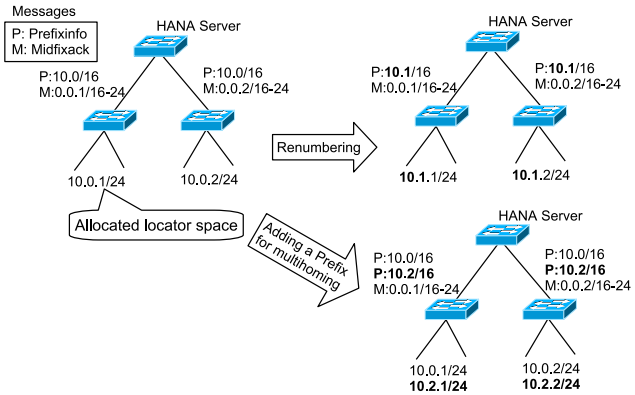


Fig. 3 Renumbering in a HANA network

locators of 1.1.2.17, 1.2.1.17, and 2.1.1.17. It can assigns locators from the ranges of 1.1.2.16/24, 1.2.1.16/24, and 2.1.1.16/24. to the lower DHCP clients.

These explained allocation and assignment of midfixes and suffixes are executed independently of the prefix distribution from outside of the organization. This enhances flexibility in renumbering.

2.3 Renumbering in HANA

When renumbering occurs, new prefixes are delivered. However, neither the midfix nor the suffix are changed in any node, even if the network adds or deletes a multihoming ISP.

In Fig. 3, there are a top L3 switch and two edge L3 switches in the network. The top L3 switch with the HANA server function allocates locator spaces of 10.0.1/24 and 10.0.2/24 to the two edge switches, respectively. When the top L3 switch changes the distributing prefix from 10.0/16 into 10.1/16 for renumbering, then two edge L3 switches newly receives a prefix of 10.1/16, and allocated to 10.1.1/24 and 10.1.2/24, respectively. Re-allocation of midfixes are not required.

For adding a prefix in addition to an already-distributed prefix, the top L3 switch distributes prefixes of 10.0/16 and 10.2/16 simultaneously. For example, the left-side edge switch is allocated to locator spaces of ‘10.0.1/24 and ‘10.2.1/24.

3. Proposal of Hierarchical Link-State Routing

We propose a hierarchical link-state routing protocol, which is a key function of HQLIP protocol. It supports multi-level hierarchical routing, and achieves the fast convergence time of the FIB in the case of renumbering. For hierarchical routing, we introduce two concepts that enable multi-level hierarchical routing. One is multi-level *areas*, and the other is *external links*. We describe how hierarchical routing is performed using these two concepts in the following sections.

Note that IPv4 or IPv6 addresses must be hierarchically allocated and assigned for hierarchical routing provided by

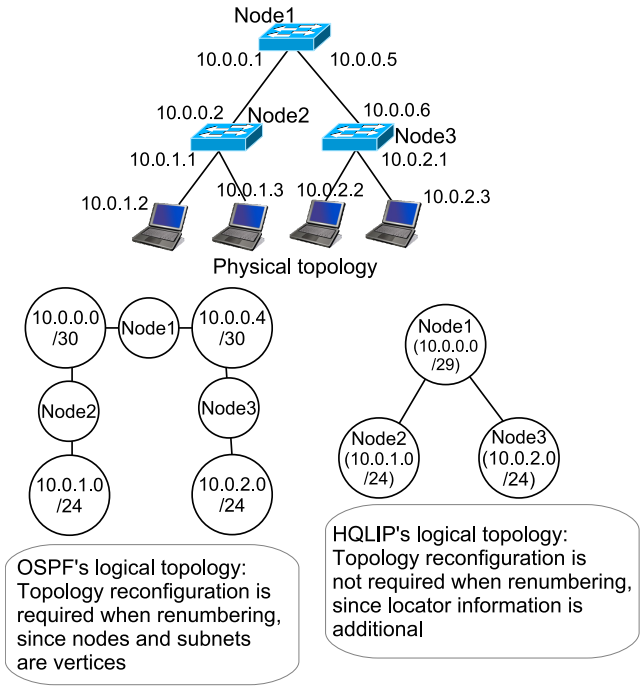


Fig. 4 Physical topology and OSPF/HQLIP logical topologies

HQLIP. A network manager can manually set up these addresses. However, employing HANA is easier for hierarchical addressing, and has the advantages of easy renumbering and fast convergence.

3.1 Creation of Topology Information

Figure 4 shows a physical topology and the corresponding OSPF and HQLIP logical topologies.

