INTERNET-DRAFT
Intended Status: Informational
Expires: August 21, 2013

Chunfeng Yao
Rong Wang
Yuanzhe Xuan
Zhefeng Yan
Huawei

February 17, 2013

Container Assisted Naming and Routing for ICN
draft-ietf-icnrg-naming-routing-00

Abstract

In ICN (Information-centric network), everything is an identifiable object with a name, therefore the number of name prefixes is a few orders of magnitude higher than that in current BGP routing table. Towards scalable routing in ICN, we propose a name scheme, called container assisted naming. With this scheme, an object name consists of two components: a content name which uniquely specifies the object within certain scope, and one or more containers which define access relationships to the object. This document illustrates the concept of container and how it assists scalable routing in ICN.

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/1id-abstracts.html

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html

Copyright and License Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1 Design Principles

1.1 Naming in ICN

Name plays a critical role in serving many fundamental functions in ICN: a unique name identifies a mutable or immutable content or information object; it is used to look up and access data in network cache; and it is used for routing and forwarding. Therefore, naming is the foundation for ICN architecture.

We follow several principles for defining a naming scheme in ICN:

-Unique: A name uniquely identifies an object or entity within some scope (e.g., within a domain or the entire Internet).

-Locatable: A name enables interested entities to locate the identified object in a network. For this purpose, the name is either routable to reach the object, or includes information to derive the routable location(s) of the object.

-Location-independent: identified object may be served by any node that has the object, which is independent of location or original source of the object.

-Scalable Routing: The number of potential prefixes in global content or information object namespaces is several orders of magnitude larger than that in IP address namespace [Cisco-Name]. Therefore, ICN name should support routing scalability.

2 Container-assisted Naming

2.1 Container

The container component in our naming scheme is appended to the original content name to assist routing. This naming scheme is not restricted to the hierarchical [NDN] or flat name [DONA].

A container is a space where content or information objects reside. A container in an ICN name can be one of the following:

-A content or object identifier prefix, e.g., Huawei.com/blog;
-An enterprise or organization name, e.g., Huawei.com, tsinghua.edu;
-A host or a device such as a mobile phone or a content storage device, e.g., chinamobile/johndoe/iphone;
-A mobile network, such as airplane, train, e.g., Airchina/ca1314;
-A network domain, e.g., cn, cn/gd, cn/gd/sz.

Access Container: If container A is contained in container B, and container B has the routing entry to container A, container B is defined as the access container for container A. One container can have multiple access containers, and it can provide access service to multiple other containers as while.

Base Container: The smallest container that can be used to assist routing for other container(s) is defined as the base container of the content. One content can have multiple base containers, such as multi-homing scenario.

2.2 Container Assisted Naming Formats

A container assisted name can be expressed in a Directed Acyclic Graph or a tree as Figure 1 shows.

```
                  +----+                            +----+
                  |Name|                            |Name|
                / +----+ \                        / +----+ \
               /     |    \                      /     |    \
    +---------++---------++---------+  +---------++---------++---------+
    |container||container||container|  |container||container||container|
    |   A     ||   B     ||   C     |  |   A     ||   B     ||   C     |
    +---------++---------++---------+  +---------++---------++---------+
            \   /     \                      /        |     \
          +---------+  +---------+    +---------++---------++---------+
          |container|  |container|    |container||container||container|
          |  D      |  |   E     |    |  D      ||  D      ||   E     |
          +---------+  +---------+    +---------++---------++---------+
                   DAG                               TREE
```

Figure 1.Container assisted naming

As illustrated in Figure 1, the content name can be hierarchical or flat, container A, container B, and container C are the base containers, container D is the access container for A, and container D and container E are the access containers for B.

As can be seen from figure 1, base containers A, B, C provide access to the content or information objects directly. These base containers may locate in or logically access to other containers, namely access containers. The base containers may be referred to as "first layer/depth containers" in the DAG or TREE, and their access containers may be referred to as "second layer/depth containers". Similarly, the access containers for the "second layer containers" may be called "third layer containers".

In an ICN request packet, we propose a new content name format by using depth-first traversal of the container access relationship tree (hereinafter referred to as CART). The name shown in Figure 1 can be represented as the following:

Name = {content name | container A || container D | container B || container D || container E | container C},

Where "|" is container separator. Each node of the tree is listed in the sequence of depth-first traversal, and separated by a separator "|". As the depth increases, the number of separators also increases. A container at higher layer provides access service for the ones at lower layer. The depth of a CART has no theoretical limitation.

2.3 Resolving Container

The location of a container can change from time to time, the same as its access containers. In order to avoid frequent changes in containers and guarantee the name persistency, we propose container resolution, which allows a container to dynamically register and query its access containers from a resolution system.

