Abstract

Many Massively Scaled Data Centers (MSDCs) have converged on simplified layer 3 routing. Furthermore, requirements for operational simplicity have lead many of these MSDCs to converge on BGP as their single routing protocol for both their fabric routing and their Data Center Interconnect (DCI) routing. This document describes a solution which leverages BGP Link-State distribution and the Shortest Path First (SPF) algorithm similar to Internal Gateway Protocols (IGPs) such as OSPF.

Status of This Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

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."

This Internet-Draft will expire on June 23, 2019.

Copyright Notice

Copyright (c) 2018 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.

This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

Table of Contents

1. Introduction ................................................. 3
   1.1. BGP Shortest Path First (SPF) Motivation ............... 4
   1.2. Requirements Language ................................. 5
2. BGP Peering Models ........................................ 5
   2.1. BGP Single-Hop Peering on Network Node Connections ... 5
   2.2. BGP Peering Between Directly Connected Network Nodes .. 6
   2.3. BGP Peering in Route-Reflector or Controller Topology .. 6
3. BGP-LS Shortest Path Routing (SPF) SAFI .................. 6
4. Extensions to BGP-LS ...................................... 7
   4.1. Node NLRI Usage and Modifications ..................... 7
   4.2. Link NLRI Usage ...................................... 8
       4.2.1. BGP-LS Link NLRI Attribute Prefix-Length TLVs ..... 9
       4.2.2. BGP-LS Link NLRI Attribute BGP SPF Status TLV ..... 9
       4.2.3. BGP-LS Prefix NLRI Attribute SPF Status TLV ..... 10
   4.3. Prefix NLRI Usage .................................... 10
   4.4. BGP-LS Attribute Sequence-Number TLV .................. 10
5. Decision Process with SPF Algorithm ....................... 11
   5.1. Phase-1 BGP NLRI Selection ........................... 12
   5.2. Dual Stack Support ................................. 13
   5.3. SPF Calculation based on BGP-LS NLRI ................ 13
   5.4. NEXT_HOP Manipulation ............................... 16
   5.5. IPv4/IPv6 Unicast Address Family Interaction .......... 16
   5.6. NLRI Advertisement and Convergence ................... 17
       5.6.1. Link/Prefix Failure Convergence ................. 17
1. Introduction

Many Massively Scaled Data Centers (MSDCs) have converged on simplified layer 3 routing. Furthermore, requirements for operational simplicity have lead many of these MSDCs to converge on BGP [RFC4271] as their single routing protocol for both their fabric routing and their Data Center Interconnect (DCI) routing. Requirements and procedures for using BGP are described in [RFC7938]. This document describes an alternative solution which leverages BGP-LS [RFC7752] and the Shortest Path First algorithm similar to Internal Gateway Protocols (IGPs) such as OSPF [RFC2328].

[RFC4271] defines the Decision Process that is used to select routes for subsequent advertisement by applying the policies in the local Policy Information Base (PIB) to the routes stored in its Adj-RIBs-In. The output of the Decision Process is the set of routes that are announced by a BGP speaker to its peers. These selected routes are stored by a BGP speaker in the speaker’s Adj-RIBs-Out according to policy.

[RFC7752] describes a mechanism by which link-state and TE information can be collected from networks and shared with external components using BGP. This is achieved by defining NLRI advertised within the BGP-LS/BGP-LS-SPF AFI/SAFI. The BGP-LS extensions defined in [RFC7752] makes use of the Decision Process defined in [RFC4271].

This document augments [RFC7752] by replacing its use of the existing Decision Process. Rather than reusing the BGP-LS SAFI, the BGP-LS-SPF SAFI is introduced to insure backward compatibility. The Phase 1 and 2 decision functions of the Decision Process are replaced with the Shortest Path First (SPF) algorithm also known as the Dijkstra algorithm. The Phase 3 decision function is also simplified since it is no longer dependent on the previous phases. This solution avails the benefits of both BGP and SPF-based IGPs. These include TCP based
flow-control, no periodic link-state refresh, and completely incremental NLRI advertisement. These advantages can reduce the overhead in MSDCs where there is a high degree of Equal Cost Multi-Path (ECMPs) and the topology is very stable. Additionally, using a SPF-based computation can support fast convergence and the computation of Loop-Free Alternatives (LFAs) [RFC5286] in the event of link failures. Furthermore, a BGP based solution lends itself to multiple peering models including those incorporating route-reflectors [RFC4456] or controllers.

Support for Multiple Topology Routing (MTR) as described in [RFC4915] is an area for further study dependent on deployment requirements.

1.1. BGP Shortest Path First (SPF) Motivation

Given that [RFC7938] already describes how BGP could be used as the sole routing protocol in an MSDC, one might question the motivation for defining an alternate BGP deployment model when a mature solution exists. For both alternatives, BGP offers the operational benefits of a single routing protocol. However, BGP SPF offers some unique advantages above and beyond standard BGP distance-vector routing.

A primary advantage is that all BGP speakers in the BGP SPF routing domain will have a complete view of the topology. This will allow support for ECMP, IP fast-reroute (e.g., Loop-Free Alternatives), Shared Risk Link Groups (SRLGs), and other routing enhancements without advertisement of addition BGP paths or other extensions. In short, the advantages of an IGP such as OSPF [RFC2328] are availed in BGP.

With the simplified BGP decision process as defined in Section 5.1, NLRI changes can be disseminated throughout the BGP routing domain much more rapidly (equivalent to IGPs with the proper implementation).

