Network Working Group
Internet Draft
Intended status: Informational
Expires: March 2015

J. Lee
ETRI
W. Lim
ETRI
W. Chun
HUFS

September 17, 2014

Scalable Domain-based Routing Scheme
draft-lee-icnrg-domainbasedrouting-02.txt

Status of this Memo

Copyright Notice

Abstract

Moving the focus from nodes to information objects raises scalability issues because the number of addressable information objects is huge compared to the number of nodes, so scalable routing is an important challenge issue of ICN. There are two types of naming scheme which have been proposed in the ICN literatures: hierarchical and flat. To guarantee clean separation of identifier from locator and scalability of routing, we chose flat name and designed domain-based routing. A significant requisite to be considered when flat name is used is an efficient name-resolution system (NRS). Bloomfilter-based NRS [BF NRS] is our proposal to this issue. Once a name is resolved into locator(s), discovery and delivery steps are carried out based on the routing scheme of locator. For scalability in routing of locator, network is projected into hierarchically-organized domain structure. A domain is a group of nodes or other domains. This composition could be physical or logical. Each domain has its own identifier, and the concatenation of domain IDs from top level domain to a certain domain which a node belongs to plays the role of "locator" for that node. Each domain has one or more domain gateways. All traffic from/to the domain should pass through any of domain gateways. Routing information based on this type of locators is exchanged among domain gateways by using modified link-state routing protocol which can suppress LSA explosion [LSR]. Thanks to this hierarchical domain structure, locators are highly aggregatable (which means scalable routing), and any node in heterogeneous network can communicate each other.

Table of Contents

1. Introduction

Moving the focus from nodes to information objects raises scalability issues. Currently, the Internet is addressing on the order of 10^9 nodes, whereas the number of addressable ICN objects is expected to be several orders of magnitude higher [ICNRG charter]. Therefore, scalable routing scheme is an important challenge issue of ICN.

ICN routing locates a data object based on its name which is initially provided by a requestor. ICN routing may comprise 3 steps: a name resolution step, a discovery step, and a delivery step. Depending on how these steps are combined, ICN routing schemes can be categorized as Route-By-Name Routing (RBNR), Lookup-By-Name Routing (LBNR), and Hybrid Routing (HR) [2].

To keep the advantage of separating identifier from locator we chose flat name scheme, which means LBNR is used as routing scheme. If LBNR is used a Name Resolution System (NRS) is required. An efficiency of NRS is a significant requisite in LBNR scheme because every name

should be resolved through the NRS. Bloomfilter-based NRS [9] is our answer to this issue.

Once the locator for the name is obtained, discovery and delivery steps may depend on the routing scheme of locator. For scalability in routing scheme of locator, we projected networks into hierarchically-organized domain structure. A domain is a group of nodes of other domains, and this group could be physical or logical. Each domain has its own identifier, and the concatenation of these IDs from top level domain to a certain domain which a node belongs to plays the role of "locator" for that node.

Each domain could have one or more domain gateways. All traffic from/to the domain should pass through any of its domain gateway. Routing information based on the locator is exchanged among domain gateways which belong to different domains by running modified link-state routing protocol. It suppresses LSA storm by using LSA-filtering rule between parent and child domain.

Thanks to this hierarchical domain structure, locators are highly aggregatable, and routing operates regardless of the underlying network protocol.

2. Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC-2119 [RFC2119].

In this document, these words will appear with that interpretation only when in ALL CAPS. Lower case uses of these words are not to be interpreted as carrying RFC-2119 significance.

In this document, the characters ">>" preceding an indented line(s) indicates a compliance requirement statement using the key words listed above. This convention aids reviewers in quickly identifying or finding the explicit compliance requirements of this RFC.

3. Terminologies

o Domain: a logical/physical group of nodes (requestor, contents server, or other "domain") which have similar characteristics (E.g. network protocol, geographical region, same type of contents, similar category of contents etc.).

o Locator: locator is a concatenation of IDs of hierarchically organized domains.

o Domain Gateway: a kind of border gateway for a domain. The domain gateway forwards request message or NDO data, and runs routing protocol carrying routing information based on the locator.

