Workgroup: Network Working Group
Internet-Draft: draft-kjsun-lisp-dyncast-00
Published: 9 August 2021
Intended Status: Informational
Expires: 10 February 2022
Authors: K. Sun Y. Kim
Soongsil University Soongsil University

LISP Support for Dynamic Anycast Routing

Abstract

This draft describes the LISP-based architecture and solutions for supporting dynamic anycast (Dyncast) routing.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

2. Terminology
3. Architecture Overview
4. Addressing Dyncast Requirements with LISP
4.1. Anycast-based Service Addressing
4.2. Instance Affinity
4.3. Encoding and Signaling of Metric
4.4. Dynamic Routing Decisions based using Metrics
4.5. Supporting Service Dynamism
5. Security Considerations
6. References
6.1. Informative References
Authors' Addresses

1. Introduction

In an environment where equivalent services are distributed in multiple geographic locations, Dynamic-Anycast (Dyncast) enables to perform resource-efficient anycast routing. To support Dyncast, according to [draft-liu-dyncast-ps-usecases], a unique service identifier that can be assigned to multiple instances in multiple edge environments should be able to be mapped as an actual routable unicast address. Since this concept is similar to the Location/ID separation method already used in the LISP design basis, the LISP protocol can be considered as one of the candidate protocols that can implement Dyncast. This draft is proposed to design the LISP-based architecture for Dyncast and analyze the extension method of LISP to meet the requirements defined in [draft-liu-dyncast-reqs] for realizing dynamic anycasting between different LISP sites.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document is to be interpreted as described in [RFC2119]. This document uses the terminology described in [RFC6830], [draft-liu-dyncast-ps-usecases], [draft-liu-dyncast-reqs].

3. Architecture Overview

Figure 1 describes the Dyncast architecture based on LISP. In the LISP architecture [draft-ietf-lisp-introduction-13], each edge network has one or more LISP routers deployed. For anycast address, [RFC6830] defines that anycast address can be assigned for both Endpoint ID (EID) and Routing Locator (RLOC) within each of their address spaces. In this draft, we called EID for dynamic anycasting as Dyncast Service ID (DSID), which is assigned to equivalent services across the multiple LISP sites. Similar to the common EID definition, the DSID cannot be routed globally by itself, and the same DSID cannot be assigned to different services. In order to forward a packet destined for a DSID between LISP edges, the addresses of the LISP Egress Tunnel Router (ETR) are used as RLOC, which operates as a Dyncast Binding ID (DBID) from a Dyncast perspective. Map-server/resolver of the LISP control plane can manage mapping information for DSID-RLOC mappings together with existing EID-RLOC mappings. Differences of DSID-RLOC from existing EID-RLOC mapping table, it that a single DSID can be mapped with multiple RLOCs from different edge sites together.

For resource-efficient forwarding decisions across multiple service instances, [draft-li-dyncast-architecture] defines Dyncast Metric Agent (D-MA) which collects metrics related network and service instances. Actual packet forwarding is handled in the Dyncast Router (D-Router) based upon collected metrics with maintaining instance affinity. In the LISP architecture, the D-Router function can be implemented on the LISP xTR, and the D-MA can be deployed as a separate component within the edge for managing service instances, or it can be deployed in combination with the LISP xTR. The LISP, control plane is logically centralized and it provides an interface with each LISP router to exchange mapping information. However, it does not mean that the LISP control plane is located in a single physical location, several mechanisms for distributing the mapping system already have been defined.

```
  LISP Edge                                    LISP Edge

+----------+                                  +----------+
|  Service |                                  |  Service |
| Instance |                                  | Instance |
|  (DBID)  |                                  |  (DBID)  |
+----------+                                  +----------+
     |           +---------------------+           |
     |           |  LISP Control Plane |           |
+----------+"""""+---------------------+"          |
|   D-MA   |    "  "                     "         |
+----------+   "  "                       "        |
     |        "  "                         "  +----------+
     |       "  "                           " |   D-MA   |
+----------+"   " +--------------------+     "+----------+
| LISP-xTR |DBID" |    Core Network    |DBID  | LISP-xTR |
|(D-Router)|----"-|    (RLOC-Space)    |------|(D-Router)|
+----------+    " +--------------------+      +----------+
     |          "           |DBID                  |
     |           "+--------------------+           |
     |            | LISP-xTR (D-Router)|           |
     |            +--------------------+           |
+----------+                |                 +----------+
|  Client  |           +----------+           |  Client  |
|  (EID)   |           |  Client  |           |  (EID)   |
+----------+           |  (EID)   |           +----------+
                       +----------+

                         LISP Edge
```