The OSPF for IPv4 regards a node and a subnet as a vertex, and a connection between a node and a subnet as an edge, in a graph that expresses a topology. The vertex is identified by the IP address of the node or the subnet. When renumbering occurs, the IP addresses (identifiers) are to be changed and the logical topology information must be reconfigured.

HQLIP regards a node as a vertex and a connection between nodes as an edge. The vertex is identified by its name. Locator information, which includes both of IPv4 and IPv6 addresses, is additional information attached to the vertex. Therefore, no matter how locators are changed, HQLIP achieves a fast routing information convergence as long as the topology remains unchanged.

Similarly, OSPFv3, which is designed for IPv6, also regards a node as vertex, and IPv6 address information is additional. However, it supports only two-level hierarchicalization, so cannot support a large-scale network. On the other hand, HQLIP provides multi-level hierarchicalization, and can supports a large scale network, which will be described in the next subsection.

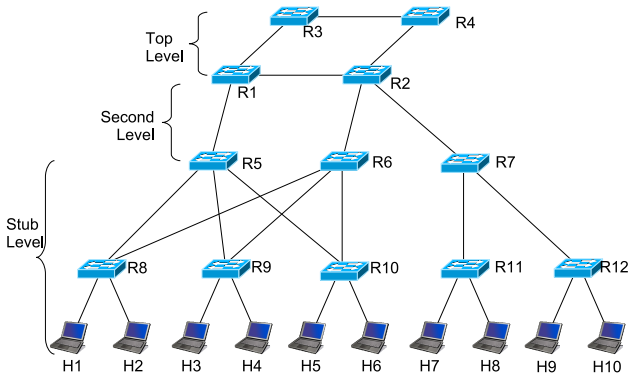


Fig. 5 Network topology

3.2 Hierarchicalization of *Area*s

HQLIP has features as follows:

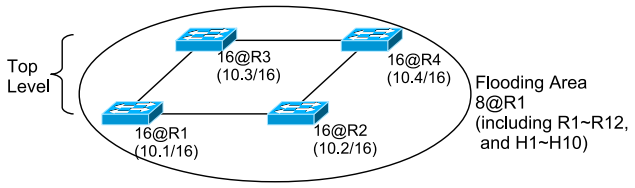
- An *area* is manually defined for expressing that some locator spaces become the destination in the *area*.
- A *flooding area* is manually defined for restricting routing messages to be advertised within a certain managed domain.
- HQLIP hierarchicalizes a network into multi-level networks by *area*s and *flooding area*s
- Two type of routing messages, an *area* information (*areainfo*) message and a link information (*linkinfo*) message are defined in order to advertise information of *area*s and links, respectively.
- An *external link* is defined in order to notify the way to reach the external *area* from a certain *area*.

Let us assume a network topology as shown in Fig. 5, and define three levels of hierarchy, the top level, the second level, and the stub level.

3.3 Top-Level *Area*

The top-level topology is constructed by four nodes as shown in Fig. 6. The top-level topology information must be advertised to all the nodes in Fig. 5. Thus, the top-level *flooding area* is defined as including all the nodes, e.g. R1∼R12 and H1∼H10.

Each of routers R1, R2, R3 and R4 advertises an *areainfo* message in the *flooding area*. An *areainfo* message is a vertex of the topology graph and consists of *area* ID and locator spaces. An *area* ID is identified by combination of the level number of hierarchy and the node name, e.g., "16@R1." The level number value of an upper level must be smaller than that of a lower level. Here, the level number is directly derived from the prefix length of the locator spaces, that is, "16." A *flooding area* restricts the advertising range, and is similarly described as an *area*, e.g, "8@R1." Here, R1 is a representative of the *flooding area* "8@R1." We will later explain the reason why a certain node must represent the *flooding area* in the explanation of



Area Information

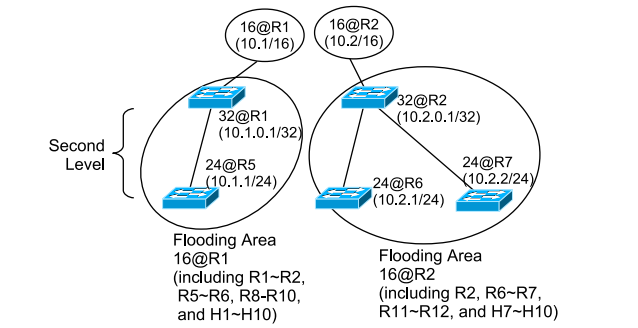
Area ID	Address Space
16@R1	10.1/16
16@R2	10.2/16
16@R3	10.3/16
16@R4	10.4/16

