Network Working Group
Internet-Draft
Intended status: Experimental
Expires: October 11, 2008

S. Brim
N. Chiappa
D. Farinacci
V. Fuller
D. Lewis
D. Meyer

April 9, 2008

LISP-CONS: A Content distribution Overlay Network Service for LISP
draft-meyer-lisp-cons-04.txt

Status of this Memo

Abstract

The Content distribution Overlay Network Service for LISP (LISP-CONS) is a protocol for distributing identifier-to-locator mappings for the Locator/ID Separation Protocol (LISP). LISP-CONS is not a routing protocol. LISP-CONS is designed to scale by using a hierarchical content distribution system comprised of Tunnel Routers, Content Access Resources, and Content Distribution Resources.

Table of Contents

1. Requirements Notation

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 [RFC2119].

2. Introduction

The Content distribution Overlay Network Service for LISP, or LISP-CONS, is a control-plane protocol for distributing identifier-to-locator mappings for the Locator/ID Separation Protocol (LISP) [LISP]. The properties of such a "locator/id split" have been discussed in depth in various venues dating back to [CHIAPPA] and [RFC1498], and as such will not be reviewed here. Rather, the reader is referred to the above references for an outline of the various benefits that may be realized by separating the functionality of IP addresses into separate Endpoint Identifier and Routing Locator name spaces.

LISP-CONS operates on a distributed Endpoint Identifier-to-Routing Locator (EID-to-RLOC) database. This database is distributed among the authoritative Answering Content Access Resources (Answering-CAR). An Answering-CAR (aCAR) advertises "reachability" for its EID-to-RLOC mappings through a hierarchical network of Content Distribution Resources (CDRs) (but importantly, not the mapping itself), and responds to mapping requests from the system. A CAR may also request mappings from the system (this a Querying-CAR, or qCAR). Ingress Tunnel Routers (ITRs) connect to one or more qCARs to query the system for EID-to-RLOC bindings; the qCAR then queries the system on behalf of the ITR. These queries follow the overlay network to the authoritative aCAR, which responds with the mapping. This response may then be cached by the 'local' CAR. Finally, note that neither a qCAR or aCAR need to hold the entire EID-to-RLOC database. Rather, the EID-to-RLOC translations are explicitly pulled by the ITRs by querying one or more of its connected qCARs.

Note that LISP-CONS is not designed for the "fast-mobility" case. That is, it is envisioned that the mappings distributed by LISP-CONS are reasonably static. LISP-CONS is also not designed to carry Locator Reachability status information; see [LISP] for details on how LISP determines locator reachability.

LISP-CONS seeks to control the "state * rate" scaling properties of the mapping service by first observing that the host mapping state is likely to be quite large (some estimates put the size of this database to be on the order of 10^10 hosts). As a result, even with aggressive aggregation, the "rate" of change of the mapping database

must be kept small. LISP-CONS manages the rate problem by distributing highly aggregated information about the location of the EID-to-RLOC mappings (which are assumed to change at low frequency) over a peering network. The peering network is comprised of ITRs, CARs and CDRs.

In summary, LISP-CONS is a hybrid "push/pull" protocol in which information about the existence of a particular mapping is "pushed" at the higher levels of the aggregation hierarchy, while the actual EID-to-RLOC mappings are "pulled" from the network elements at the lowest level of the hierarchy. In particular, LISP-CONS carries mapping requests and replies to and from the lowest level of the hierarchy where the EID-to-RLOC mappings reside.

While this draft focuses on a router-based solution, there is no architectural reason that LISP-CONS functionality could not be implemented in other devices (i.e., hosts). However, in keeping with the architectural direction taken by the LISP data-plane proposal [LISP], LISP-CONS is based on the the theory that building the solution into the network should facilitate incremental deployment of the technology on the Internet. In order to minimize the required investment in deployment of new hardware, it is assumed that much, if not all, the initial implementation will be in routers. Finally, while the detailed protocol specification and examples in this document assume IP version 4 (IPv4), there is nothing in the design that precludes the use of the same techniques and mechanisms for IPv6.

The remainder of this document is organized as follows: Section 3 provides the set of definitions that are used in this document, and Section 4 provides an overview of LISP-CONS operation. Section 5 describes the LISP-CONS protocol, and Section 6 provides details of the LISP-CONS message types. Section 7 outlines operational considerations, Section 8 discusses locator reachability, and Section 9 considers the interaction of LISP-CONS with mobile nodes. Section 12 outlines security considerations for LISP-CONS.

Finally, this proposal (as well as the LISP data-plane proposal) was stimulated by the problem statement effort at the IAB Routing and Addressing Workshop (RAWS) [RFC4984], which took place in Amsterdam in October 2006.

3. Definition of Terms

The LISP-CONS protocol operates on two name spaces and is comprised of four network elements. This section provides high-level definitions of the LISP-CONS name spaces, network elements, and

message types.

3.1. LISP-CONS Name Spaces

Endpoint ID (EID): A 32- or 128-bit value used in the source and destination fields of the first (most inner) LISP header of a packet. A packet that is emitted by a system contains EIDs in its headers and and LISP headers are prepended only when the packet hits an Ingress Tunnel Router (ITR) on the data path to the destination EID.