Container Resolution System is a distributed system composed of several container resolution servers deployed in the network to store the mapping relationship between a container and its access containers. The system supports dynamic register and query operations.

A container resolution request can be initiated by a content consumer or intermediate network node, when the request carries a resolvable container.

A container in a name or in a resolution result can carry an attribute "resolvable", if "resolvable = yes", this container can be further resolved in the resolution system to get its access container(s); if "resolvable = no", it is not resolvable; that is, it may be the deepest level container in the resolution tree. The default value is set as "resolvable = no".

A resolved container from resolution system can carry the following two attributes:

-- Cacheable: defines whether the resolution result can be cached in the network thus can be shared with other nodes or consumers. The default value is set as "cacheable = no".

-- Time To Live (TTL): defines the fresh time a resolved container result can live in the network. The default value is set as "TTL = 0", namely uncacheable.

If a resolved container from the resolution system is also resolvable, the container resolution process is iterated.

3 Container Assisted Forwarding

3.1 Container Assisted Forwarding Procedure

When a request is received, an ICN router first checks the content store for a matched content with the content name in the request packet. If a matched content is acquired, it is returned to the consumer or the preceding router.

If there is no matched content in content store, the router looks up its FIB for outbound faces. The forwarding decision is made based on content name first. If there is no match in the FIB, the decision is then made based on carried container(s) in the name. There can be two ways to assist forwarding with containers by the router.

3.1.1 Forwarding with full container resolution

The procedure of forwarding with full container resolution can be described as follows:

Firstly, the router checks whether there is a container carried in the request. If not, it forwards the request to the default routing entry or drops it.

When there is a container carried, the router checks whether the request includes resolvable container or not. If "yes", the router initiates a container resolution request to the resolution system. The result can be responded either by a server in the resolution system or an intermediary's cache where a previous resolution result resides. If the resolved container is also resolvable, the resolution process is iterated by the router with the result container(s).

When all resolvable containers are resolved, the complete set of container(s) including the carried ones in the name and the resolved ones from the resolution system are used to forward the request. The traversal order can be depth-first or can be breadth-first. In case one of these containers matches a FIB entry, the request is forwarded to the corresponding outbound interfaces. Otherwise, the request is forwarded to default outbound faces or dropped.

3.1.2 Forwarding with iterative container resolution

The procedure of forwarding with iterative container resolution can be described as follows:

Firstly, the router checks whether there is at least one container carried in the request. If there is not, the request is forwarded to default outbound interfaces or dropped.

For all carried containers, the router matches them with its FIB entries, if one container matches, the request is forwarded according to the matched outbound faces.

When the carried container(s) have no match in FIB, the router checks whether one of the carried containers is resolvable. If yes, it sends container resolution request to the resolution system to get the access container for the resolvable container(s).

After that, the router matches the resolved container(s) with FIB. When one container matches an FIB entry, it forwards accordingly. If there is no match, the container resolution can be iterated with other resolvable containers in the request.

When there is no more resolvable container in the request, the request can be forwarded to default outbound faces, or dropped.