Link Information

Source Area ID	Destination Area ID	Metric
16@R1	16@R2	1
16@R1	16@R3	1
16@R2	16@R3	1
16@R2	16@R4	1
16@R3	16@R1	1
16@R3	16@R4	1
16@R4	16@R2	1
16@R4	16@R3	1

Source \ Destination	Link Information 16@R1	16@R2	16@R3	16@R4	Area Information
16@R1		1	1	1	10.1/16
16@R2	1		1	1	10.2/16
16@R3	1	1		1	10.3/16
16@R4	1	1	1		10.4/16

Fig. 6 Topology and messages in top-level *flooding area*



Source \ Destination	Link Information 32@R1	24@R5	16@R1	Area Information
32@R1		1	0.1	10.1.0.1/32
24@R5	1			10.1.1/24
16@R1				10.1/16

Source \ Destination	Link Information 32@R2	24@R6	24@R7	16@R2	Area Information
32@R2		1	1	0.1	10.2.0.1/32
24@R6	1				10.2.1/24
24@R7	1				10.2.2/24
16@R2					10.2/16

Fig. 7 Topology and messages in second-level *flooding area*

the second level.

Each node also advertises *linkinfo* messages, which are edge information of the topology graph. A *linkinfo* message consists of a source *area* ID, a destination *area* ID, and a metric. In the example network, all the metric of physical line is defined as '1' for simplicity.

The tables of *linkinfo* and *areainfo* messages are described in Fig. 6. The bottom-side table is built by combining two top-side tables into one, and solely described in the following figures, Fig. 7 and Fig. 8. In the bottom-side table, a cell between unlinked edges are written as blank. A cell with a diagonal line means that the *linkinfo* message cannot be defined between the edges under the HQLIP protocol.

As a result, all the node in the *flooding area*, e.g., all the nodes depicted in Fig. 5 share the advertised *linkinfo* and *areainfo* messages in the table.

3.4 Second-Level *Area*

In the second level, two *flooding area*s are defined, as shown in Fig. 7. These *flooding area*s are individually managed and do not exchange *areainfo*/*linkinfo* messages to each other. In the *flooding area* of "16@R1," R5 advertises an *areainfo* message of "24@R5," which includes the lower locator space information of "10.1.1/24." On the other hand, R1 advertises an *areainfo* message of "32@R1," which includes the locator information of its own "10.1.0.1/32."

HQLIP allows to advertise a *linkinfo* message directed to an *area* that resides outside the *flooding area* in order to notify the way to reach the *area*. We call this link *external link*. For example, R1 advertises a *linkinfo* message of "source:32@R1 destination:16@R1 metric:0.1" within the *flooding area* of "16@R1." The metric is defined as the extremely small value of "0.1," since this link is a virtual one in R1. The *area* of "16@R1" in Fig. 7 and the *area* of "16@R1" in Fig. 6 has the same *area* ID. HQLIP constructs hierarchical logical topology by regarding that they are identical.

The *flooding area* of "16@R1" in Fig. 7 also has the same *area* ID. However, this information is not used for constructing the hierarchical logical topology. This is the information for a network manager to recognize the hierarchical relationship. That is, R1 is a representative of the *area* of "16@R1" and advertises *areainfo* messages to the other nodes in the *flooding area* of "8@R1." R1 is also a representative of the *flooding area* of "16@R1" and local *areainfo*/*linkinfo* messages are exchanged inside the *flooding area* of "16@R1."

3.5 Stub-Level *Area*

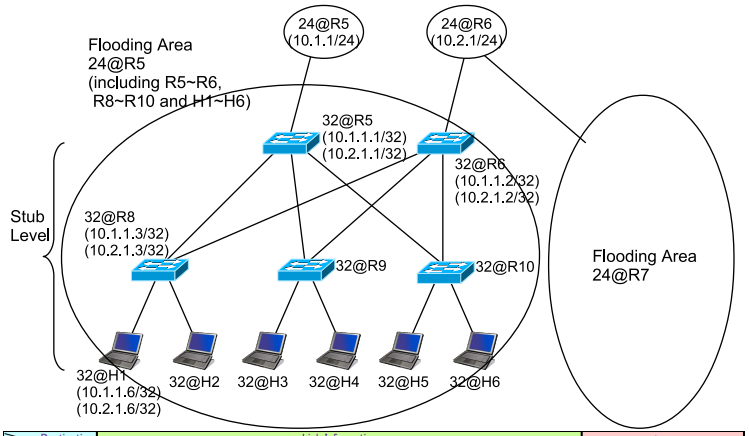
In the stub level, two *flooding area*s are defined, as shown in Fig. 8. The internal topology of the *flooding area* of "24@R7" is omitted. Most of the locator information are also omitted, so refer to the table for the details. The left-side *flooding area* is multihomed to the two second level *area*s, and allocated two locator spaces of "10.1.1/24," and "10.2.1/24." Thus, each of the nodes in the *flooding area* has two locators.