In LISP-CONS, EID-prefixes MUST BE assigned in a hierarchical manner (in power-of-two or larger chunks) such that they can be aggregated either by Content Access Resources or Content Distribution Resources (see below). In addition, a site may have site-local structure in how EIDs are topologically organized (subnetting) for routing within the site; this structure is not visible to the global routing system.

EID-Prefix Aggregate: A set of EID-prefixes said to be aggregatable in the [RFC4632] sense. That is, an EID-Prefix aggregate is defined to be a single contiguous power-of-two EID-prefix block. Such a block is characterized by a prefix and a mask.

Routing Locator (RLOC): The IP address of an egress tunnel router (ETR). It is the output of a EID-to-RLOC mapping lookup. An EID maps to one or more RLOCs. Typically, RLOCs are numbered from topologically-aggregatable blocks that are assigned to a site at each point to which it attaches to the global Internet; where the topology is defined by the connectivity of provider networks, RLOCs can be thought of as Provider Aggregatable (PA) addresses.

EID-to-RLOC Mapping: A binding between and EID and the RLOC-set that can be used to reach the EID. We use the term "mapping" in this document to refer to a EID-to-RLOC mapping.

3.2. LISP-CONS Network Elements

LISP-CONS consists of the four network element types described below. Peering connections between these element types use RLOCs so that the underlying routing system can keep the LISP-CONS peering connections up (i.e., to avoid circular dependencies on the mapping system). Each peering connection is required to be configured with a keyed-hash message authentication code (HMAC) key. A connection MUST NOT be established without the TCP HMAC option included.

Content Distribution Resource (CDR): A CDR provides aggregation of EID prefix lists, propagation of EID-prefix lists to parent CDRs, and routing of mapping requests to and from CARs.

There may be several levels of aggregation of CDRs. CDRs do not themselves carry EID prefix to RLOC mappings. CDRs are arranged in a hierarchical manner in order to enable aggressive aggregation of EID-prefixes.

Content Access Resource (CAR): A CAR fills one or both of the following roles:

Answering-CAR (aCAR): A CAR is the source of authority for one or more EID prefix to RLOC mappings which which it has been administratively configured, and responds to Map Requests for these EID-to-RLOC mappings. Each aCAR provides to parent CDRs a list of prefixes that it is responsible for, but not the mappings themselves.

In particular, aCARs peer with CDRs to propagate aggregated information about how to find a particular EID-to-RLOC mapping upward (but importantly, not the mapping itself). However, aCARs do not peer with other CARs. The primary difference between the aCAR and CDR is that a CAR maintains two databases: A EID-to-RLOC mapping database, and a EID-prefix database. A CDR maintains only an EID-prefix database.

Querying-CAR (qCAR): A CAR that generates Map-Request messages on behalf of one or more of its ITR peers (see below). Note that qCAR has peering connections with ITRs whereas an aCAR does not have to. Finally, both functionalities (qCAR and aCAR) MAY be co-located in the same device. In particular, qCAR MUST also be an aCAR, while an aCAR need not be a qCAR.

Egress Tunnel Router (ETR): A router that accepts an IP packet where destination address in the "outer" IP header is one of its own RLOCs. The router strips the "outer" header and forwards the packet based on the next IP header found. In general, an ETR receives LISP-encapsulated IP packets from the Internet on one side and sends decapsulated IP packets to site end-systems on the other side.

Ingress Tunnel Router (ITR): A router which accepts an IP packet with a single IP header (more precisely, an IP packet that does not contain a LISP header). The router treats this "inner" IP destination address as an EID and performs an EID-to-RLOC mapping lookup. The router then prepends an "outer" IP header with one of its globally-routable RLOCs in the source address field and the

result of the mapping lookup in the destination address field. Note that this destination RLOC may be an intermediate, proxy device that has better knowledge of the EID-to-RLOC mapping closest to the destination EID. In general, an ITR receives IP packets from site end-systems on one side and sends LISP-encapsulated IP packets toward the Internet on the other side. ITRs may also have TCP connections to qCARs in order to send mapping requests and receive replies (noting that a qCAR, an aCAR, and an ITR may be co-located).

3.3. Relationship Between LISP-CONS Network Elements

Each LISP-CONS device is known by a single identifier, which is used for peering from all peers, and in path-vector (PV) lists. This identifier MAY be an IP address. An implementation SHOULD use a loopback address for this purpose. Note that this address MUST be routable by the core routing system.

LISP-CONS network elements peer with each other in one of three peering relationships: parent, child, or sibling. The relationship is carried in the LISP-CONS OPEN message (see [LISP]). The permitted peering relationships are as follows:

- ITRs exist at lowest (unnumbered) level in the peering hierarchy, and peer only with one or more CARs. An ITR MUST NOT peer with another ITR or with a CDR.