3.2 Scalable Container Assisted Routing

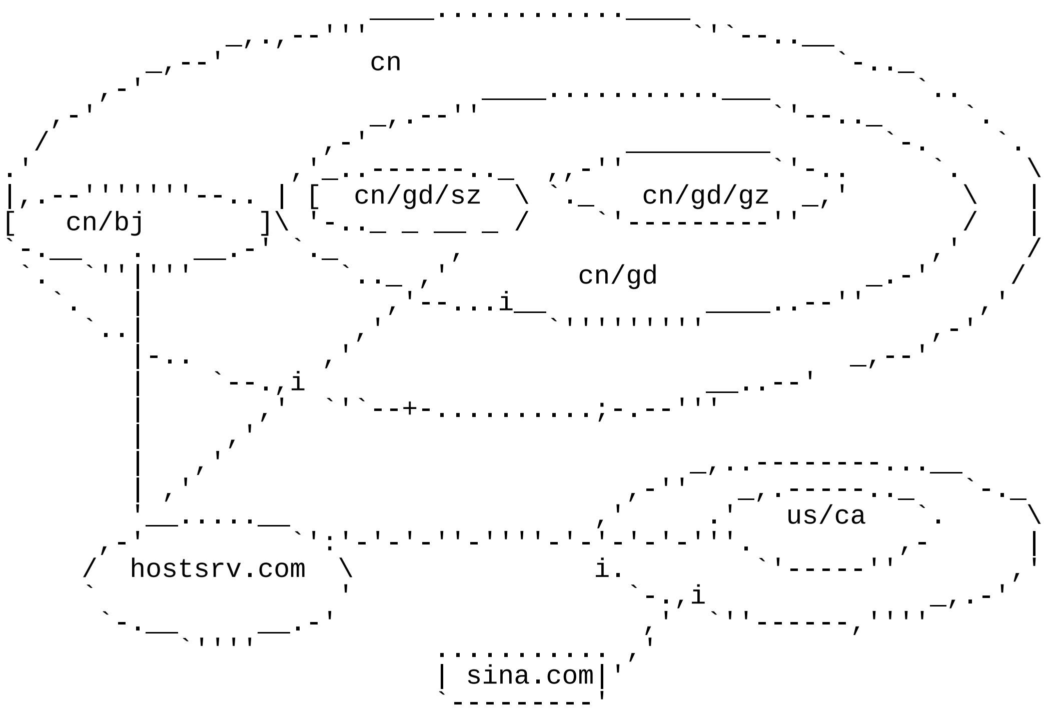

Figure 2. Solution to Routing Scalability

The scalability of name-based routing in ICN can be well addressed by containers. We divide containers into two categories: topology oriented and non-topology oriented.

3.2.1 Topology oriented containers

Topology oriented containers aggregate naturally. For example, as illustrated in Figure 2, the whole network of China can be viewed as a national level container "cn", and province level containers such as "cn/gd" for Guangdong province and "cn/sd" for Shandong province are contained in "cn". Similarly, a city level container such as "cn/gd/sz" is contained in "cn/gd". Giant ISPs can also be treated as topology oriented containers. For example, China Telecom, as a top level container "ct", contains "ct/gd", which further contains "ct/gd/sz".

As LPM is used in FIB matching, a routing entry for a container may only propagate within the network domain of its access containers. For example, a routing entry for "cn/gd" does not need to propagate out of "cn". In this case, a core router in the network domain of "us" with only a routing entry "cn" can forward all the packets with "cn" as a container prefix. Obviously the size of FIB can be greatly reduced due to the recursive aggregation of topology oriented containers. Therefore, the routing entries created by topology oriented containers are the basis for FIB compression.

3.2.2 Non-topology oriented containers

According to the scale-free property of the Internet, we further divide non-topology oriented containers into popular ones and non-popular ones.

The majority of the non-topology oriented containers have non-popular content and low visiting volume, such as small companies and organizations, home networks, and personal digital devices. The large number of this type of containers is the major cause of the routing scalability problem. Since these containers do not have to propagate outside of their access containers, we can greatly reduce the size of FIBs in core routers with access containers. For example, the corporate network of a company "hostsrv.com" locates in three different places, "cn/gd", "cn/beijing", "us/ca", which can be seen as three topology oriented containers that provide access service to "hostsrv.com". The route to "hostsrv.com" exists only inside these three containers, and any outside router can use these three containers to assist packet routing in order to reach "hostsrv.com".

A small portion of non-topology oriented containers has several orders of magnitude higher visiting volume than others, such as large corporations (e.g., huawei.com) and large portal websites (e.g., sina.com). The small number of these frequently visited containers does not cause scalability problems for core routers.

To conclude, with the container assisted naming scheme and routing, the size of FIBs in core routers is determined by the number of "aggregated topology oriented containers" and "frequently visited non-topology oriented containers", which could be smaller than the size of routing table in current Internet's core routers.

4 Container Assisted Mobility

4.1 Container Assisted Terminal Mobility

By terminal we mean either consumer side terminal devices or data source side terminal devices or hosts. In the scenario where a data source terminal is static and a consumer terminal is moving, the consumer can resend requests to fetch the latest data packets after the movement.

In the scenario where the data source is moving, a carried container in data request packets can assist the forwarding during the data source movement.

```
                                       .---.
              ,,_.----------.          |E1 |      ,,_.----------.
            ,-''     .---.   `--.      `---'    ,-''            `--.
          ,'         |E2 |       `.           ,'                    `.
         /           `---'         \         /                        \
        (   A               O---O   )       (                     B    )
                            | C |<---------`---------                 /
          `.                O---O,'         `.                      ,'
            '--.                _.-'          '--.              _.-'
                `----------''                     `----------''
```

Figure 3. Data source terminal movement

As Figure 3 shows, if the data source in container C moves within its access container B, its routing update is contained within the access container, and there is no routing update out of container B.

If the data source moves out of its access container B to a new container A, it needs to register a new route in the new access container A, and updates the access container information in the resolution system. After this, its data consumers or on-path routers can acquire the latest base container (i.e., container A) with container resolution. The register process can refer to [Huawei-Resolution].