Multi-level hierarchicalization suppresses topology information that each node has to manage. For instance, in the top level, the lower topology information is not advertised. In the lower level, topology information of its own and the upper levels are advertised, e.g., R8 maintains the information of the *flooding area*s of "8@R1," "16@R1," and "16@R2."

3.6 Example of Hierarchicalized Route Selection

We explain hierarhicalized route selection in the network by showing a logical topology from the viewpoint of R7 (Fig. 9).

Suppose that R7 calculates routes from R7 to H1.



Source \ Destination	32@R5	32@R6	32@R8	32@R9	32@R10	H1	H2	H3	H4	H5	H6	24@R5	24@R6	Area Information
32@R5			1	1	1							0.1		10.1.1.1/32 , 10.2.1.1/32
32@R6			1	1	1								0.1	10.1.1.2/32 , 10.2.1.2/32
32@R8	1	1				1	1							10.1.1.3/32 , 10.2.1.3/32
32@R9	1	1						1	1					10.1.1.4/32 , 10.2.1.4/32
32@R10	1	1								1	1			10.1.1.5/32 , 10.2.1.5/32
32@H1			1											10.1.1.6/32 , 10.2.1.6/32
32@H2			1											10.1.1.7/32 , 10.2.1.7/32
32@H3				1										10.1.1.8/32 , 10.2.1.8/32
32@H4				1										10.1.1.9/32 , 10.2.1.9/32
32@H5					1									10.1.1.10/32 , 10.2.1.10/32
32@H6					1									10.1.1.11/32 , 10.2.1.11/32
24@R5														10.1.1/24
24@R6														10.2.1/24

Fig. 8 Topology and messages in stub-level *flooding area*

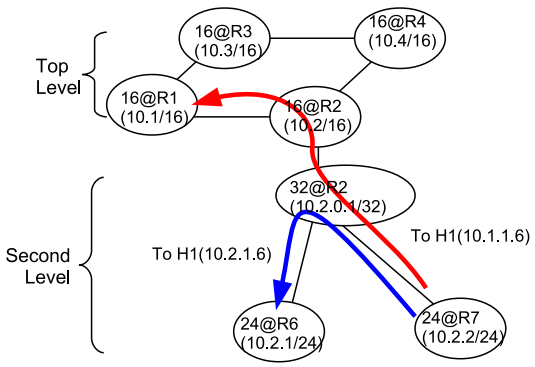


Fig. 9 Logical topology and routes from the viewpoint of R7

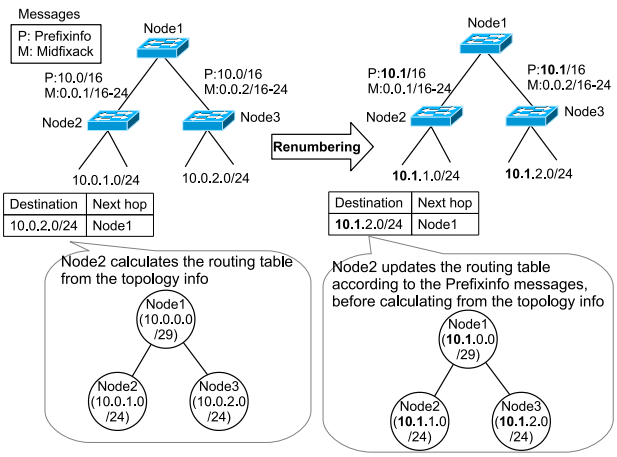


Fig. 10 Fixed-Midfix Renumbering

H1 has the multiple addresses of 10.1.1.6 and 10.2.1.6 in Fig. 8. R7 obtains the *areainfo*/*linkinfo* messages of the table in Fig. 6 and the right-side table in Fig. 7. The best-matching *areainfo* messages to 10.1.1.6 and to 10.2.1.6 are 16@R1(10.1/16) and 24@R6(10.2.1/24), respectively. Note that R7 does not obtain the left-side table in Fig. 7, so that 24@R5(10.1.1/24) is not selectable. To reach the destinations, R7 selects the routes of R7→R2→R1, and R7→R2→R6, respectively, as shown in Fig. 9. R7 does not have to calculate routes beyond R5 and R6 because of hierarchicalization. R5, R6 and the routers beyond them calculates the rest of the routes.

