Design of Flexible Layer-3 Routing Protocol with Variable-Length Address Information and Its Implementation

YASUHIRO SATO
Japan Coast Guard Academy
Faculty of Maritime Safety Technology
5–1 Wakaga, Kure, Hiroshima
JAPAN
sato@jcga.ac.jp

YUSUKE TOJI, SHINGO ATA, and IKUO OKA
Osaka City University
Graduate School of Engineering
3–3–138 Sugimoto, Sumiyoshi, Osaka
JAPAN
{toji@n.,ata@,oka@}info.eng.osaka-cu.ac.jp

Abstract: In next-generation network architectures, the number of nodes and equipment connected to information networks increase more than ever. Due to the explosive increase of network equipment, an information retrieval technique to access information surely and efficiently will be strongly required. For this, one of the current approaches is intelligent routing, such as DHT, which is implemented on upper layers than IP layer. However, the inefficiency in processing packets and the mutual interference between layers are caused, because similar functions are implemented on different layers as different routing protocols. In this paper, we propose a new layer-3 routing protocol that can achieve flexible information retrieval implemented on overlay networks. For this purpose, we design a new routing protocol that can use variable-length address information, such as keyword or device name, as its address information. Moreover, we consider the feasibility of our routing protocol by implementing it to GNU Zebra as an extension of BGP.

Key–Words: Next-generation network, routing protocol, overlay routing, clean-slate architecture

1 Introduction

In next-generation network architectures, the number of nodes and equipment, such as PCs, phones and home electronics, connected to networks increase more than ever. Moreover, the amount of information treated on networks will increase explosively, and each of network equipment will generate various types of traffic, such as text messages, photos, and videos. To access information efficiently, information retrieval techniques based on not only IP address but also semantic information, such as keyword, contents type and other content information, have been considered. Some of the current approaches to achieve this purpose are intelligent routing algorithms by using DNS (Domain Name Service) or DHT (Distributed Hash Table) [1][2] etc. These techniques are implemented as one of overlay networks to treat domain names of network node or keywords of contents as one of address information. Although flexibility and efficiency of information retrieval are achieved by intelligent routing, end nodes in the current Internet must eventually use IP address as layer-3 address information to communicate each other. This is because most of routers on the current network architecture are specialized and optimized for IP networks, and these routers can recognize only IP address as address information.

Intelligent routing protocols implemented on overlay networks have some differences in detailed functions of each protocol, which include calculation function of path metrics. However, main functions of routing protocols such as exchanging routing information and forwarding packets are slightly different. Namely, in the current approaches, similar functions of each routing protocol are implemented on different layers as different routing protocols, although most of the functions are overlapping. Decentralization of the equivalent functions causes a lot of problems such as the inefficiency in packet processing and the mutual interference between layers [3][4] etc. For example, the amount of redundant traffic and transmission latency increase due to the inconsistency between the physical topology and a logical topology of overlay networks. Moreover, the instability of networks may be caused, because the existence of node is not guaranteed in overlay networks.

Therefore, realization of a layer-3 routing protocol with flexibility that has been previously implemented on overlay networks is important to overcome the problems described above. Our research group has been described a brief view of a new layer-3 routing mechanism in [5]. In this paper, we propose a new layer-3 routing mechanism that can achieve flexible information retrieval. Our main idea is that

variable-length information is utilized as address information in our routing protocol, instead of IP address. We assume that variable-length information is device names, domain names, and keywords of contents etc. For this, we first abstract the common routing functions from existing routing protocols that have been implemented on layer-3 or the upper layers so far. Second, we describe the differences of the functions among the existing routing protocols, and then design an abstraction model that can become the fundamental model of various routing protocols. Figure 1 shows the conceptual diagram of our abstraction routing model. Finally, in order to consider the feasibility of our abstraction model, we implement a new routing protocol as an extension of BGP (Border Gateway Protocol) [6]. This is because it is hard to migrate from IP to our proposed protocol directly in the current Internet. We consider that our protocol is accepted easily by implementing it as an extension of BGP, which is one of widely used routing protocols. We call the protocol *BGPGA* (BGP with Generalized Address) throughout our paper. Our implementation in this paper is the first step to achieve a flexible routing on layer-3, BGPGA. Our routing protocol enables routers to treat intelligent routing, and then the routers can select a communication path according to semantic information. This function reduces routing process at client machine and redundant communication latency.

The rest of this paper is organized as follows. In Section 2, we describe the design guideline of routing abstraction for BGPGA. Furthermore, we describe the concept of our proposed architecture in Section 3. The abstraction model we propose is described in Section 4 and we design BGPGA based on the abstraction model in Section 5. Moreover, the implementation of BGPGA and verification of its behavior are shown in Section 6. Finally, we conclude our paper and discuss plans for future topics.

2 The design of BGPGA

In this section, we describe the design guideline of BGPGA, while considering problems in the current Internet. For this, we point out some problems relevant to usage of IP address in the current IP network architecture.