- CARs exist at level 0 in the peering hierarchy, and peer only with parent CDRs or with a child ITR. A CAR MUST NOT peer with another CAR; this rule allows the aCARs to aggregate EID prefixes as low in the hierarchy as possible. Note that this rule also means that mapping requests and replies are routed over the peering topology, not directly between the CARs.

- CDRs exist at level 1 (and above) and aggregate EID-prefixes learn from its aCAR peerings. When a two CDRs start their peering connection, if one is a parent, the other MUST BE a child. Otherwise, they both MUST BE siblings.

- If any of these checks fail, the peering connection MUST NOT be established.

4. Overview of Operation

LISP-CONS constructs a multi-level content distribution overlay which achieves scalability by imposing a strict aggregation hierarchy on the participating elements. The LISP-CONS hierarchy consists of ITRs

the bottom of the hierarchy, CARs at level 0, and CDRs at levels 1 and above; this is depicted in Figure 1. Each level of the hierarchy is a strict tree. That is, there are no transit loops in the hierarchy; redundancy is achieved by meshing CDR connectivity within in a single level of the hierarchy, and the LISP-CONS protocol assures that message flow is loop-free.

In LISP-CONS, the EID-to-RLOC mappings are held in the aCARs, while the CDRs maintain information about how to find the aCAR holding a particular EID-to-RLOC mapping. That is, the Push-Add and Push-Delete messages (see [LISP]) only contain EID-prefixes (i.e., Locator-sets are not included in these messages and are not stored in the CDRs).

In general, LISP-CONS uses network element redundancy to avoid mapping database inconsistencies that may arise in those cases in which a CAR or CDR crashes. Similarly, connectivity outages are avoided by configuring a redundant underlying topology.

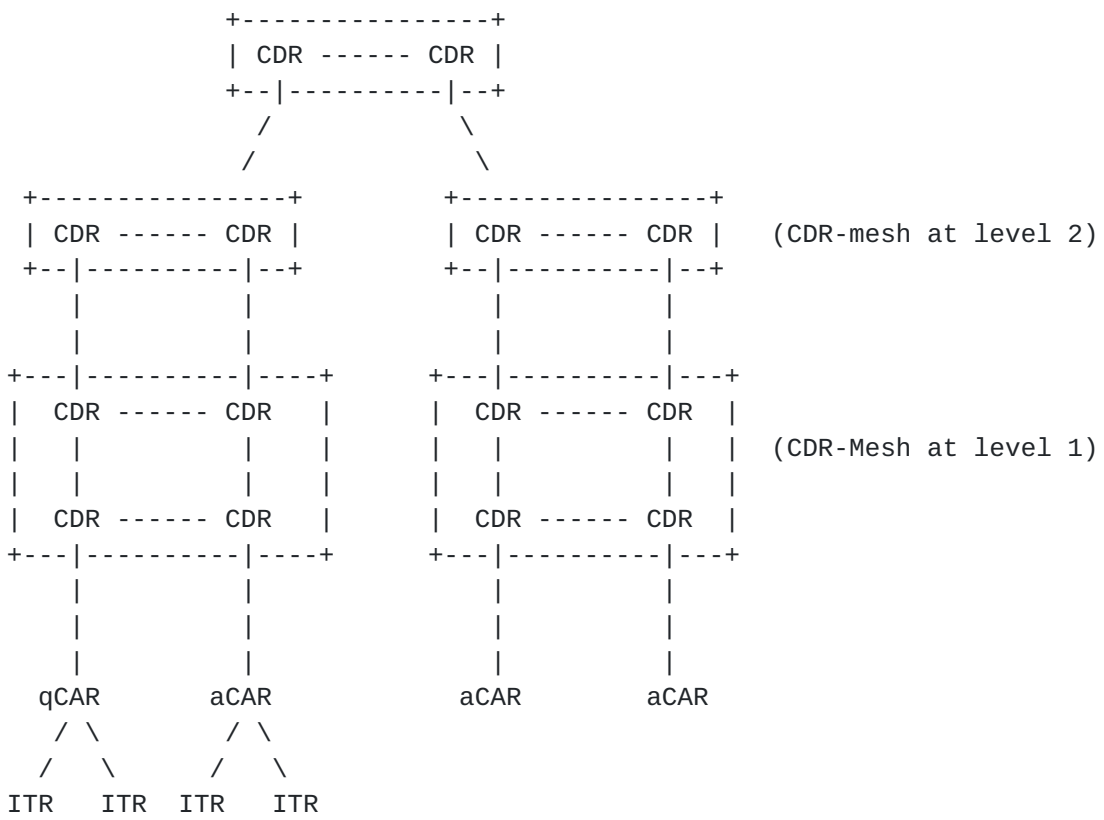

Figure 1: LISP-CONS Hierarchy

Figure 2 depicts the details of the first three levels of hierarchy.

Note that there are no horizontal TCP connections between the ITRs or between the CARs. Note that qCARs (abbreviated "Req-CAR") peer with the ITRs, while the aCARs may not. The CDRs at level 1 are meshed so that the two aCARs can aggregate to the same mesh level.