HQLIP is able to hierarchicalize a network into any number of levels, unlike OSPF/OSPFv3 or IS-IS. The multi-level hierarchicalization reduces route calculation at each node.

4. Proposal of Fixed-Midfix Renumbering in HQLIP

As we have already mentioned in Sect. 3, HQLIP exceeds OSPF technically where the topology remains unchanged even if renumbering occurs. Furthermore, we propose a fixed-midfix renumbering (FMR) method to be employed additionally in order to accelerate the convergence of the FIB. FMR utilizes the HANA renumbering mechanism.

By using a HANA feature that a midfix is fixed, we accelerate the convergence of the FIB when renumbering occurs. That is, HQLIP updates the FIB soon after receiving prefix information by the HANA protocol (Fig. 10). Note that this method is only applicable when all the nodes in the

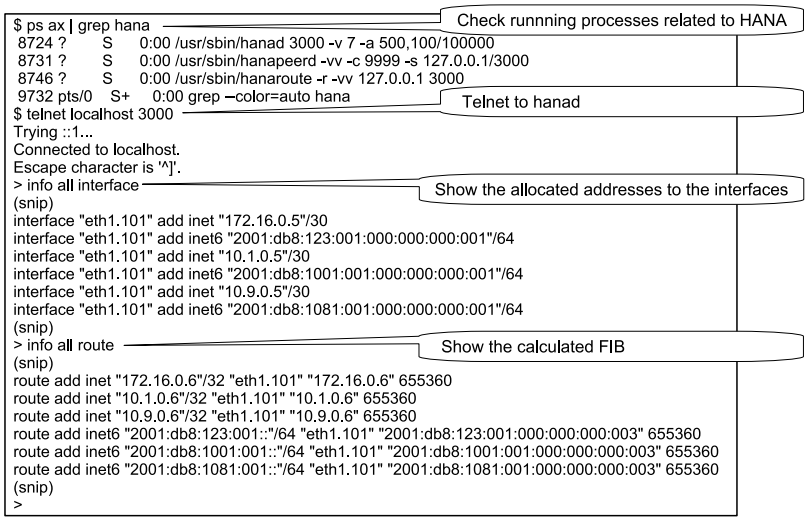


Fig. 12 HANA/HQLIP processes on PC/Linux

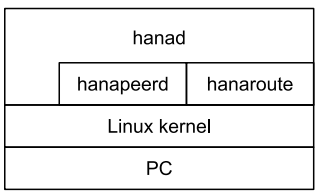


Fig. 11 HANA/HQLIP processes on Linux OS

network runs HANA and HQLIP protocols.

The procedures of FMR is as follows:

1. A node updates the FIB in the kernel according to the prefix information received under the HANA protocol, when the prefix information is changed. It assumes that the midfixes of all the nodes are not altered.
2. It receives updated *areainfo* messages created by the other nodes, and re-calculates the FIB according to the messages. Note that *linkinfo* messages are not changed in a condition of renumbering.
3. After the re-calculated FIB becomes stable, the node writes re-calculated FIB into the kernel. The result of this re-calculation matches the FIB created at (1), as long as the network topology is not changed. Therefore, the updates of the FIB was completed at (1).

Consequently, FMR completes FIB updates according to the prefix information obtained from HANA messages before receiving HQLIP messages. HQLIP with FMR quickly stabilizes the FIB in the link-state routing protocol network.

5. Implementation of HQLIP to HANA

We implemented the HQLIP function to the HANA software suite on Linux OS (Ubuntu 10.04 LTS). HANA/HQLIP software suite consists of three processes, called *hanad*, *hanapeerd* and *hanaroute* (Fig. 11). Each PC becomes a HANA/HQLIP-supported node when executing the whole process. If a PC has multiple interfaces, it functions as a HANA/HQLIP router.

hanad

A *hanad* is the main process of a set of processes. It can run both of the HANA and HQLIP protocols. A *hanad* connects to the *hanad*s in other adjacent HANA/HQLIP nodes by TCP connections, and exchanges HANA/HQLIP messages. It is controllable via command line interface (CLI) established by a TCP connection. The CLI is used by a network manager and the other processes.

hanapeerd

A *hanapeerd* seeks adjacent HANA/HQLIP nodes by using an IPv6 link-local multicast address and informs a *hanad* in the same node of the adjacent nodes via the *hanad* CLI.

hanaroute

A *hanaroute* obtains locators and route information from *hanad* using the CLI of *hanad*, and reflects them to the kernel of the node. That is, a *hanaroute* sets addresses of the interfaces and next hop information of the FIB of the node.