4. Domain-based Routing Scheme

4.1. Network as Hierarchically-organized Domain Structure

```
                      domain ID: A
        +---------------------------------------+
        |    +-----+       +-----+              |
        |    |     +-------+  1  |              |
        |    |     |       |     |              |  tier-0
        |    ++----+       +-----+              |  locator: A
        |   / |      \                          |
        |  /  |          \                      |
        | /   |             \                   |
        +/----|-----------------\---------------+
        /     |                      \
       /      |       domain ID: B      \
    +---------|----------------------------+
    |         |                            |
    |    +----++                +-----+    |
    |    |  2  +----------------+  3  |    |
    |    |     +---------+      |     |    |       tier-1
    |    +-----+         |      +-----+    |       locator: A:B
    |                  +-+----+            |
    |                  |      |            |
    |                  |      |            |
    |                  +-+----+            |
    |                 /  |     \           |
    |                /   |      \          |
    |               /    |       \         |
    +--------------/-----|--------\--------+
                  / +----+         \
                 /  | domain ID: C  \
               +----|-----------------+
               |    |                 |
               | +--+--+    +-----+   |
               | |  4  +----+  5  |   |           tier-2
               | |     |    |     |   |           locator: A:B:C
               | +-----+    +-----+   |
               |                      |
               +----------------------+
```

Figure 1 Hierarchical domain structure

In our scheme network is projected as a set of domains, and each of which may be further decomposed into smaller domains recursively. Each domain has its own ID and the concatenation of consecutive domain IDs can play the role of "locator". Figure 1 shows an example of domain structure. For example, locator for node 3 is A:B, and locator for node 5 is A:B:C. All traffic to and from a domain should pass through a domain gateway. Node 2, 4 in this figure are domain gateways.

4.2. Name Resolution System (NRS)

If flat name scheme is used, efficient name resolution system is very important. In Name resolution step a requestor queries current locator of a name to the NRS. For fast processing hashing technique (like bloom filter) could be used to implement NRS. We also proposes Bloom filter based NRS [BF NRS].

4.3. Scalable Domain-Based Routing

Generally speaking, routing means the process of selecting best paths in a network along which to send network traffic. In other words, routing is the process of building network topology database and computing routing table based on that network topology. In our domain based structure, we build network topology graph by link-state routing protocol. However, not like existing link-state routing protocol, vertex of our network topology graph could be an actual communication entity or "a domain". This means each domain gateway which belongs to different domain in different tier has different topology graph. The higher tier a domain gateway belongs to, more abstracted topology graph it would have.

```
                   domain ID: A
          +----------------------------------------+
          |                                        |
          |    +------+       +------+             |
          |    |     +--------+  1  |              |
          |    |     |       |     |               |  tier-0
          |    +------+       +------+             |  locator: A
          |                                        |
          |                                        |
          |                                        |
          +----------------------------------------+
     Figure 2 Domain gateway 1's view for whole network topology
```

o Contents server as a domain gateway

Usually a contents server maintains many NDOs (i.e. entities with name), and therefore the contents server could be considered as another "domain gateway", which means it also joins routing procedure. But it does not need to maintain forwarding cache for NDOs (i.e. contents server already knows where NDOs are!).

4.3.1. LSA filtering to prevent LSA flooding

Each domain gateway runs modified link-state routing protocol. Each LSA transfers "locator" information instead of IP prefix. Thus, locators appears in destination field of routing table.

To suppress LSA storm each domain gateway forwards LSAs to the other domain gateways according to the following filtering rules:

o LSAs in tier N domain are forwarded to all domain gateways in same tier domain.

o LSAs in tier N-1 are injected to tier N gateways.

o LSAs in tier N+1 are filtered, LSA which has aggregated locator is injected to tier N instead. (E.g. see Figure 6, [GW #2] delivers LSA which includes 0x0B:0x0C to [GW #1], locator for domain 0x0E (0x0B:0x0C:0x0E) is not flooded into [GW #1]).

This results in topology reduction effect. Therefore, each domain gateway would have topology graph which is as reduced as possible in its current position (tier).

4.4. Name based Forwarding

Under the principle of separating identifier from locator, header of each request message and data packet include name information only. Therefore, each domain gateway should know where to forward packets to reach the NDO. The domain gateways can get this information from "forwarding cache". Forwarding cache for certain name is built during discovery step. Forwarding cache includes following information.

o Destination name: name of NDO, or name of requestor

o Next-hop address: address of next-hop domain gateway. It can be any type of address (e.g. IPv4 address, IPv6 address, Ethernet address, ...)

o Next-hop protocol id: protocol for next-hop gateway (e.g. IPv4, IPv6, Ethernet, ...)