2.1 Problems of IP architecture

The IPv4 address depletion has been a concern since the Internet started to diffuse explosively. Although address conservation techniques, such as NAT (Network Address Translation), are adopted to deal with

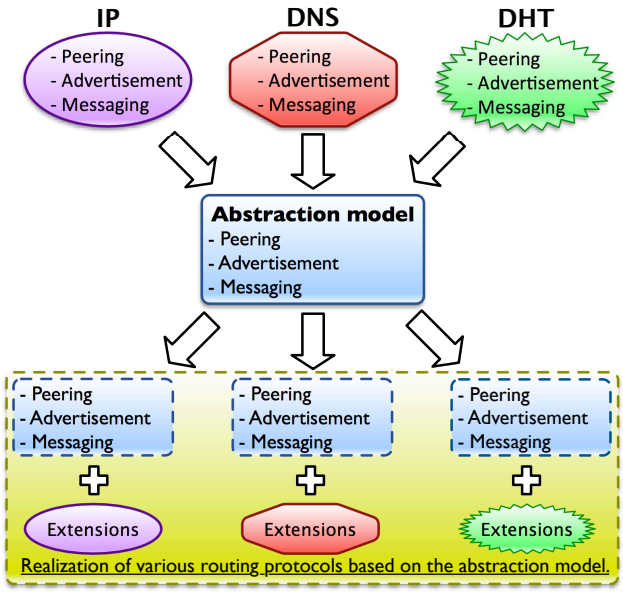


Figure 1: Concept of our abstraction model

the depletion, the end-to-end transparency of the Internet is considerably compromised. To overcome this problem fundamentally, the address length is extended to 128 bits to treat larger address space, which is implemented in IPv6 [7].

However, using a single IP address as both node identifier and location identifier causes some serious problems. In a mobility scenario, when the location of a mobile device is changed, we cannot use IP address as a node identifier. In a multi-homing scenario, IP address that cannot be aggregated has to add to the routing table of routers as new routing entry. As a result, the number of routing entries increases vastly, and it becomes hard for routers to manage the routing entries. To solve these problems, LINA (Location Independent Network Architecture) [8], HIP (Host Identity Protocol) [9], Six/One [10], LISP (Locator Identifier Separation Protocol) [11], and MILSA (Mobility and Multihoming supporting Identifier Locator Split Architecture) [12] have been studied so far. Specifically, authors in [13] considered a mapping system between address information and location information for Internet routing to split host identifier and its locator. In Mobile IP [14], a care-of address is used to identify where a mobile device is in. However, these techniques assume that IP address is essentially used to communicate between end nodes.

For the reasons stated above, the current IP should be redesigned. In fact, the new network architecture is being researched by various research projects such as FIND (Future Internet Design) [15] and FP7 (Seventh Framework Programme) [16].

2.2 Routing by variable-length address

To achieve flexible and efficient information access, it is necessary to treat not only numerical characters such as IP address but also semantic information such as text, domain name, and keyword.

However, the conventional IP routers are designed simply and can process only IP packets. Therefore, information search based on the semantic information are achieved on overlay networks. In order to achieve such information search on layer-3, a new layer-3 protocol that can recognize the semantic information as address information is required. Most of domain names and keywords that are currently used in the Internet are variable-length information. Therefore, the address of host or contents should be variable-length information, and a new routing protocol should identify host or contents uniquely by using the variable-length information. Moreover, we need to modify forwarding functions of the routers to treat such variable-length information.

2.3 Abstraction of routing functions

In the current Internet, various kinds of intelligent routings are performed on overlay networks. Although these intelligent routing protocols have many similar functions, the functions have been implemented on different layers as different routing protocols.

Functions of most routing protocols in upper layers are constructed by three main functions: (1) registering nodes; (2) formulating and distributing advertised information; and (3) forwarding messages. Although the detailed particulars of three functions differ in each protocol, routing processes in these functions differ only slightly. Decentralizing such equivalent functions causes a lot of problems such as the inefficiency of packet processing and the mutual interference between layers in the current network. Specifically, the amount of traffic and the latency increase due to a mismatch between the physical topology and the logical topology. In addition, the network structure becomes unstable, because end nodes participating to overlay network join and leave frequently.

The individualized design of these routing functions will make implementation costs and redundant processes increase. In next-generation networks, greater diversity of routing functions that can adapt to various demands from users will be required. Consequently, the router architecture will have to be able to support various routing functions flexibly. Hence, it is necessary to perform abstraction of the fundamental routing functions from existing protocols in layer-3 or overlay networks. Moreover, specific routing functions in each protocol should be implemented by an extension of the abstraction model.

2.4 Distributed management of routing information

The routers of our architecture can forward packets according to variable-length information as address information. The information of which the routers should manage increases drastically. In IPv4, the amount of BGP routing information is approximately 400,000 entries (as of 2012 [17]). For instance, in case of retaining a FQDN (Fully Qualified Domain Name) as address information, the number of these FQDNs was 9 hundred million entries as of July 2012 [18]. Namely, the amount of the semantic information such as FQDN, which is treated as address information of our protocol, is extremely huge. To handle the huge amount of routing information, we consider distributing the routing information uniformly to all routers. In the Internet, routing protocols can be distinguished into two major categories: IGP (Internal Gateway Protocol) and EGP (External Gateway Protocol). In the viewpoint of distributing uniformly in the whole network, distributing the information to routers located in various autonomous systems (ASes) is more efficient than doing so in intra-ASes. In other words, the routing protocol that can exchange variable-length information between ASes is required. Although scalability of our protocol is one of important issues, we especially treat design of the abstraction model and its implementation in this paper. We plan to focus on scalability of our protocol in our future work.

2.5 Feasibility

Even if a new architecture is superior, it is not always true that the architecture is accepted as the infrastructure of the Internet. We give careful consideration on the feasibility of the router architecture as an infrastructure for the Internet. As for the routing protocol, we need to consider the following points.