Note that to avoid request and reply black-holes, all CDRs that are responsible for a segment of the address space must be siblings (i.e., at the same level).

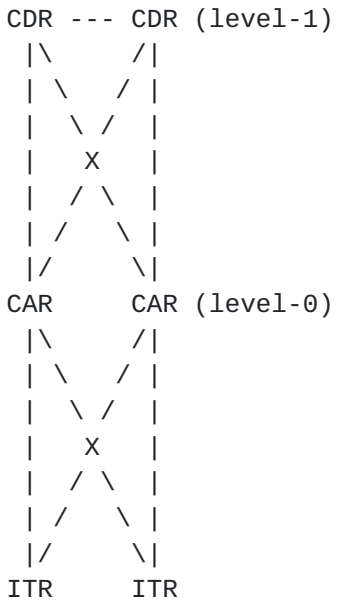

Figure 2: LISP-CONS Hierarchy Detail

LISP-CONS operates as follows: An aCAR receives EID-to-RLOC mappings by administrative configuration. The aCARs aggregate these EID-prefixes into power-of-two less specific EID-prefixes, and "push" the aggregated EID-prefixes to their (parent) CDRs in Push-Add messages (see [LISP]). CDRs then flood the Push-Add messages to their sibling CDRs. Note that the Push messages contain EID-prefix reachability information, not locator sets.

If a CDR is a child, it then pushes the aggregate for the EID-prefix (i.e., the aggregate that "covers" the EID-prefix) to its parent CDRs. This CDR MUST also originate the default EID-prefix 0.0.0.0/0 or 0::0/0 (this allows Requests and Replies to flow up and down the aggregation hierarchy). This default is contained within the level of the sibling mesh. Note that aggregates MUST only be generated when the components of the aggregate are all longer prefixes than the aggregate (and importantly, NOT equal in length). For example, a CDR MUST NOT generate an aggregate such as A.B.0.0/16 if it has not heard a A.B.*.0/24 from either a child or sibling peer.

When an ITR needs a mapping, it sends a Map-Request message to its directly connected qCARs. If any of those CARs have cached the requested mapping, the result is immediately returned to the ITR. Otherwise, the Map-Request message is routed through the CDR hierarchy to the aCAR which holds the mapping. That CAR then returns the mapping in a Map-Reply message (which is routed over the peering topology) to the qCAR, which then forwards it on to the requesting ITR.

Finally, note that this type of advertisement hierarchy allows EID lookups to have lower Round Trip Times (RTTs) when the EID-prefix is "close" (in the EID allocation hierarchy) to the site's attached CAR. However, for scalability reasons, a request may have to travel extra hops to get an EID-prefix that can only be obtained by going up the tree (and in the worse case, by going to the top of the hierarchy and down to the aCAR that hold the mapping).

5. The LISP-CONS Protocol

This section describes the LISP-CONS protocol in detail, starting with how LISP-CONS builds a distributed mapping database, how an ITR queries the database, and how the database is maintained.

LISP-CONS operates on three different data structures:

EID-to-RLOC Database: The EID-to-RLOC mapping database, which is administratively configured and held in the aCARs.

Mapping Cache: The Mapping Cache (hereafter cache) is the result of a Map-Request and is stored in the ITRs and qCARs.

EID-Prefix Table: The EID-Prefix table is used to route Map-Requests and Map-Replies in the overlay network. It is stored only by CDRs, and associates an EID-prefix with a 64-bit sequence number, a path-vector, and a priority and weight (to facilitate later aggregation, if possible).

5.1. Building the LISP-CONS Database

When an aCAR is configured with an EID-to-RLOC mapping, it checks to see if it can aggregate the just learned EID-prefix with any of the other EID-prefixes it has been configured with. The CAR then sends ("pushes") the EID-prefix (or an aggregate, if possible) to its parent CDR in a Push-Add message.

An aCAR generates an aggregate when it has at least one more specific prefix that matches the aggregate. A more specific prefix of an

aggregate is when the high-order bits of the more-specific prefix and the high-order bits of the aggregate are the same. The number of bits tested is the mask-length of the aggregate.

When a more-specific prefix is added to the EID-prefix table, the corresponding aggregate is sent in a Push-Add message from a child peer to a parent peer in a different level.

Push-Add messages contain an EID-prefix, and Originator Address, a 64-bit sequence number, and a PV that records the path the message took in the CDR level (see [LISP]). Note that the Originator Address is an EID used to route a Reply back to the requesting ITR The PV list will always contain Locators.

When a CDR receives a Push-Add message, it first checks to see if the sequence number for the EID-prefix is numerically larger than what it has stored for the EID-prefix. If it is not, the message is dropped. Otherwise, the CDR next checks for its own address in the PV. If it exists, the message is discarded. Otherwise, the CDR stores EID-prefix and the associated PV. Note that the CDR can store all different combination of PVs or just the shortest path ones. If the CDR has one or more parent peerings configured (i.e., the CDR is a child), it will aggregate this EID-prefix with other EID-prefixes into a more coarse EID-prefix. The CDR does not need to advertise anything to lower-level CDRs because child peers will auto-generate a default EID-prefix into their level simply due to having a child-parent peering relationship.