- o Output interface

Each domain gateway maintains routing table based on "locator". When the domain gateway receives packets destined for certain "name", it queries NRS to get current "locator" of NDO mapped with the "name". With this "locator" the domain gateway looks up its routing table to find routing table entry matching "locator". If matching one is found, forwarding cache for "name" is built by merging "name" and "forwarding information" part of the routing table entry.

- o The forwarding cache is built reactively, and maintained by lifetime timer.

- o Usually many forwarding cache can be shared by many request from many other requestor.

4.5. In-network Caching

Under our structure, NDO can be cached on any domain gateway, or it could even be cached any node in the domain managed by the domain gateway. Once the NDO is cached on any node, the node adds its "locator" to the name of NDO through NRS. Afterwards, during discovery step cached NDO which has closer locator could be used.

4.6. Mobility Management

Mobility support can be implemented very easily because all messages and data packets don't include any location-related information (e.g. locator). Usually if any domain gateway detects transmission failure, it restarts discovery step, then several forwarding caches on the path will be up-to date. There is almost nothing that end nodes should do.

4.7. Inner-domain Communication

TBD.

4.8. Security

TBD.

4.9. Transport Service

TBD.

4.10. Operations

4.10.1. Presenting

The first thing to do that every node (e.g. requestor, contents server...) attaches network is to update (or add newly) its current location. This is done by the domain gateway of domain which the nodes belongs to.

```
+---------+  +-------------+  +----+  +----------------+  +---------+
|requestor|  | domain gw   |  |NRS |  | domain gw      |  |contents|
|         |  |for requestor|  |    |  |for contents svr|  |  svr   |
+---------+  +-------------+  +----+  +----------------+  +---------+
     |               |           |             |                |
     |    1.0        |           |             |                |
     |-------------> |           |             |      2.0       |
     |               |   1.1     |             |<---------------|
     |               |---------->|    2.1      |                |
     |               |           |<------------|                |
     |               |           |             |                |
     |               |           |             |                |
                Figure 3 Presenting current location
```

- o 1.0 - presence message: deliver {requestor's name} to the domain gateway
- o 1.1 - update message: domain gateway sends update including {requestor's name, its locator} to NRS
- o 2.0 - presence message: deliver {name of NDO} to the domain gateway
- o 2.1 - update message: domain gateway sends update including {name of NDO, its locator} to NRS

4.10.2. Name Resolution & Discovery

When a requestor issues request for a NDO on the network, name resolution and discovery procedure is carried out by domain gateways.

During this procedure, forwarding cache entry for NDO includes followings:

- o NDO's ID
- o Next hop information (found in the routing table entry)

Forwarding cache entry for the requestor includes followings:

o Requestor's ID

o Previous hop information (found in the packet header)

```
                                   +----+
                                   |NRS |
                                   +----+
   +---------+  +-------------+       |    +----------------+  +-------+
   |requestor|  | domain gw   |       |    | domain gw      |  |contens|
   |         |  |for requestor| .. ...| .. |for contents svr|  |  svr  |
   +---------+  +-------------+       |    +----------------+  +-------+
        |               |             |            |               |
        |    1.0        |             |            |               |
        |-------------> |    1.1      |            |               |
        |               |------------>|            |               |
        |               |    1.2      |            |               |
        |               |<------------|            |               |
        |               |             |            |               |
        |               |----+        |            |               |
        |               |    |1.3     |            |               |
        |               |<---+        |            |               |
        |               |             | 1.4        |               |
        |               |------------------------->|               |
        |               |             |            |----+          |
        |               |             |            |    |1.5       |
        |               |             |            |<---+          |
        |               |             |            |       1.6     |
        |               |             |            |-------------->|
        |               |             |            |       1.7     |
        |               |             | 1.8        |<--------------|
        |     1.9       |<-------------------------|               |
        |<--------------|             |            |               |
        |               |             |            |               |
```

Figure 4 Discovery procedure

o 1.0 - request message: request message including {name of NDO}

o 1.1 - lookup message: request {locator} for {name of NDO}

o 1.2 - reply message: deliver {locator(s)} for {name of NDO}

o 1.3 - build bi-directional forwarding cache

o 1.4 - forward request message: this request message includes {locator for requested NDO} to avoid unnecessary name resolution. (While this request message is being forwarded to the domain gateway for contents server, domain gateways on the path build bi-directional forwarding caches)

o 1.5 - build bi-directional forwarding cache

o 1.6 - deliver request message to the contents server

o 1.7 ~ 1.9 - delivery step, deliver NDO data to the requester

4.10.3. Delivery

After discovery step, forwarding caches are ready on path to the NDO, and on path to the requestor. Forwarding caches to the NDO may be reused by other request from other requestors who want same NDO. Forwarding caches to the requestor are used to deliver NDO.