Figure 1: LISP-based Dyncast Architecture

Figure 2 shows an example of LISP-based Dyncast deployment where two services each deployed two instances at different edges. In this scenario, two services are assigned an RLOC according to the ETR address of the LISP site. Both Service_A and Service_B instances connected to ETR_2 are assigned RLOC2, which is the RLOC of ETR_2, as a binding ID. In the case of the edge where ETR_2 is located, as an edge composed only of service instances, the LISP Router function can be operated by being strongly coupled to the edge computing server. In this case, the D-MA function can be implemented on the ETR to insert service-instance-related metrics directly into the LISP protocol packet. In case that a service instance and a client co-exist like an edge where ETR_3 is located, the D-MA entity can be independently deployed proximity of the service instance is running, transparent from the LISP operation for clients. Mapping information update for DSID is performed through the LISP protocol Map-Register message, and service-instance-related metric can be delivered through in the LISP protocol header or other methods. A method of inserting service-instance-related metric information into the LISP protocol will be discussed later. When the ITR_1 receives a packet destined for the DSID of the service by service request from the Host_1, the ITR can acquire the RLOC mapped to the requested DSID from the LISP control-plane through the Map-Request message. At the control plane, it may select an proper DBID address on the collected metric information and return it to the ITR or return the DBID list of multiple service instances with metric information to the ITR so the ITR selects the proper DBID in the list. A method for determining an appropriate DBID will be discussed later.

```
                                                             Service_A
                                                              +------+
                  Map-Register              D-Router        +-|DSID_A|
                 (DSID_A, DBID2, <metric>) +-------+------+ | +------+
                 (DSID_B, DBID2, ,metric>) | ETR_2 | D-MA |-|
                                           +-------+------+ | +------+
                                               |            +-|DSID_B|
                         +------------------+  | RLOC2        +------+
  Host_1     D-Router    | +--------------+ |--+(DBID2)      Service_B
+--------+  +-------+    | |     LISP     | |
| EID_H1 |--| ITR_1 |----| | Control Plane| |
+--------+  +-------+    | +--------------+ |
                    RLOC1|    RLOC-Space    |--+ RLOC3
                         +------------------+  |(DBID3)
                                           D-Router      Host_2
                  Map-Register             +-------+   +--------+
                 (DSID_A, DBID3, <metric>) | ETR_3 |---| EID_H2 |
                 (DSID_B, DBID3, <metric>) +-------+   +--------+
                                               |
                                            +------+
                                            | D-MA |
                                            +------+
                                               |
                                         +-----+-----+
                                         |           |
                                     +------+     +------+
                                     |DSID_A|     |DSID_B|
                                     +------+     +------+
                                     Service_A    Service_B
```

Figure 2: LISP-based Dyncast Example Scenario

4. Addressing Dyncast Requirements with LISP

4.1. Anycast-based Service Addressing

To support Dyncast routing, the system must provide a method for searching a service identifier allocated as an anycast address and mapping it to a specific unicast address. From this point of view, the LISP is a suitable protocol for separating ID/Location of service and managing mapping information. When the system allocates the same D-SID to each service instance for service equivalency, the LISP can define an anycast address space for the DSID and assign it to service instances created across multiple sites. Also, the DBID can be used as an RLOC address of LISP xTR that can be routed between edges as unicast. That is, it is necessary to define a separate space for anycast address within the existing EID space and to allocate it in advance so that it can be used in all edge networks where the service instances are located. In the LISP definition, the EID assigned to each service has a globally unique value and, in particular, [RFC6830] defines that anycast address can be assigned within an EID or RLOC block spaces. In each LISP site, same as the EID which is defined to enable internal routing, the DSID can be able to be routed without the DBID encapsulation process to the EID within a single site.

4.2. Instance Affinity

For Dyncast routing, it is required that the system must set "Instance Affinity" for one or several service requests to provide routing to the same service instance for the same flow. In LISP, the RLOC mapping information for the destination EID is stored in a local cache called Map-cache in the ITR for a certain period of time, and it is maintained for a set time-to-live (TTL) time. Therefore, mapping information for a specific service once requested from a client is generally maintained in the ITR until the corresponding session expires and can be delivered to the RLOC stored in the map-cache entry. However, in order to have a flexible selection of service instances between different flows at the same point, it is additionally required to assign different RLOCs for different flows depending on metrics dynamically changed. For that, it is necessary to enhance ITR Map-cache to maintain destination RLOC for each flow. In addition, although the general TTL value in LISP ITR is defined as 24 hours, in Dyncast the system requires a shorter TTL time for changing network path depending on dynamically updated network-related and service-instance-related metrics.

4.3. Encoding and Signaling of Metric

In Dyncast routing, the one of most important requirements is that it should be able to collect various metrics of service-instances-related as well as network-related, and include them in-network routing decisions. For that, it is necessary to define how to collect these metrics and forward them, and also where to make a decision. In the LISP environment, since that the entire EID-RLOC mapping information is managed in the control plane, one possible scenario is that the D-MA function which collects service-instance-related metrics updates them to the DSID mapping entry in the LISP control plane. For that, it can be used an encoding method proposed in [draft-farinacci-lisp-name-encoding] that defines to insert specific information such as parameters for a specific EID or RLOC using an ASCII string. Using that, it is possible to encode a string that is pre-defined of a specific metric to interpret in the control plane and send a Map-Request message so that the control plane can select an appropriate RLOC based on it. In order to insert service-instance-related metrics, the D-MA must forward the DSID of the requested service to the LISP ITR so that the metric can be inserted into the header of the Map-Register message. This metric information encoded into the Map-Register message can help the LISP control plane to uses a mechanism to make a routing decision based on the metric information of the requested or updated BSID.