1. Compatibility with existing routing protocols
2. Saving the implementation costs by reducing modifications for traditional routers

2.6 Summary of our design guideline

We summarize the design guideline for a new layer-3 protocol we described in this section. We propose a new layer-3 protocol that has the following characteristics and concepts.

1. The address information is variable-length information.
2. Various protocols can be realized by our protocol.
3. Our protocol can perform inter-domain routing.
4. Compatibility with existing routing protocols is guaranteed.
5. The implementation cost should be low.

3 Concept of our architecture

In this section, we describe the network architecture and the router architecture that realize a routing algorithm according to variable-length information.

3.1 The architectures of our router and FIB

The routers in our architecture have to process the packets with variable-length information. The variable-length information is various kinds of semantic information including keyword of contents, domain name, device name, and phone number etc. The routers have to manage these information in order to forward packets to appropriate hosts. In existing approaches, to treat different types of address information, each type of information is managed in different FIBs (Forwarding Information Base) according to its information type. The routers in our architecture do not have an individual FIB in each protocol but just a single routing table. Thus, address information on each routing protocol is stored in the single FIB. To identify address information in each protocol, we define *namespace* in the FIB. Address information of the same namespace is grouped, and the group means the same routing protocol.

Figure 2 shows our router architecture and concept of searching a matching entry. When a packet arrives at a router, the router extracts the header of the packet. The address field included in the header is encoded as the destination address with namespace. In this paper, we describe address information with namespace as "**Namespace:Address**" (e.g. "DHT:abcde.txt"). The router first searches corresponding entries of FIB that are the same namespace of arrived packet. After looking it up, if there is a corresponding entry, the router forwards the packet to the next router in accordance with the next hop field in the routing table where the router manages. Generally, the interface of the router where the packet should be forwarded is stored in the next hop field. The interface is not only a physical interface ID but also a logical interface ID. For example, a router can

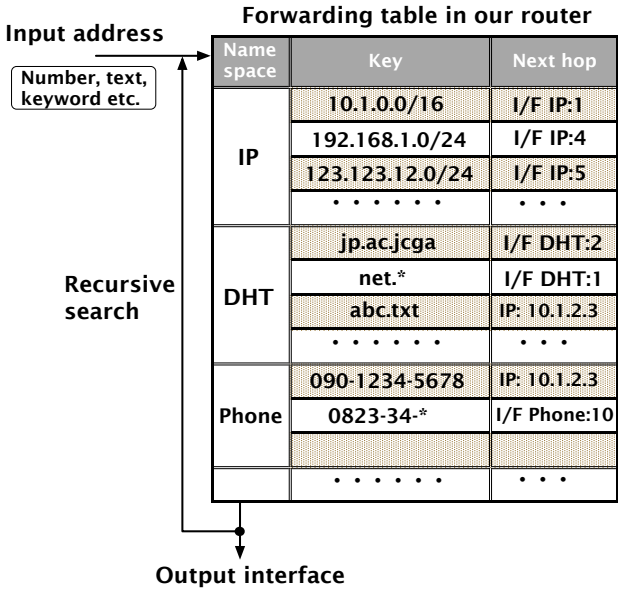


Figure 2: Concept of entry search in our router

store "IP:10.1.2.3" in the next hop field as a result of looking up "Phone:090-1234-5678" denoted in Fig. 2. In this case, the router searches for "IP:10.1.2.3" in the FIB entries again until the router finds a physical/logical interface in the next hop field. Moreover, if an input packet is a search query, the received router returns information of the next hop field to the source node.

3.2 Network architecture

Figure 3 shows a brief overview of our network architecture. In this figure, we assume that all routers can recognize variable-length information as address information.

First, we consider a communication between host "jcga.ac.jp" and host "foo.com". Host "jcga.ac.jp" generates a packet in which the destination address is "foo.com". The packet is processed in router R:2-2, and a hash value is obtained by a hash function of which SHA-1 is assumed in this figure. Moreover, an entry with the hash value is found in FIB on R:2-2, and R:2-2 sends a query to router R:1-3, which is denoted in the next hop field of the matching entry. R:1-3 has an entry that matches the query in its FIB, and replies that R:3-2 can forward directly the packet to host "foo.com". According to the information returned from R:1-3, R:2-2 adds the address information to the packet, and forwards the packet to R:3-2. Although we send a request query to the host denoted in the next hop field, we can also send the packet directly to the next hop field in our architecture. We describe details of the forwarding mechanism in the

next section.

To reach for the remote host, all routers between source and destination nodes have to treat our routing protocol that uses variable-length address information. The number of addresses such as FQDN or keywords is even more enormous than IP addresses. Thus, all routers in the network store addresses of a single routing protocol that can reach to any other host in the whole network. The addresses of other protocols are encapsulated in the reachable addresses. In the current network, IPv4 address is the most suitable to encapsulate the packets of other routing protocols. If all routers in network can treat variable-length addresses with namespace, we can use the variable-length addresses as reachable addresses.

4 Abstraction of fundamental routing functions

In this section, we define fundamental routing functions that are commonly implemented in various routing protocols. Moreover, we propose the abstraction model to achieve generalized routing protocol.

4.1 Fundamental routing functions

We compare routing functions among five routing algorithms: OSPF (Open Shortest Path First), BGP-4, DNS, DHT, and P2P. OSPF and BGP-4 are layer-3 routing protocols, and the others are intelligent routings implemented on overlay networks. Furthermore, we extract four fundamental routing functions from the five protocols. Various routing protocols including the protocols mentioned above can be realized as one of extensions based on our fundamental functions.