See 1.7 ~ 1.9 in Figure 4.

4.10.4. Mobility Management

TBD.

5. Comparison with other ICN approaches [ICN survey]

	PSIRP	NetInf	Domain-based Routing
Namespace structure	flat with structure	flat with structure	flat no structure
Human-readable names	no	no	no
NDO granularity	Objects	Objects	Objects
Routing aggregation	scope /explicit	publisher	Domain
Routing of NDO request	NRS	Hybrid NRS & name based	NRS
Routing of NDO	Source routing using BF	Reverse request path/ direct IP	Reverse request path
API	Publish /Subscribe	Synchronous get	IPlug/Dsocket
Transport	IP/PSIRP	Many including IP	Many including IP

Figure 5 Comparison with other ICN approaches

Figure 5 shows summary of different ICN approaches. PSIRP and NetInf are using flat name like ours, however our approach uses flat name without any structure for strict separation of name from location. Names for all of them are non-human readable. Routing is aggregated by the domain in our approach. All of them are using NRS for routing of NDO request. To route NDO to the requestor NetInf and our approach are using reverse request path. Regarding transport method, NetInf and our approach can support any protocol including IP, which means they support communication between nodes in heterogeneous network.

6. Example scenario

```
   +-----------------------------------------------------------------+
   |                                                                 |
   |                                   (Domain Id: 0x0A, IPv4)       |
   |                         +------[GW #0]----------------------+   |
   |                         |        |                          |   |
   |                         |        |                          |   |
   |                         |        |                          |   |
   |                         |        |                          |   |
   |                         |        +--------------------------+   |
   |                         |                                       |
   |  (Domain Id:0x0B, IPv4) |                                       |
   |  +-------------------[GW #1]--------------------------------+   |
   |  |        ^^^^^^^^      |      <------------------------>   |   |
   |  |        | NRS  |----- +------< requestor A (Id: 0xFF) >   |   |
   |  |        ^^^^^^^^      |      <------------------------>   |   |
   |  |                      |                                   |   |
   |  |                      +--------------------------------+  |   |
   |  |                      |                                |  |   |
   |  |(Domain Id:0x0C, IPv6)|                                |  |   |
   |  |  +----------------[GW #2]-+ (Domain Id:0x0D, Ethernet)|  |   |
   |  |  |                   |    |    +------------------[GW #3]|   |
   |  |  |                   |    |    |   server3          |  | |   |
   |  |  |  server1          |    |    |  +========+        |  | |   |
   |  |  | +=============+   |    |    |  |        |        |  | |   |
   |  |  | | StarWars    |   |    |    |  |        +--------+  | |   |
   |  |  | | (0xEE)      |   |    |    |  |        |        |  | |   |
   |  |  | |             |   |    |    |  +========+        |  | |   |
   |  |  | |             +---+    |    |                    |  | |   |
   |  |  | +=============+   |    |    |      <------------->  | |   |
   |  |  |                   |    |    |      < requestor B >  | |   |
   |  |  |(Domain Id:0x0E, IPv4)  |    |      <  (Id: 0xAF) >  | |   |
   |  |  | +-----------[GW #4]-+  |    |      <------------->  | |   |
   |  |  | |   server2      |  |  |    |                       | |   |
   |  |  | |  +==========+  |  |  |    +-----------------------+ |   |
   |  |  | |  |          |  |  |  |                              |   |
   |  |  | |  |          +--+  |  |                              |   |
   |  |  | |  +==========+     |  |                              |   |
   |  |  | +-------------------+  |                              |   |
   |  |  +------------------------+                              |   |
   |  +----------------------------------------------------------+   |
   +-----------------------------------------------------------------+
```

Figure 6 Sample Domain Architecture

6.1. Routing tables

Destination	nexthop	out if
0x0B:0x0C	GW #2's address	if1
0x0B:0x0D	GW #3's address	if1
0x0B	-	if1
0x0A	GW #0's address	if0