As shown in Figure 3, during the movement, if a data consumer E2, which resides within the same access container (i.e., container A) as the data source in container C, sends a request to retrieve the data, the request packet can be forwarded straightly to the mobile source.

However, if a router E1 outside the access container A receives such request, it cannot find the route directly by the carried container in the name (i.e., container C). Therefore, the router sends resolution request to obtain the data source's access container. After gets the access container A, the router forwards the requests to A firstly, and then to destination C by content object name or carried base container(s).

4.2 Container assisted network mobility

A mobile container can provide access to a set of containers that moves together with it, for example, a train, airplane, or ship can provide access service to its passengers. This can be seen as network mobility.

```
                                    .---.
              _.-.---.----.         |E1 |   ,---------.
          ,--''  |E2 |     `---.    `---'  ,'__        `.
        ,'       `---'____     `.      __..-''          )
       /         ,-''     ``-.  __.--''    `.    D     ,'
      / A      ,'  .---.  __.--:'  \         `-+--------'
     (         | B | E3|        |   )         /
      \        \   `---' O--O   '  /         /   ,---------.
       \        `.       |C | ,'  /         /  ,'          `.
        `.       `-..____O--O' ,'          /  (              )
          '---.            i.--'          /    `.'   F     ,'
               `----------'' `.          /    .' `---------'
                               `.       /   .'
                                 `. ,-+---.'
                                  ;:       `.
                                 (     G     )
                                  `.       ,'
                                    `-----'
```

Figure 4. Network mobility

As Figure 4 shows, when container B moves, the processing is similar as the data source movement in previous case. If container B moves insides of its access container, i.e., container D, routing update only propagates within this container. If B moves out of its access container, it needs to register a new routing to its new access container, e.g., container A, and updates its new access container relationship in container resolution system. When container B moves across several containers, it only needs to update its access containers (e.g., from D to A) in the resolution system, while all contained containers (e.g., container E3 and C) within itself do not need to do any update since their routing in the network (inside of container B) and their access container relationships do not changed. With this container assisted routing, the updating and querying frequency to the resolution system can be significantly reduced.

Consider the mobile network B as a high-speed train. Along with its inside containers (E3 and C), it moves from original access container D to a new access container A. During this movement, if a consumer within the same mobile network E3 requests for data in contained container C, the request is forwarded directly.

If a consumer in container E2, which is outside B, requests data in container C, the router in the access container A cannot forward the request directly by the content name. It then sends resolving request to the resolution system. With the resolved container B, the router can forward the request to the mobile network B, which in turn forwards to data source C with content name or carried container.

If a consumer in container E1, which is outside the access container of the mobile network B, requests data in contained container C, it needs two-level resolution: it obtains the mobile network B with the first level resolution, and the network's new access container A with the second. Routers can route the request with the second resolution result (i.e., container A), and then with the network container B, and finally to the destination container C with content object name in request or carried base container(s). Note that the number of resolution level is not restricted in our scheme, which is decided by access relationships among containers.

5 IANA Considerations

No IANA consideration for this draft.

6 Conclusions

7 References

7.1 Normative References

7.2 Informative References

[Cisco-Name]
NDN and IP Routing, Can it Scale?
http://trac.tools.ietf.org/group/irtf/trac/raw-
attachment/wiki/icnrg/IRTF%20-

%20CCN%20And%20IP%20Routing%20-%202.pdf

[CCN] V. Jacobson et al. Networking named content. Proceedings of the 5th ACM International Conference on Emerging Networking Experiments and Technologies (CoNEXT 2009). NY: ACM; 2009; 1-12.

[DONA] T. Koponen, M. Chawla, B.-G. Chun, A. Ermolinskiy, K. H. Kim, S. Shenker, and I. Stoica. A Data-Oriented (and Beyond) Network Architecture. In Proc. of SIGCOMM, 2007.

[Huawei-Resolution] Draft-ietf-icnrg-icn-container-resolution-system-00.txt.

Authors' Addresses

Chunfeng Yao
Huawei Technologies
Huawei Building,
No.156, BeiQing Rd., Haidian District
Beijing, 100095
P.R. CHINA

EMail: chunfengyao@huawei.com

Rong Wang
Huawei Technologies
Huawei Building,
BanTian, LongGang District
Shenzhen, GuangDong 518129
P.R.CHINA

EMail: WANG.RONG@huawei.com

Yuanzhe Xuan
Huawei Technologies
Huawei Building,
BanTian, LongGang District
Shenzhen, GuangDong 518129
P.R.CHINA

EMail: xuanyuanzhe@huawei.com

Zhefeng Yan
Huawei Technologies
Huawei Building,
BanTian, LongGang District
Shenzhen, GuangDong 518129
P.R.CHINA

EMail: yanzhefeng@huawei.com