1. *Forwarding messages:* This function is the most basic routing function to achieve communications between nodes and other routing functions including the following operations.

2. *Establishing peers:* When a router or a node joins to network, it sends a request message to establish a connection to existing nodes. For instance, a router sends information including its address information, router ID, protocol version, neighbor peers' information, and authentication algorithm etc.

3. *Advertisement and construction of FIB:* Routers and nodes participating in network advertise information to keep a reachable path within the network. The information consists of destination network ID, next hop, and cost etc. In addition, routers construct its FIB according to the rule of its routing protocol when they receive the message.

4. *Keeping peers:* Routers and nodes send a keepalive message periodically to detect change of the network topology and path failure.

4.2 Comparison among existing protocols

We describe functional differences between the protocols in terms of the four functions described in the previous section.

1. *Forwarding messages:* Message-forwarding methods are distinguished into two main branches. We define one as *notification-type forwarding* and the other as *search-type forwarding*.

   - **Notification-type forwarding**
     A source node generates a message with address information of the destination node, and sends the message to a neighbor node, which is directly connected to the source node. The intermediate routers belonging to the path between the source and the destination nodes relay the message to the next hop according to the address information in the header part of the message. If the destination node receives the message correctly, then an acknowledge message is sent to the source node.
   - **Search-type forwarding**
     A source node sends a search query to one of its neighbor nodes without specifying the destination node. The neighbor node that received the message searches an appropriate node in its own network by recursive lookup or iterative lookup. After that, the neighbor node sends information about the appropriate node, such as node ID, and nodes' address information, to the source node.

2. *Establishing peers:* This function is for establishing connections between nodes participating in network. Differences in establishing peers are caused from the differences between the message forwarding methods as described above. The differences in establishing peers among routing protocols are also categorized into two types. We define these two types as direct peering and reference peering, respectively.

   - **Direct peering**

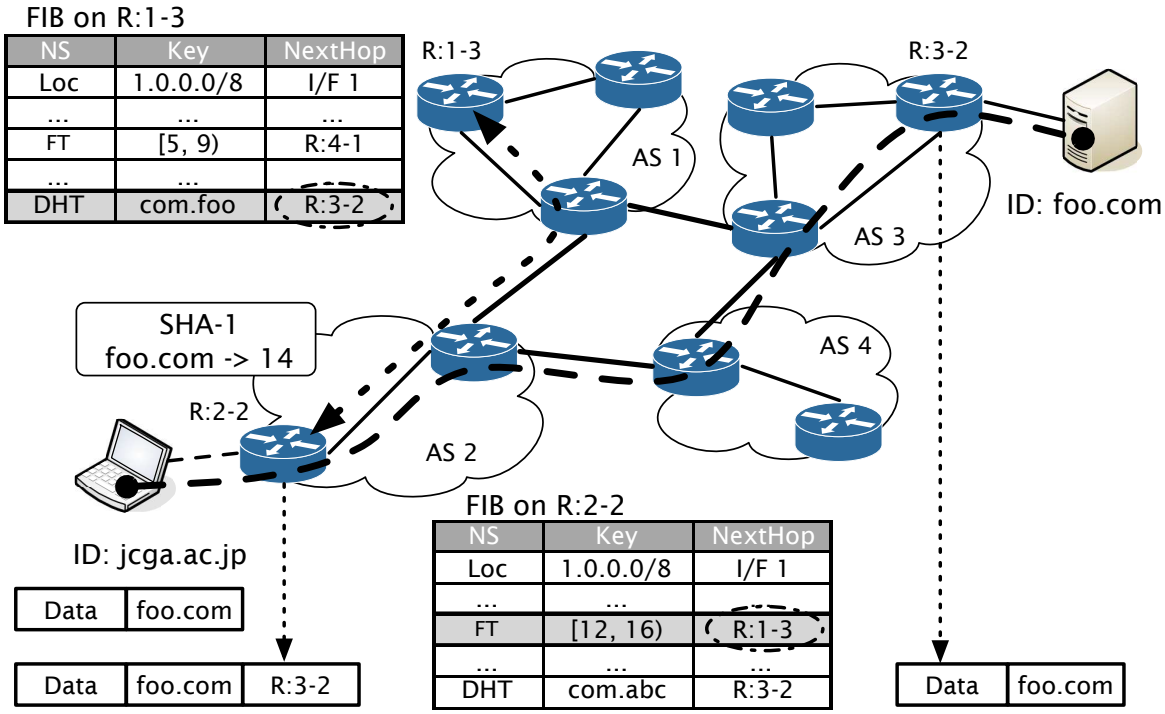


Figure 3: Overview of our network architecture

When a node wants to establish a new connection, the node sends a peering message including information of its node and information of the destination node to a neighbor node. The message is relayed to the destination node, and then the node that generated the message establishes a connection to the destination node. This peering type uses the notification-type forwarding to relay peering messages.

- **Reference peering**
  In this type, a node sends a peering message including only information of its node to a neighbor node. The neighbor node that received the message searches an appropriate node that should establish a new connection to the source node, and returns information of the appropriate node to the source node. The source node establishes a new connection to the appropriate node according to the returned information.

In case of the direct peering, participating nodes utilizes the notification-type forwarding to treat peering messages. On the other hand, the reference peering uses the search-type forwarding. Thus, we conclude that two different types of establishing peers are equivalent of that of the message-forwarding methods.

3. *Advertisement and construction of FIB:* Advertisement of routing information is one of important roles in order to construct the FIB at each router. We categorize this function into two different types, which are aggregation method and registration method.