When a CDR sends a Push-Add message to a parent, the stored PV is not propagated to the parent in the aggregated EID-prefix; rather, it includes a one element PV which contains the address of the CDR originating the "aggregated push". It also includes a new sequence number, indicating that this is a different EID-prefix than the ones it has stored.

Finally, if a CDR is a child, it pushes a "EID-default" to its siblings. This Push message has EID-prefix 0.0.0.0/0 or 0::0/0 and a PV containing the address of the CDR that is sourcing the default.

5.2. Querying the LISP-CONS Database

Map Requests are routed along the LISP-CONS multi-level topology from requesting ITR to aCAR holding the requested mapping. The Map-Request message includes a PV which records the route traversed by the Map-Request message. This PV is used to control request routing and for debugging purposes.

When an ITR wants to query the LISP-CONS database for a mapping, it

prepares a Map-Request message, which is sent to one of its directly connected qCAR(s). The Map-Request message is routed over the peering topology to the aCAR that holds the mapping. If the qCAR has cached the mapping (perhaps from a previous request), in which case it returns the mapping immediately.

When a qCAR receives a Map-Request from from an ITR, it MAY respond immediately if it has the cached requested mapping. Otherwise, it MUST forward the Map-Request message to its parent CDRs. This CAR is identified by the Originator address in the Map-Request message (see [LISP]). The Originator address allows a replying CDR to forward a No-Map message (see [LISP]) back to the qCAR. This case arises when source-site is LISP-enabled (i.e., there is an ITR deployed), but the destination-site has not deployed LISP yet so there is no ETR.

When a Map-Request arrives at a CDR, the CDR first scans its PV for its address. If its address is present, it drops the packet. If its address is not present, it consults its EID-prefix table for the longest match "next-hop" towards the aCAR holding the mapping for the prefix. If a next-hop is found, the CDR appends its address to the PV, and forwards the Request to the next-hop.

When a Map-Request arrives at a CDR which cannot route it, a LISP-CONS No-Map message (see [LISP]) MUST BE sent back to the qCAR. This No-Map message is a signal that indicates that there is no mapping for the requested EID in the system, and is immediately communicated to the ITR.

When a Map-Request message arrives at an aCAR, it first queries its mapping database for the EID contained in the Map-Request message. If the mapping is found, it constructs a Map-Reply message (see [LISP]) containing the EID, the corresponding RLOC-set, and an PV containing its address appended to the reverse of the received PV. The CAR then sends the Map-Reply message over the peering topology to the qCAR (i.e., to the Originating CAR EID-Prefix in the Map-Request message).

If no mapping is found, the aCAR sends a Map-Reply with the requested EID and a Locator count of 0 back to qCAR. This creates a negative cache entry in the requesting ITR.

In LISP-CONS, the PV for Map-Request and Map-Reply messages are preserved across the hierarchy, while the PV lists carried in Push-Add and Push-Delete messages are not. As a result, LISP-CONS also has cross-level loop suppression.

5.3. Maintaining the LISP-CONS Database

While LISP-CONS is not a routing protocol (and as such when peering connections go down EID-prefix entries are not immediately withdrawn from the local EID-prefix table), it does uses a link-state-like sequence number scheme to detect changes in topology. Similarly, LISP-CONS uses a path vector scheme to detect and suppress message looping. There are four database maintenance cases to consider:

- An EID-Prefix Is Administratively Removed From The Infrastructure (Section 5.3.1)
- A CAR's Connectivity Changes (Section 5.3.2)
- A CAR Becomes Unreachable (Section 5.3.3)
- A CDR Becomes Unreachable (Section 5.3.4)

Each case is considered below.

5.3.1. An EID-Prefix Is Administratively Removed From The Infrastructure

EID-prefix mappings are removed from the LISP-CONS infrastructure by administrative configuration at the aCAR that was configured with the mapping. The CAR queries its EID-prefix database for the mapping. If no match for the EID-prefix exists, no further action is taken.

When all the more-specific prefixes that matches the aggregate are removed from the EID-prefix table, the aggregate is sent in a Push-Delete message from a child peer to a parent peer in a different level. The Push-Delete message behaves exactly like the Push-Add message, except that it removes the corresponding state along its path(s).

When a Push-Delete message arrives at a CDR, the CDR checks for its own address in the PV. If it exists, the message is discarded. Otherwise, the CDR queries its EID-prefix database for the EID-prefix in the received Push-Delete message. If it finds a matching entry, it removes the entry from its database, appends its address to the PV, and forwards the message to its siblings.

If the CDR is a child, it checks to see if the EID-prefix in the Push-Delete message is the last in an aggregate it had previously pushed to its parent CDR. If not, no further action is taken. Otherwise, the CDR computes a new aggregate (minus the prefix from the Push-Delete), sends a Push-Delete for the old aggregate to its parent, and sends a Push-Add with the new aggregate to its parent

CDR.