Figure 7 Example of Routing Table (GW #1)

Destination	nexthop	out if
0x0B:0x0C:0x0E	GW #4's address	if1
0x0B:0x0D	GW #3's address	if0
0x0B:0x0C	-	if1
0x0B	GW #1's address	if0
0x0A	GW #1's address	if0

Figure 8 Example of Routing Table (GW #2)

Destination	nexthop	out if
0x0B:0x0C	GW #2's address	if0
0x0B:0x0D	-	if1
0x0B	GW #1's address	if0
0x0A	GW #1's address	if0

Figure 9 Example of Routing Table (GW #3)

6.2. Server Discovery (path discovery)

When a requestor gets locator for a specific NDO from NRS, it build a path discovery message and forwards to the default domain gateway (in

this example, "requestor A" send path discovery message to GW #1). The path discovery message includes followings:

```
+------------------------------------------------------------+
| Network header                                             |
+------------------------------------------------------------+
| Requestor's ID                                             |
+------------------------------------------------------------+
| NDO's ID                                                   |
+------------------------------------------------------------+
| NDO's current locator                                      |
+------------------------------------------------------------+
```

If a domain gateway receives path discovery message, it looks up its forwarding cache table firstly. If there is no forwarding cache entry for the NDO's ID, the domain gateway builds new forwarding cache entry. The domain gateway searches its routing table by using "locator (included in the path discovery message)" as key, then it builds forwarding cache entry for the NDO's ID and requestor's ID.

Forwarding cache entry for NDO includes followings:

o NDO's ID

o Next hop information (found in the routing table entry)

Forwarding cache entry for the requestor includes followings:

o Requestor's ID

o Previous hop information (found in the network header)

After building the forwarding cache, the path discovery message is forwarded to the next hop domain gateway (network header is changed according to the next-hop information of forwarding cache), and repeat this procedure until it reaches domain which includes the NDO.

Forwarding caches are managed in the manner of timer-based way.

6.3. Name-based Forwarding

When the contents server which has requested NDO receives path discovery message, it sends NDO data by using following message format:

```
+-------------------------------------------------------------------+
| Network header                                                    |
+-------------------------------------------------------------------+
| Requestor's ID                                                    |
+-------------------------------------------------------------------+
| NDO data                                                          |
+-------------------------------------------------------------------+
```

Network header would be changed according to the next hop information of forwarding cache.

7. Security Considerations

TBD.

8. IANA Considerations

TBD.

9. References

9.1. Informative References

[ICNRG charter] http://irtf.org/icnrg

[ICN Challenges] D.Kutscher, "ICN Research Channelges", internet-draft, July 2013

[aRoute] Ahmed, Reaz, Md Faizul Bari, Shihabur Rahman Chowdhury, Md Golam Rabbani, Raouf Boutaba, and Bertrand Mathieu. "aRoute: A Name Based Routing Scheme for Information Centric Networks." In IEEE International Conference on Computer Communications (INFOCOM) Mini-Conference, 2013.

[ICN survey] Ahlgren, Bengt, Christian Dannewitz, Claudio Imbrenda, Dirk Kutscher, and Borje Ohlman. "A Survey of Information-Centric Networking." Communications Magazine, IEEE 50, no. 7 (2012): 26-36.

[Id net] http://www.idnet.re.kr/

[BF NRS] J. Hong, "Bloom Filter-based Flat Name Resolution System for ICN", http://www.ietf.org/internet-drafts/draft-hong-icnrg-bloomfilterbasedname-resolution-00.txt

[LSR]   W. Lim, "Design of Scalable Link-State Routing Protocol for
Efficient Inter-Domain Routing" in IEEE Communication
letter (is being under examination)

Authors' Addresses

Joo-Chul Lee
ETRI
161 Gajeong-dong, Yuseong-gu, Daejon

Phone:
Email: rune@etri.re.kr

Wan-Seon Lim
ETRI
161 Gajeong-dong, Yuseong-gu, Daejon

Phone:
Email: wslim@etri.re.kr

Woo-Jik Chun
HanKuk University of Foreign Studies
81, Oedae-ro, Mohyeon-myeon, Cheoin-gu,Yongin-si, Gyeonggi-do, 449-
791, Korea

Phone:
Email: woojikchun@gmail.com