Figure 12 shows that the HANA/HQLIP processes are running on PC/Linux (Ubuntu 10.04LTS). The PC/Linux are running as a router with four interfaces. Three IPv4 addresses and three IPv6 addresses are assigned to each interface.

6. Experiments

We constructed a logical network that consists of 22 logical L3 switches. It emulates a data center as shown in Fig. 13. The physical network for the emulation consists of seven Linux PCs and a single gigabit ethernet

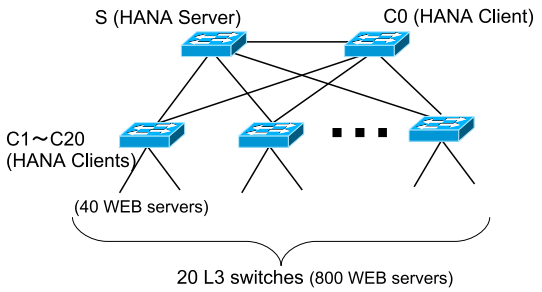


Fig. 13 Experimental network

(GbE) switch. Each PC has Intel Xeon 8-core CPU (2.93GHz), 8 Gbytes memory and one GbE interface. All the GbE interfaces of the PCs are connected to a single GbE switch. We measure the convergence times of HQLIP, OSPF and OSPFv3 on the network. Each logical L3 switch consists of *hanad*/*hanapeerd*/*hanaroute* that provides the HANA/HQLIP protocols and OSPF processes, zebra and ospfd. Zebra and ospfd are a part of Quagga routing suite [17]. These processes run on seven Linux PCs. The processes of logical L3 switch S run on one PC, and those of switch C0 run on another PC. The processes of switches C1–C20 run on the rest of five PCs. The logical links among the switches are constructed using VLANs.

The top two routers are supposed to be connected to the upper network. S is one of the top two routers, and is a HANA server. C0 is the other one, but a HANA client. From C1 to C20, they are 20 HANA clients. They are also HANA servers for the lower terminals at the same time. Each of C1–C20 is supposed to be connected 40 terminals that are WEB servers. It requests /24(IPv4) and /56(IPv6) locator spaces. There are 800 WEB servers in the whole network. In terms of a network topology for link state protocols, the 22 switches are the vertices in cases of HANA and OSPFv3, and the 22 switches and the 61 subnets are the vertices in case of OSPF.

Locators are allocated by HANA in either case where HQLIP, OSPF or OSPFv3 is used as a link-state routing protocol.

First, we run *hanad* on each switch. It allocates locators to each node (switch or WEB server). Then, we try HQLIP, HQLIP with FMR (HQLIP-FMR), OSPF (IPv4), and OSPFv3 (IPv6), to start with allocated locators one at a time. Here, HQLIP delivers IPv4 and IPv6 addresses simultaneously.

By changing the configuration of S, the prefix information is changed and distributed from S in the network. Each node updates the FIB in the kernel according to HQLIP, HQLIP-FMR, OSPF or OSPFv3. Then we measure the prefix distribution time and the convergence time of the FIB. The prefix distribution time is measured from the time when the HANA server sends prefix information messages to the clients to the time when the all the clients receives the messages. The convergence time of the FIB is measured from the time when all the clients receives the messages to the time when the FIBs of all the clients become stable.

Table 1 Experimental results of 22 switch network

	HQLIP	HQLIP-FMR	OSPF	OSPFv3
Prefixinfo distribution time (sec)	0.78	same to the left		
Convergence time (sec)	0.68	0.01	9.96	4.45

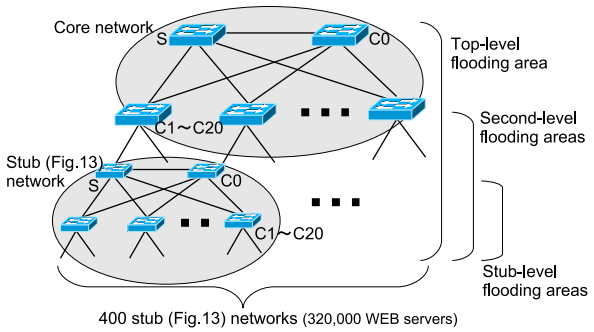


Fig. 14 Stacked experimental network

Table 1 shows the averages of ten times experiments. From Table 1, the convergence time of the HQLIP is 0.68 seconds, and is smaller than those of OSPF and OSPFv3, 9.96 and 4.45 seconds. In addition, the convergence time of HQLIP-FMR is 0.01 seconds and is further faster than the other methods, HQLIP, OSPF and OSPFv3. Even if the prefix distribution time of 0.78 seconds is added to, the convergence time of HQLIP-FMR is less than one second.