5.3.2. A CAR's Connectivity Changes

Changes in CAR connectivity are signaled by changes in the sequence numbers in a Push-Add messages. For example, in Figure 3, consider the case in which the D<->B TCP connection breaks. In this case, D sends a Push-Add with EID-Prefix EID/(n-1), sequence number, S+1, and path vector [D] (denoted push(EID/(n-1), S+1, [D])) to C. C aggregates the pieces of EID and forwards push(EID/n,S+1,[C,D]) to B. Now, before the failure, B had an entry in its EID-prefix table for EID/n with sequence number S and PV [D]. Since B sees a new push message originated by D with sequence number S+1, it knows the previous entry (EID/n,S,[D]) is no longer valid.

Similarly, A will see push messages with both [C,D] and [B,C,D] and with sequence number S+1, so it knows the existing entries ([B,D] and [C,B,D], with sequence number S) are both obsolete.

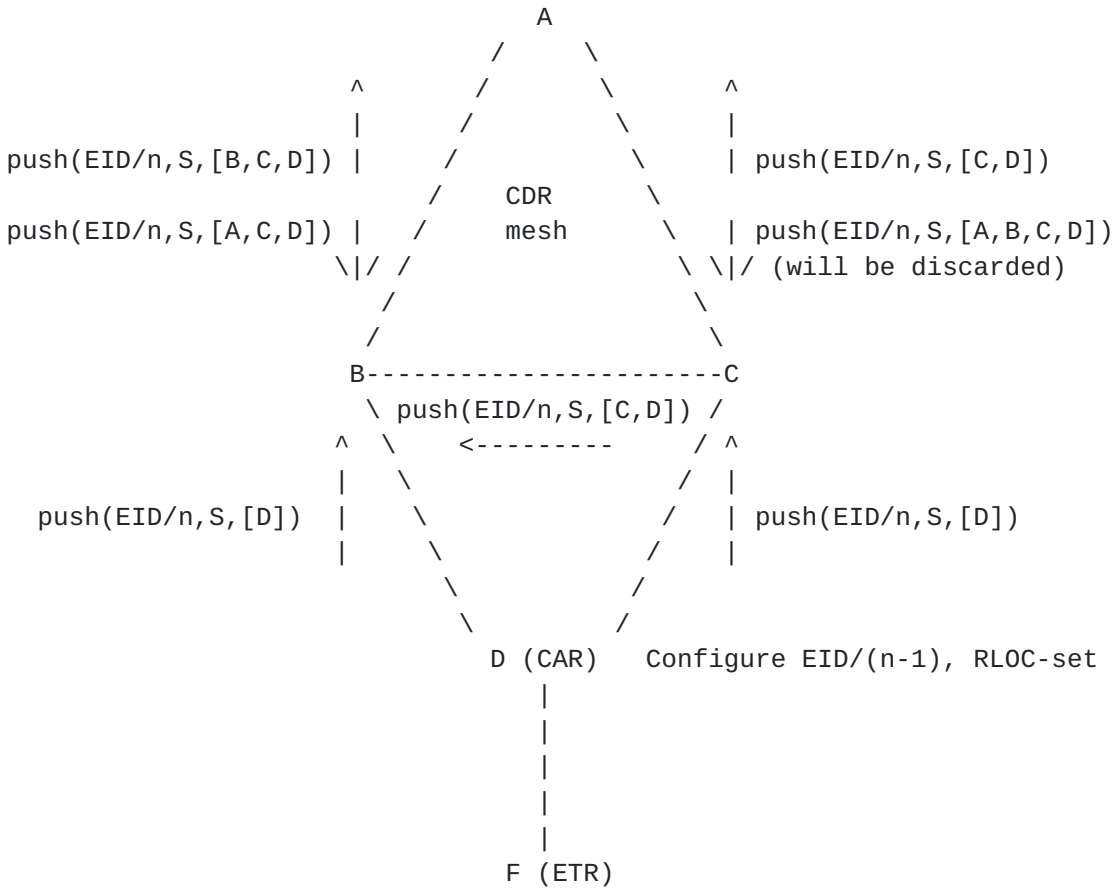

Figure 3: Sequence Number Processing

5.3.3. A CAR Becomes Unreachable

If the TCP connection between a CAR its peer CDR drops, a timer associated with the EID-prefix received from the CAR in the Push-Add message is started. The timer, called the CAR-CDR-TCP-TIMER, is set to a default value of 60 minutes.

If the TCP connection comes back up before the timer expires, the timer is stopped and no further action is taken.

If the timer expires, the CDR builds a Push-Delete message for each EID-prefix it received from the aCAR, and sends the Push-Delete to its siblings. The Push-Delete message contains the EID-prefix to be removed, a sequence number, and PV containing only the CDR's address.

If the CDR also has a parent peering, it checks to see if any of the EID-prefixes it received from a child peering were the last more specific prefix in an aggregate it previously pushed to a parent CDR. If not, no further action is taken. If so, it sends a Push-Delete for the aggregate to its parent(s). In either case, the CDR deletes the entries received from the failed CAR from its EID-prefix table.