   - **Aggregation method**
     Each node aggregates its routing information, and sends an advertisement message to neighbor nodes when a connection is established. The advertised information is propagated from lower nodes to upper nodes in hierarchical network to construct the path information in the whole network, which is managed in the FIB of each node. After establishing a connection, the FIB of node is not updated until the node receives an advertisement message.
   - **Registration method**
     To construct FIB table, each node sends a request message to an upper node that manages path information in the whole network. The path information is managed at the upper nodes according to rules of corresponding routing protocol. When upper nodes receive a request message, upper nodes notify of appropriate path informa-

tion to the requested node. The requested node updates its own FIB according to path information notified by the upper node.

We can use the aggregation method for address information with hierarchical structures such as IP address or FQDN. In contrast, we use the registration method for address information that cannot be aggregated due to ID/Locator separation. The aggregation method utilizes the notification-type forwarding to treat request messages. On the other hand, the registration method utilizes the search-type forwarding. Thus, we conclude that the difference between the aggregation method and the registration method is derived from the difference between the message forwarding methods as described above.

4. *Keeping peers:* A keep-alive message is sent periodically to detect path failures and to maintain a network topology. In this function, there is not much different between existing routing protocols.

4.3 The design of the abstraction model

We summarize the functional differences among the five routing protocols are the following three pairs.

1. Message forwarding: Notification-type forwarding and Search-type forwarding
2. Establishing peers: Direct peering and Reference peering
3. Advertisement and construction of FIB: Aggregation method and Registration method

The abstraction model that we propose should be the foundation prototype of various protocols. Namely, it is important to support these three differences in our abstraction model. As we mentioned in Sec. 3, in order to achieve information access with variable-length address information, our routers must also be able to exchange variable-length address with namespace.

5 Realization of the abstraction model by BGP-4

We describe the specification of BGPGA. BGPGA realizes the abstraction model we defined in Sec. 4.3.

5.1 Extension of BGP-4

To realize new router architecture such as the abstraction model, it is necessary to replace existing routers that are currently operating in the Internet, or to expand functions of the routers. Additionally, it is required to achieve the architecture with a low hardware implementation cost. In this paper, we use BGP-4 to design and implement the abstraction model. BGP-4 is commonly adopted as inter/intra-domain routing in the current Internet. By using one of existing routing protocols, most router venders can reduce the implementation cost, and it is easy to maintain compatibility with the conventional protocols. For future topic, we plan to implement our routing protocol as a native layer-3 protocol. We note that our implementation with BGP is an experimental, and BGP is not the fundamental protocol for our routing mechanism.

The routing functions of BGP-4 support only the notification- type message forwarding, direct peering, and the aggregation method to construct the FIB. BGP-4 uses UPDATE messages to notify advertisement information to other BGP routers. Moreover, BGP-4 cannot exchange variable-length address with namespace. Differences of each type of both peering and advertisement functions are caused by differences between messaging functions. Therefore, we extend the two following functions to BGP-4, and define the extended protocol as BGPGA.

1. Support of variable-length address with namespace
2. REQUEST/RESPONSE message for the search-type message forwarding

5.2 Support of variable-length address with namespace

In BGPGA, variable-length address with namespace is used by UPDA TE messages and REQUEST/RESPONSE message. We use mpBGP (Multiprotocol Extensions for BGP-4) [19] to support the variable-length address with namespace. The mpBGP was designed to enable BGP-4 to carry routing information for multiple network layer protocols (e.g., IPv6, IPX, etc.). The design concept of mpBGP resembles our abstraction concept. However, mpBGP cannot just support the variable-length address with namespace, because mpBGP is assumed to construct FIB for conventional IP addresses. To enable BGP-4 to support routing functions for multiple network layer routing protocols, mpBGP expands MP_REACH_NLRI and MP_UNREACH_NLRI attributes in the Path Attribute field of BGP UPDATE message.

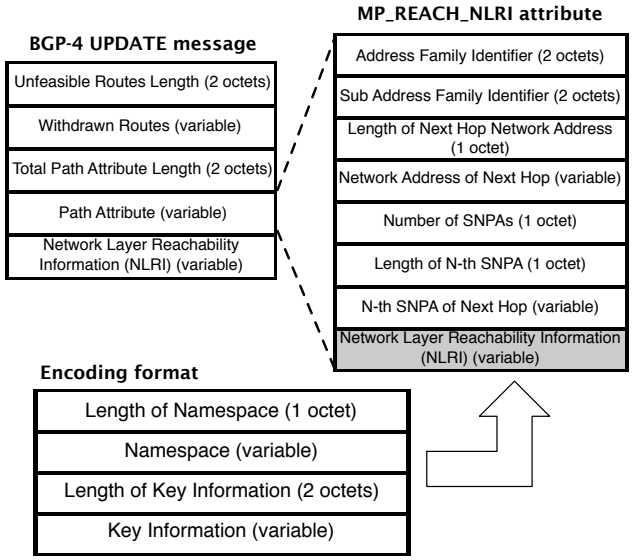


Figure 4: Encoding format of variable-length address with namespace

Figure 4 shows the format of UPDATE message used in mpBGP. In accordance with Fig. 4, we encode the variable-length addresses with namespace into NLRI field of MP_REACH_NLRI attribute, which is the bottom field of the format. The meaning of these fields is as follows:

- *Length of Namespace:* The length of the Namespace field as measured in octets.
- *Namespace:* This field contains the namespace.
- *Length of Key Information:* The length of the Key Information field as measured in octets.
- *Key Information:* This field contains variable-length address information.