We adapt the experimental results to a larger HQLIP network. As shown in Fig. 14, we can combine and stack networks easily, since both of HQLIP and HANA support two-or-more-level hierarchy. Figure 14 shows the stacked network. First, we prepare a core network that has the same L3 switch topology of Fig. 13. Then, we connect Fig. 13 networks to the core network as stub networks. Switches S and C0 in a stub Fig. 13 network are connected to different edge switches of the core network for fault tolerance. 400 stub networks can be connected to the core network instead of 800 WEB servers. Switches S and C0 in a stub network become HANA clients and receive different prefixes. We define Top-level, second-level, and stub-level *flooding area*s, as shown in Fig. 14. 320,000 (800×400) WEB servers can be connected to the edge switches of the stub networks.

According to Fig. 14, we configured a stacked network that consists of one core network and one stub network. We run the HANA/HQLIP processes on the stacked network, and measured prefix distribution times and convergence times of the core and stub networks (Table 2). Table 2 shows the averages of ten times experiments. We did not run OSPF on this network, because OSPF supports only two-level hierarchy at a maximum. We employed the seven Linux PCs. The processes of logical L3 switches S's on the core/stub networks run on one PC, and those of switches C0s on the two networks run on another PC. The processes of switches C1–C20 on the two networks run on the rest of five PCs.

When we change the prefix setting in switch S in the core network, all the locators of the nodes are changed in the

Table 2 Experimental results of stacked network

	Core network		Stub network		
Method (FIB type)	HQLIP	HQLIP-FMR	HQLIP	HQLIP-FMR (FIB toward external nodes)	HQLIP-FMR (FIB toward internal nodes)
Prefix distribution time (sec)	1.22		2.11		
Convergence time (sec)	0.85	0.01	1.20		0.01

whole network. The convergence time of the core network is supposed to be the same to that of Table 1, since the locator changes at the nodes in the stub networks are completely hidden to the core network, as mentioned in Sect. 3.2. The experimental result of the convergence time was a little longer than that of Table 1. This seems to have occurred since each PC emulates more switches than the previous experiment. As for a stub network, in case of HQLIP, the convergence time is longer than the Table 1 results, since nodes in a stub network has to manage the topologies of the top-level, second-level and stub-level *flooding area*s. In case of HQLIP-FMR, the convergence time of FIB toward the external nodes is longer than that of Table 1 because of the same reason of the HQLIP case. However, the convergence time of FIB toward the nodes in the stub network is the same to that of Table 1, since the convergence time is determined only by the receiving time of prefix information.

As a result, HQLIP-FMR achieves the faster convergence time of link-state routing protocol when renumbering occurs. We can also run the hierarchical links-state routing in HQLIP with more than two-level hierarchy on the stacked network. These contribute construction of the stable and scalable networks with link-state routing protocols.

7. Conclusions

The proposed hierarchical link-state routing in HQLIP easily scales up the network by combining and stacking configured networks. It also achieves fast convergence of the FIB without reconfiguring topology information in case of renumbering in the network. In addition, we propose a fixed-midfix renumbering (FMR) method that accelerates convergence time even faster. FMR is available when HQLIP is employed with HANA at the same time. We showed that HQLIP with FMR is superior to OSPF in convergence time. The combination of HQLIP and HANA enables quickly converging renumbering in hierarchical networks.

Regarding the framework for future action, we will implement HQLIP-FMR function into L3 switches. Then we will yield practical applications of HANA/HQLIP-FMR in enterprise networks and data centers.

Acknowledgments

We would like to express our thanks to Yoichi Koyama, Trans New Technology Inc. for his valuable insights and strong supports.