5.3.4. A CDR Becomes Unreachable

There are three cases to consider here: A sibling CDR peering goes down, a parent peering goes down, and an child peering goes down. Each is considered below.

5.3.4.1. A Sibling CDR Becomes Unreachable

When the TCP connection drops between a CDR and a sibling CDR, a timer associated with the EID-prefixes received from the sibling CDR in the Push-Add message is started. This timer, called CDR-SIBLING-TCP-TIMER, defaults to TBD.

If the TCP connection comes back up before the timer expires, the timer is stopped and no further action is taken.

If the timer expires, the CDR builds a Push-Delete message for each EID-prefix it received from the CDR, and sends the Push-Delete to its siblings. The Push-Delete message contains the EID-prefix to be removed, a sequence number, and PV containing only the CDR's address.

If the CDR also has a parent peering, it checks to see if any of the EID-prefixes it received from the failed CDR were the last more specific prefix in an aggregate it previously pushed to a parent CDR. If not, no further action is taken. If so, it sends a Push-Delete for the aggregate to its parent(s). In either case, the CDR deletes

the entries from its EID-prefix table.

5.3.4.2. A Parent CDR Becomes Unreachable

When the TCP connection drops between a CDR and a parent CDR, the child starts a timer (the CDR-CDR-TCP-TIMER) associated with the parent CDR.

If the TCP connection comes back up before the timer expires, the timer is stopped and no further action is taken.

If the timer expires, the CDR deletes the EID-prefix entry, and builds a Push-Delete message for the default EID prefix and sends it to its siblings. The Push-Delete message contains the EID-prefix 0.0.0.0/0 or 0::0/0, a sequence number, and PV containing only the CDR's address.

5.3.4.3. A Child CDR Becomes Unreachable

Since nothing is ever "pushed down", no action needs to be taken when a child CDR becomes unreachable. See Section 5.3.4.2 for the actions a child CDR takes when a parent becomes unreachable.

6. LISP-CONS Message Types

LISP messages are sent over either UDP or TCP sockets using well-known IANA-assigned port number 4342.

In all message formats, IPv4 or IPv6 addresses can be mixed or match. So a payload of IPv6 addresses can be sent over a TCP connection (or be UDP encapsulated) that runs over IPv4 and vice-versa. You can also mix EID-to-RLOC mappings. That is, an IPv6 EID-prefix can have a set of IPv4 or IPv6 Locator addresses associated with it and vice-versa. Originator addresses and Path Vector lists can also be mixed as well.

A TCP connection is established by two LISP-CONS peers by having the higher IP address side of the connection do a passive-open and the lower IP address side to an active open. This is done to avoid 2 connections from call colliding. This is similar to the procedures in [RFC3618].

See [LISP] for packet type value definitions and formats.

7. Operational Considerations

TBD: However, mention that there will be less policy than in BGP. That is, information cannot be altered, like a CAR cannot add or remove locators, path-vectors can't be made to look longer, etc....

Future revisions of this document will have a more through description of deployment scenarios, once we get some implementation and pilot deployment experience.

8. LISP-CONS and Locator Reachability

It is important to note that LISP-CONS is designed to as a mapping database that defines EID-to-RLOC mappings, where the RLOCs are IP addresses of ETRs and does not indicate if the ETRs, or the path to the ETRs are up.

In general, LISP determine reachability through either ICMP No-Map messages or LISP data-plane Locator Reach bits that are transmitted in LISP Data messages [LISP].

The design principle underlying LISP-CONS is to keep the mapping database service scalable. As such, the design discourages high frequency changes in mappings.

9. LISP-CONS and Mobility

The mapping database does not convey Foreign Agent locator addresses. This can be achieved in the data plane but will be documented in another Internet Draft.

10. Open Issues

- Do we need a Close Message? (dual of open). Otherwise EID-prefixes may not get removed until a timeout.

- No mapping exists in the ITR: You have a configuration option to either 1) drop the packet, or 2) do LISP 1.5 where the packet is routed on another topology. The other option is to allow the ITR get a push of 0.0.0.0/0 or 0::0/0 from its peering CARs (or have it configured in the ITR).

- Security Section: We need to finish the evaluation of vulnerabilities. Map vulnerabilities against security mechanisms. At first blush, the real outstanding question remaining (as you

note in your notes above) is transitive message security (ala dns-sec).

- Security Model: Is the implied transitive trust sufficient?

- From http://ana-3.lcs.mit.edu/~jnc/tech/lisp/optimizations.txt:

  Caching of bindings in the CDR hierarchy: This is such a win, it gives you a system which is almost as fast as a 'push' system, but without the overhead of giving updates to people who don't need them

  'Piggybacking' of client bindings when a request is made: This will greatly increase the speed of responses for everyone, big and small; it is a considerably more complex optimization than any other, but the payoff is so significant I think it's probably worth it

  Direct reply to queries via UDP: This optimizes response time to cache misses on qCARs; not a big gain in performance, but it's very simple to do, so worth it overall

  Push of 'delegation' info (actually, advertisments): This will minimize the path length for requests traversing the server pseudo-hierarchy; it needs a good heuristic algorithm to limit the distance upwards, which I haven't seen yet (but feel confident we can come up with)

  Direct notification of outdated bindings: This is needed to make caching of bindings work

11. Acknowledgments

Many of the ideas described in this document developed during detailed discussions with Eliot Lear, Mark Handley, and Dave Oran. Robin Whittle also made several insightful comments on earlier versions of this document.