4.4. Dynamic Routing Decisions based using Metrics

The Dyncast system is required that in must make routing decisions for all service requests, and this must be done under an understanding of all metrics. Routing decisions in the LISP can be done in the control plane or ITR by specifying priority and weight values for each RLOC. In case that routing decisions are made in the control plane, the Map-Resolver dynamically sets the priority and weight values of each mapped DBIDs collected from D-MAs, selects a proper DBID based on them, and forward it to the requested ITR using the Map-Reply message. However, since this centralized approach may not be calculated based on point of requested ITR, the actual routing path may not be optimal. In case that routing decision is determined at the ITR, the LISP control plane may return one or more DBID values for the requested DSID to the ITR, including priority and weight values based on the collected metrics. After receiving multiple DBIDs, the ITR stores them in map-cache entry and selects an appropriate one to forward the data packet. For that, a mechanism for estimating appropriate priority and weight values based on both network-related and service-instance-related metrics is required for the control plane or ITR.

In the Dyncast architecture described in [draft-li-dyncast-architecture], the D-Router collects metrics by exchanging metric information of the service identifier between another edge D-Routers and make a decision itself. This approach can minimize the signaling for routing decisions by decentralizing the authority for the anycast routing decision to an entity in the actual packet path, but the signaling for collecting metrics between each D-Router is bound to increase. In contrast, when the LISP is used, it can reduce effectively signaling of collecting metrics from the ITR since that the mapping information for D-SID and D-BID can be managed in a centralized control plane.

4.5. Supporting Service Dynamism

For service dynamism, the Dyncast system should support different selections for each flow according to a dynamically changing metric while considering various requirements in the selection of a service instance. As mentioned in Section 4.2, if the map-cache can be maintained for each flow, the forwarding path can be dynamically changed to the different service instances by allocating target DBID to the map-cache entry per-flow according to dynamic changes of metrics. In order to refresh the DSID-DBID mapping upon changing metric, the Solicit Map-Request message can be used to update so that the ITR can update the weight and priority for the DBID which is already received from the Map-server. Additionally, as proposed in [draft-farinacci-lisp-telemetry], telemetry data can be collected between Encapsulating/Decapsulating xTRs of the current flow, which is expected to be used for dynamic service path reselection.

5. Security Considerations

TBD

6. References

6.1. Informative References

[draft-farinacci-lisp-name-encoding]
Farinacci, D., "LISP Distinguished Name Encoding", May 2021, <https://datatracker.ietf.org/doc/draft-farinacci-lisp-name-encoding/>.

[draft-farinacci-lisp-telemetry] Farinacci, D., Ouissal, S., and E. Nordmark, "LISP Data-Plane Telemetry", May 2021, <https://datatracker.ietf.org/doc/draft-farinacci-lisp-telemetry/>.

[draft-ietf-lisp-introduction-13] Cabellos, A. and D. Saucez, "An Architectural Introduction to the Locator/ID Separation Protocol (LISP)", April 2015, <https://datatracker.ietf.org/doc/draft-ietf-lisp-introduction/>.

[draft-li-dyncast-architecture] Li, Y., Iannone, L., Trossen, D., and P. Liu, "Dynamic-Anycast Architecture", February 2021, <https://datatracker.ietf.org/doc/draft-li-dyncast-architecture/>.

[draft-liu-dyncast-ps-usecases]
Liu, P., Willis, P., and D. Trossen, "Dynamic-Anycast (Dyncast) Use Cases & Problem Statement", February 2021, <https://datatracker.ietf.org/doc/draft-liu-dyncast-ps-usecases/>.

[draft-liu-dyncast-reqs] Liu, P., Willis, P., and D. Trossen, "Dynamic-Anycast (Dyncast) Requirements", February 2021, <https://datatracker.ietf.org/doc/draft-liu-dyncast-reqs/>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119, March 1997, <https://datatracker.ietf.org/doc/rfc2119/>.

[RFC6830] Farinacci, D., Fuller, V., Meyer, D., and D. Lewis, "The Locator/ID Separation Protocol (LISP)", RFC 6830, January 2013, <https://datatracker.ietf.org/doc/rfc6830/>.

Authors' Addresses

Kyoungjae Sun
Soongsil University
369, Sangdo-ro, Dongjak-gu
Seoul
06978
Republic of Korea

Phone: +82 10 3643 5627
Email: gomjae@dcn.ssu.ac.kr

Younghan Kim
Soongsil University
369, Sangdo-ro, Dongjak-gu
Seoul
06978
Republic of Korea

Phone: +82 10 2691 0904
Email: younghak@ssu.ac.kr