References

[1] J. Moy, "OSPF version 2," RFC 2328, April 1998.
[2] R. Coltun, D. Ferguson, and J. Moy, "OSPF for IPv6," RFC 2740, Dec. 1999.
[3] D. Oran, "OSI IS-IS intra-domain routing protocol," RFC 1142, Feb. 1990.
[4] M. Al-Fares, A. Loukissas, and A. Vahdat, "A scalable, commodity data center network architecture," SIGCOMM Comput. Commun. Rev., vol.38, no.4, pp.63–74, Aug. 2008.
[5] R.N. Mysore, A. Pamboris, N. Farrington, N. Huang, P. Miri, S. Radhakrishnan, V. Subramanya, and A. Vahdat, "PortLand: A scalable fault-tolerant layer 2 data center network fabric," pp.39–50, Aug. 2009.
[6] J. Mudigonda, P. Yalagandula, M. Al-Fares, and J.C. Mogul, "SPAIN: COTS data-center ethernet for multipathing over arbitrary topologies," Proceedings of the 7th USENIX Conference on Networked Systems Design and Implementation (NSID), 2010.
[7] A. Basu and J. Riecke, "Stability issues in OSPF routing," SIGCOMM Comput. Commun. Rev., vol.31, no.4, pp.225–236, Aug. 2001.
[8] P. Francois, C. Filsfils, J. Evans, and O. Bonaventure, "Achieving sub-second IGP convergence in large ip networks," SIGCOMM Comput. Commun. Rev., vol.35, no.3, pp.35–44, July 2005.
[9] ATM Forum, "Private network-network interface specification v1.0," March 1996.
[10] K. Fujikawa, H. Harai, and M. Ohta, "The basic procedures of hierarchical automatic locator number allocation protocol HANA," Asia Workshop on Future Internet Technologies (AWFIT2011), pp.124–131, Nov. 2011.
[11] K. Fujikawa, H. Tazaki, and H. Harai, "Inter-AS locator allocation of hierarchical automatic number allocation in a 10, 000-AS network," 2012 IEEE/IPSJ 12th International Symposium on Applications and the Internet (SAINT2012), pp.68–73, July 2012.
[12] R. Droms, "Dynamic host configuration protocol," RFC 2131, March 1997.
[13] O. Troan and R. Droms, "IPv6 prefix options for dynamic host configuration protocol (DHCP) version 6," RFC 4423, Dec. 2003.
[14] M. Crawford, "Router renumbering for IPv6," RFC 2894, Aug. 2000.
[15] D. Karrenberg, "PI vs PA address space," RIPE 127, ftp://ftp.ripe.net/ripe/docs/ripe-127.txt, May 1995.
[16] K. Fujikawa and Y. Jin, "Extensions of hierarchical/automatic locator number allocation protocol HANA for DNS cooperation," vol.112, no.250, pp.101–105, Oct. 2012.
[17] Quagga Routing Suite, http://www.nongnu.org/quagga/.

Kenji Fujikawa received the M.E. and Ph.D. degrees in Informatics, Kyoto University, Japan, in 1995 and 2000, respectively. After completing graduate school, became Assistant Professor in the Graduate School of Informatics, Kyoto University in 1997, Senior Researcher at ROOT Inc. in 2006, and joined National Institute of Information and Communications Technology in 2008. His research topic is hierarchical routing and autoconfiguration of network. He is a member of IEICE, IPSJ and IEEE.

Hiroaki Harai received the M.E. and Ph.D. degrees in information and computer sciences from Osaka University, Osaka, Japan, in 1995 and 1998, respectively. In April 1998, he joined Communications Research Laboratory (CRL), Tokyo, Japan. He is currently a Director of Network Architecture Laboratory, National Institute of Information and Communications Technology (NICT, formerly CRL), Tokyo, Japan. His current research topic is in the ares of future networks and optical networks. He is a member of IEICE and IEEE.

Motoyuki Ohmori received the B.S. and M.S. degrees in Computer Science and Communication Engineering from Kyushu University in 1999 and 2001, respectively. He had been a lecturer in Faculty of Literature, Chikushi Jogakuen University from 2004 to 2013, and has been an associate professor in Center for Information Infrastructure and Multimedia, Tottori University since 2013. His research interests include network architecture, multicasting, routing, mobile networks and energy efficient network management. He is a member of the IPSJ, IEICE, JSSST, IEEE and ACM.

Masataka Ohta received the B.S. and M.S. degrees in Information Science from University of Tokyo in 1982 and 1984, respectively and Ph.D. on “High Quality Computer Graphics” from Tokyo Institute of Technology in 1994. He is a lecturer of Tokyo Institute of Technology. His research area includes Computer Graphics, High Performance Computing, UNIX and Networking. He is a member of IEICE, IPSJ and IEEE.