12. Security Considerations

LISP-CONS is a straightforward protocol to secure. Its combination of simplicity, explicit peering, and explicit configuration provides for a well understood set of relationships between elements. Its security mechanisms are comprised of existing technologies in wide operational use today.

As a hybrid push-pull protocol, LISP-CONS shares some of security characteristics of pull (DNS) and push (BGP) protocols. Securing LISP-CONS is much simpler than either of those examples however. Compared to DNS, the fact that messages traverse a explicit hierarchy of TCP connections, and the message make-up itself makes LISP-CONS less susceptible to denial of service and amplification attacks. Compared to BGP, LISP-CONS CDRs are not topologically bound, allowing them to be put in locations away from the vulnerable AS border (unlike eBGP speakers).

12.1. Apparent LISP-CONS Vunerabilities

This section briefly lists of the apparent vulnerabilities of LISP-CONS.

Mapping Integrity: Can you insert bogus mappings to black-hole (create a DoS) or intercept LISP data-plane packets?

CAR Availability: Can you DoS the aCAR(s) holding the mappings for a particular ETR? Without access to its 1-2 available CAR(s) an ITR has no ability to connect to the rest of the Internet.

ITR Mapping/Resources: Can you force an ITR to drop legitimate mapping requests by flooding it with random destinations that it will have to query for? Seems like a problem with any pull based system (DNS has this problem). Is this an ITR implementation issue, or is there a way we can assist ITR implementers here in the LISP-CONS spec?

Path Vector Exploits for Reconnaissance: Can you learn about the LISP topology by sending legitimate mapping requests messages and then observing the path-vector information. Is this information useful in attacking or subverting peer relationships? Not data plane but control plane service - this vulnerability seems unique to LISP-CONS. ITRs cannot do this, since they don't have access to the PVs (the PVs aren't sent along to the ITRs). Note that LISP has a similar data-plane reconnaissance issue.

Scaling of CAR/CDR Resources: Can you flood the system with requests or replies due to the limited capacity of the control plane? TCP prevents anycasting to add capacity, and one of the issues has to be how do we scale if we need to?

12.2. Survey of LISP-CONS Security Mechanisms

Use of Device Loopbacks: From levels 0 to 1 (or n) in the topology, these loopbacks should come from known infrastructure subnets (as do say BGP peers) that should allow for some isolation via Access Control Lists (ACLs) and anti-spoofing mechanisms.

Explicit Peering: The devices themselves can both prioritize incoming packets as well as potentially do key checks in hardware to protect the control plane.

Use of TCP to Connect Loopbacks: This makes it difficult for third parties to inject packets.

Use of HMAC Protected TCP Connections: HMAC is used to verify message integrity and authenticity, making it nearly impossible for third party devices to either insert or modify messages.

Message Sequence Numbers and Nonce Values in Messages: This allows for devices to verify that the mapping-reply packet was in response to the mapping-request that they sent.

Path Vectors: Path Vectors prevent arbitrary messages from traversing the topology, and raise the bar for spoofing/invalid Path-Delete messages.

13. IANA Considerations

This document creates no new requirements on IANA namespaces [RFC2434].

14. References

14.1. Normative References

[RFC1498] Saltzer, J., "On the Naming and Binding of Network Destinations", RFC 1498, August 1993.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC3618] Fenner, B. and D. Meyer, "Multicast Source Discovery Protocol (MSDP)", RFC 3618, October 2003.

[RFC4632] Fuller, V. and T. Li, "Classless Inter-domain Routing (CIDR): The Internet Address Assignment and Aggregation Plan", BCP 122, RFC 4632, August 2006.

[LISP] Farinacci, D., Fuller, V., Oran, D., and D. Meyer, "Locator/ID Separation Protocol (LISP)", draft-farinacci-lisp-06 (work in progress), Apr 2008.

14.2. Informative References

[RFC2434] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 2434, October 1998.

[RFC4984] Meyer, D., Zhang, L., and K. Fall, "Report from the IAB Workshop on Routing and Addressing", RFC 4984, September 2007.

[CHIAPPA] Chiappa, J., "Endpoints and Endpoint names: A Proposed Enhancement to the Internet Architecture", Internet Draft, http://ana.lcs.mit.edu/~jnc/tech/endpoints.txt, 1999.

Authors' Addresses

Scott Brim

Email: sbrim@cisco.com

Noel Chiappa

Email: jnc@mercury.lcs.mit.edu

Dino Farinacci

Email: dino@cisco.com

Vince Fuller

Email: vaf@cisco.com

Darrel Lewis

Email: darlewis@cisco.com

David Meyer

Email: dmm@cisco.com

Full Copyright Statement

Intellectual Property