5.3 REQUEST/RESPONSE message

In BGP-4, the Type field of the BGP header identifies the type of message. We define REQUEST/RESPONSE message for 7 with the value of the Type field. Figure 5 shows the format of the REQUEST/RESPONSE message. The meanings of each field in the REQUEST/RESPONSE message are as follows:

- *Type:* This field distinguishes whether the message is a request or a response. If the message is a request, the value of this field is 1. If the message is a response, the value is 2.

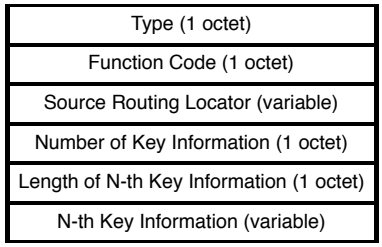


Figure 5: Format of REQUEST/RESPONSE message

- *Function Code:* This field provides additional information about the type of the key information carried in the message.
- *Length of Source Routing Locator:* The length of the Source Routing Locator field as measured in octets.
- *Source Routing Locator:* This field includes the network address of the source router.
- *Number of Key Information:* This field is the number of distinct key information to be listed in the N-th Key Information field. The value "1" may be used to indicate that the message is a request.
- *Length of N-th Key Information:* The length of the key information stored in the N-th Key Information field as measured in octets.
- *N-th Key Information:* In case of a request message, a search key is stored in this field. In the case of a response message, reply information is stored.

REQUEST message is used as a search request of recursive lookup or iterative lookup. When a router receives a REQUEST message, the router identifies the type of the Key Information from the namespace in Source Routing Locator field and Function Code field. If the router manages the key stored in Key Information field, the router sends a RESPONSE message to the requested router. If not, the router forwards the REQUEST message to another router.

RESPONSE message is used to reply of REQUEST message. The router that receives a RESPONSE message specifies the type of message from the namespace in Source Routing Locator field and Function Code field. In accordance with the type of message, the router processes the information of Key Information field.

```
#define AF_GA 134

struct ga_addr
     {
        unsigned char namespacelen;
        unsigned char addrlen;
        char namespace[32];
        char addr[256];
};
```

Figure 6: Address structure for variable-length address

6 Implementation of BGPGA

In this section, we implement BGPGA into GNU Zebra [20], which is one of routing software, based on our design guideline. The GNU Zebra supports many routing protocols including BGP-4 and mpBGP. Furthermore, many organizations of NSPIXP-6 [21] have adopted the Zebra as router software.

6.1 Implementing environment

We prepare two generic PCs to implement our abstraction model, BGPGA, and the PCs are used as BGPGA routers. We installed FreeBSD 7.2-RELEASE and GNU Zebra into the PCs. The PCs are connected directly each other, and also can process IPv4/IPv6 packets.

6.2 Definition of address family

The router that receives a UPDATE message including MP_REACH_NLRI field identifies AFI (Address Family Identifier) field, and decodes NLRI field of the MP_REACH_NLRI attribute depending on the type of the address family. To decode variable-length address with namespace, it is necessary to define a new address family for adopting variable-length address with namespace. We define an address family AF_GA to manage variable-length address with namespace. Figure 6 shows definitions of new address family and the address structure for variable-length address information with namespace in the kernel source of our routers. BGP connects to other BGP speaker by using TCP connection, which is established by UNIX socket. We do not modify the protocol families used by UNIX socket in the implementation of this paper. We use AF_INET and AF_INET6 to establish a TCP connection. Namely, our routers use IPv4/IPv6 address to establish a BGP connection.

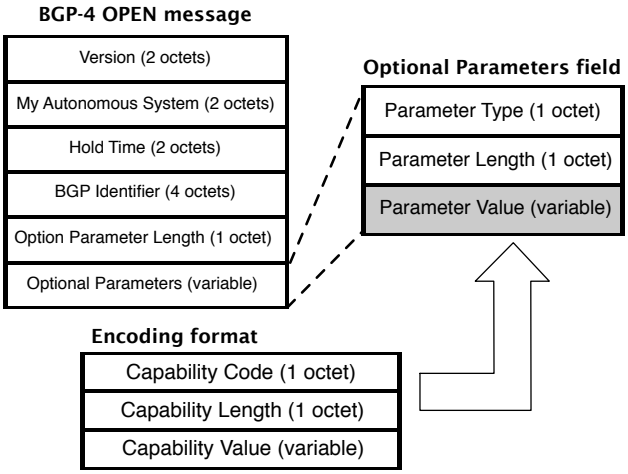


Figure 7: Encoding format of capabilities advertisement message

```
address-family GA           ! Capability advertisement of BGPGA
neighbor fe80::132 active   ! Allow negotiation
neighbor fe80::132 GA:DHT   ! Support for DHT as a BGPGA extension
neighbor fe80::132 GA:phone ! Support for phone as a BGPGA extension
exit-address-famiry
```

Figure 8: Zebra commands for capabilities advertisement

6.3 Capabilities advertisement

We adopt Capabilities Advertisement [22] to know whether other router can recognize variable-length address information with namespace. BGP OPEN message is sent when a BGP router establishes a BGP peering with another BGP speaker. Figure 7 shows the encoding format of the capabilities advertisement including the BGP OPEN message. The capabilities advertisement message is encoded in the Optional Parameters field in the BGP OPEN message, and the Parameter Type field included in the Optional Parameters field is set to 2.

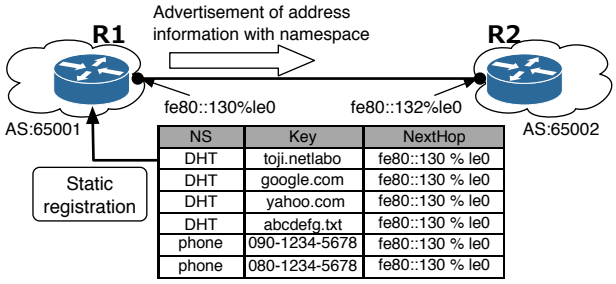
Moreover, BGPGA sets 130 to the Capability Code field of the capabilities advertisement message. The namespace of each protocol implemented as an extension of BGP is stored in the Capability Length field and the Capability Value field.

6.4 Extension and implementation of Zebra command

The capabilities advertisement is implemented as a function of Zebra BGP daemon. We implement some commands that can treat the capabilities advertisement as shown in Fig. 8. Moreover, in order to confirm advertised and received information, we also implement few commands to display this information, which is shown in Fig. 9.

```
! Display advertised address information with namespace
show bgp neighbors fe80::132 advertised-GA-routes
! Display received address information with namespace
show bgp neighbors fe80::132 received-GA-routes
```

Figure 9: Zebra commands to display advertised/received information



NS	Key	NextHop
DHT	toji.netlabo	fe80::130 % le0
DHT	google.com	fe80::130 % le0
DHT	yahoo.com	fe80::130 % le0
DHT	abcdefg.txt	fe80::130 % le0
phone	090-1234-5678	fe80::130 % le0
phone	080-1234-5678	fe80::130 % le0

Figure 10: Network environment for verification of BGPGA

6.5 Verification

We verify that our protocol can exchange variable-length address with namespace of BGPGA on our routers.

6.5.1 Verification environment

Figure 10 shows our network environment that has two routers peering with each other directly. Router R1 has an interface `le0` where the IPv6 address is `fe80::130` in IPv6. Router R2 has an interface `le0` where the IPv6 address is `fe80::132`. We configure some variable-length information addresses to R1 as static addresses described in Fig. 10.

When R1 or R2 tries to peer with the other router, peering router sends BGPGA capabilities advertisement message including namespaces of protocols implemented at the router. After establishing a BGP peering, R1 sends the address information with namespace. R2 receives the advertised information, and then constructs the FIB table by using the received information.

6.5.2 Verification results

We verify whether our router can recognize the capabilities advertisement message by using BGP OPEN message, and identify namespaces in the capabilities advertisement message. Moreover, we verify whether a router can advertise and receive variable-length information with namespace.

We implement the commands shown in Fig. 8 at both R1 and R2. These commands enable the routers to treat variable-length address information

```
bgpd> show bpg neighbors fe80::132 advertised-GA-routes
BGP table version is 0, local router ID is 172.16.55.130
   Network             Next Hop            Metric LocPrf Weight Path
*> DHT:toji.netlabo    fe80::130                0         32768 ?
*> DHT:google.com      fe80::130                0         32768 ?
*> DHT:yahoo.com       fe80::130                0         32768 ?
*> DHT:abcdefg.txt     fe80::130                0         32768 ?
*> phone:090-1234-5678 fe80::130                0         32768 ?
*> phone:080-1234-5678 fe80::130                0         32768 ?
Total number of prefixes 6
```

Figure 11: Advertised information from R1 to R2 supporting “DHT” and “phone”

```
bgpd> show bpg neighbors fe80::130 received-GA-routes
BGP table version is 0, local router ID is 172.16.55.132
   Network             Next Hop            Metric LocPrf Weight Path
*> DHT:toji.netlabo    fe80::130                0             0 65001 ?
*> DHT:google.com      fe80::130                0             0 65001 ?
*> DHT:yahoo.com       fe80::130                0             0 65001 ?
*> DHT:abcdefg.txt     fe80::130                0             0 65001 ?
*> phone:090-1234-5678 fe80::130                0             0 65001 ?
*> phone:080-1234-5678 fe80::130                0             0 65001 ?
Total number of prefixes 6
```

Figure 12: Received information at R2 supporting “DHT” and “phone”

with namespace, which are “DHT” and “phone”. Then, we confirm advertised information at R1, and received information at R2. Figure 11 shows the routing information advertised by R1, and Fig. 12 shows the received information at R2. As seen in these figures, R1 advertises all of the address information with namespace, and R2 received the routing information about “DHT” and “phone” because R1 and R2 can recognize the namespaces “DHT” and “phone” by using the commands shown in Fig. 8.

We next consider that R2 can recognize a part of the namespaces that R1 can treat. We configure so that R1 recognizes “DHT” and “phone”, and R2 recognizes “phone” only. R1 recognizes namespaces that R2 can treat by using the capabilities advertisement message, and sends advertised information including only namespaces that R2 can treat. Figures 13 and 14 show the confirmation result when R2 supports “phone” only. Although R1 can recognize “DHT” and “phone”, R1 sends the information that R2 can recognize, because R2 advertise the supporting namespaces by using the capabilities advertisement message.

Based on the results of our verification, BGPGA does not advertise variable-length address information with namespace to the traditional BGP speakers by using the capabilities advertisement message. Moreover, variable-length address information with namespace is advertised and received between BGP speakers that BGPBA is implemented.

7 Concluding remarks

In this paper, we have considered the fundamental routing functions from existing protocols imple-

```
bgpd> show bpg neighbors fe80::132 advertised-GA-routes
BGP table version is 0, local router ID is 172.16.55.130
   Network            Next Hop            Metric LocPrf Weight Path
*> phone:090-1234-5678 fe80::130                0         32768 ?
*> phone:080-1234-5678 fe80::130                0         32768 ?
Total number of prefixes 2
```

Figure 13: Advertized information from R1 to R2 supporting "phone" only

```
bgpd> show bpg neighbors fe80::130 received-GA-routes
BGP table version is 0, local router ID is 172.16.55.132
   Network            Next Hop            Metric LocPrf Weight Path
*> phone:090-1234-5678 fe80::130                0             0 65001 ?
*> phone:080-1234-5678 fe80::130                0             0 65001 ?
Total number of prefixes 2
```

Figure 14: Received information at R2 supporting "phone" only

mented in layer-3 or overlay networks. Moreover, we have designed the abstraction model of main routing functions as BGPGA. Finally, we have implemented and verified behaviors of BGPGA in a simple network environment. As a result, we have shown that BGPGA can treat variable-length address information with namespace, and BGPGA routers can exchange routing information including variable-length address. For future topics, we need to implement various routing protocols as extensions of BGPGA. Although we have implemented our routing protocol as an extension of BGP, we next plan to implement the protocol into layer-3 without using BGP-4.

Acknowledgements: This research was partially supported by National Institute of Information and Communications Technology (NICT) of Japan, the Grant-in-Aid for Young Scientists (B) (No. 11025086) from the Ministry of Education, Culture, Sports, Science and Technology (MEXT) of Japan.

References:

[1] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup service for Internet applications," in *Proceedings of ACM SIGCOMM 2001*, vol. 11, (San Diego, CA), pp. 149–160, August 2001.

[2] A. Rowstron and P. Druschel, "Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems," in *Proceedings of IFIP/ACM International Conference on Distributed Systems Platforms (Middleware 2001)*, (Heidelberg, Germany), pp. 329–350, November 2001.

[3] Z. Li and P. Mohapatra, "The impact of topology on overlay routing service," in *Proceedings of IEEE INFOCOM 2004*, vol. 1, (Hong Kong, China), pp. 408–418, March 2004.

[4] Y. Liu, H. Zhang, W. Gong, and D. Towsley, "On the interaction between overlay routing and underlay routing," in *Proceedings of IEEE INFOCOM 2005*, vol. 4, (Miami, FL), pp. 2543–2553, March 2005.

[5] Y. Sato, Y. Toji, S. Ata, and I. Oka, "Design of flexible layer-3 routing protocol to support variable-length address," in *Proceedings of IADIS International Conference WWW/Internet 2009*, (Roma, Italy), pp. 267–272, Novmeber 2009.

[6] Y. Rekhter, T. Li, and S. Hares, "A Border Gateway Protocol 4 (BGP-4)," *RFC 4271*, January 2006.

[7] S. Deering and R. Hinden, "Internet Protocol, version 6 (IPv6) specification," *RFC 2460*, December 1998.

[8] M. Ishiyama, M. Kunishi, K. Uehara, H. Esaki, and F. Teraoka, "LINA: A new approach to mobility support in wide area networks," *IEICE Transactions on Communications*, vol. E84–B, pp. 2076–2086, August 2001.

[9] R. Moskowitz and P. Nikander, "Host Identity Protocol (HIP) architecture," *RFC 4423*, May 2006.

[10] C. Vogt, "Six/One router: A scalable and backwards compatible solution for provider-independent addressing," in *Proceedings of the 3rd ACM International Workshop on Mobility in the Evolving Internet Architecture (MobiArch 2008)*, vol. 11, (Seattle, WA), pp. 13–18, August 2008.

[11] D. Farinacci, V. Fuller, D. Oran, D. Meyer, and S. Brim, "Locator/ID Separation Protocol (LISP)," *Internet Draft draft-farinacci-lisp-12.txt*, March 2009.

[12] J. Pan, R. Jain, S. Paul, and C. So-in, "MILSA: A new evolutional architecture for scalability, mobility, and multihoming in the future Internet," *IEEE Journal on Selected Areas in Communications*, vol. 28, pp. 1344–1362, October 2010.

[13] M. Menth, M. Hartmann, and M. Hofling, "FIRMS: A mapping system for future Internet routing," *IEEE Journal on Selected Areas in*

Communications, vol. 28, pp. 1326–1331, October 2010.

[14] C. Perkins, D. Johnson, and J. Arkko, "Mobility support in IPv6," *RFC 6275*, July 2011.

[15] "GENI: Global Environment for Network Innovations." `http://www.geni.net/index.html`.

[16] "FIND: Future Internet Design." `http://bgp.potaroo.net/bgprpts/rva-index.html`.

[17] "BGP statistics from Route-Views data." `http://bgp.potaroo.net/bgprpts/rva-index.html`.

[18] "Distribution by top-level domain name by host count Jul. 2012." `http://ftp.isc.org/www/survey/reports/2012/07/`.

[19] T. Bates, R. Chandra, D. Katz, and Y. Rekhter, "Multiprotocol Extensions for BGP-4," *RFC 4760*, January 2007.

[20] "GNU Zebra." `https://www.mangob2b.com/en/zebra`.

[21] "Peering information NSPIXP6." `http://wide.ad.jp/nspixp6/peering-status.html`.

[22] J. Scudder and R. Chandra, "Capabilities Advertisement with BGP-4," *RFC 5492*, February 2009.