Another primary advantage is a potential reduction in NLRI advertisement. With standard BGP distance-vector routing, a single link failure may impact 100s or 1000s prefixes and result in the withdrawal or re-advertisement of the attendant NLRI. With BGP SPF, only the BGP speakers corresponding to the link NLRI need withdraw the corresponding BGP-LS Link NLRI. This advantage will contribute to both faster convergence and better scaling.

With controller and route-reflector peering models, BGP SPF advertisement and distributed computation require a minimal number of sessions and copies of the NLRI since only the latest version of the NLRI from the originator is required. Given that verification of the adjacencies is done outside of BGP (see Section 2), each BGP speaker
will only need as many sessions and copies of the NLRI as required for redundancy (e.g., one for the SPF computation and another for backup). Functions such as Optimized Route Reflection (ORR) are supported without extension by virtue of the primary advantages. Additionally, a controller could inject topology that is learned outside the BGP routing domain.

Given that controllers are already consuming BGP-LS NLRI [RFC7752], reusing for the BGP-LS SPF leverages the existing controller implementations.

Another potential advantage of BGP SPF is that both IPv6 and IPv4 can be supported in the same address family using the same topology. Although not described in this version of the document, multi-topology extensions can be used to support separate IPv4, IPv6, unicast, and multicast topologies while sharing the same NLRI.

Finally, the BGP SPF topology can be used as an underlay for other BGP address families (using the existing model) and realize all the above advantages. A simplified peering model using IPv6 link-local addresses as next-hops can be deployed similar to [RFC5549].

1.2. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

2. BGP Peering Models

Depending on the requirements, scaling, and capabilities of the BGP speakers, various peering models are supported. The only requirement is that all BGP speakers in the BGP SPF routing domain receive link-state NLRI on a timely basis, run an SPF calculation, and update their data plane appropriately. The content of the Link NLRI is described in Section 4.2.

2.1. BGP Single-Hop Peering on Network Node Connections

The simplest peering model is the one described in section 5.2.1 of [RFC7938]. In this model, EBGP single-hop sessions are established over direct point-to-point links interconnecting the SPF domain nodes. For the purposes of BGP SPF, Link NLRI is only advertised if a single-hop BGP session has been established and the Link-State/SPF address family capability has been exchanged [RFC4790] on the corresponding session. If the session goes down, the corresponding
Link NLRI will be withdrawn. Topologically, this would be equivalent to the peering model in [RFC7938] where there is a BGP session on every link in the data center switch fabric.

2.2. BGP Peering Between Directly Connected Network Nodes

In this model, BGP speakers peer with all directly connected network nodes but the sessions may be multi-hop and the direct connection discovery and liveliness detection for those connections are independent of the BGP protocol. How this is accomplished is outside the scope of this document. Consequently, there will be a single session even if there are multiple direct connections between BGP speakers. For the purposes of BGP SPF, Link NLRI is advertised as long as a BGP session has been established, the Link-State/SPF address family capability has been exchanged [RFC4790] and the corresponding link is considered is up and considered operational. This is much like the previous peering model only peering is on a single loopback address and the switch fabric links can be unnumbered. However, there will be the same number of sessions as with the previous peering model unless there are parallel links between switches in the fabric.

2.3. BGP Peering in Route-Reflector or Controller Topology

In this model, BGP speakers peer solely with one or more Route Reflectors [RFC4456] or controllers. As in the previous model, direct connection discovery and liveliness detection for those connections are done outside the BGP protocol. More specifically, the Liveliness detection is done using BFD protocol described in [RFC5880]. For the purposes of BGP SPF, Link NLRI is advertised as long as the corresponding link is up and considered operational. This peering model, known as sparse peering, allows for many fewer BGP sessions and, consequently, instances of the same NLRI received from multiple peers. It is discussed in greater detail in [I-D.ietf-lsvr-applicability].

3. BGP-LS Shortest Path Routing (SPF) SAFI

In order to replace the Phase 1 and 2 decision functions of the existing Decision Process with an SPF-based Decision Process and streamline the Phase 3 decision functions in a backward compatible manner, this draft introduces the BGP-LS-SFP SAFI for BGP-LS SPF operation. The BGP-LS-SPF (AF 16388 / SAFI TBD1) [RFC4790] is allocated by IANA as specified in the Section 6. A BGP speaker using the BGP-LS SPF extensions described herein MUST exchange the AFI/SAFI using Multiprotocol Extensions Capability Code [RFC4760] with other BGP speakers in the SPF routing domain.
4. Extensions to BGP-LS

[RFC7752] describes a mechanism by which link-state and TE information can be collected from networks and shared with external components using BGP protocol. It describes both the definition of BGP-LS NLRI that describes links, nodes, and prefixes comprising IGP link-state information and the definition of a BGP path attribute (BGP-LS attribute) that carries link, node, and prefix properties and attributes, such as the link and prefix metric or auxiliary Router-IDs of nodes, etc.

The BGP protocol will be used in the Protocol-ID field specified in table 1 of [I-D.ietf-idr-bgpls-segment-routing-epe]. The local and remote node descriptors for all NLRI will be the BGP Router-ID (TLV 516) and either the AS Number (TLV 512) [RFC7752] or the BGP Confederation Member (TLV 517) [RFC8402]. However, if the BGP Router-ID is known to be unique within the BGP Routing domain, it can be used as the sole descriptor.

4.1. Node NLRI Usage and Modifications

The SPF capability is a new Node Attribute TLV that will be added to those defined in table 7 of [RFC7752]. The new attribute TLV will only be applicable when BGP is specified in the Node NLRI Protocol ID field. The TBD TLV type will be defined by IANA. The new Node Attribute TLV will contain a single-octet SPF algorithm as defined in [RFC8402].
The SPF Algorithm may take the following values:

0 - Normal Shortest Path First (SPF) algorithm based on link metric. This is the standard shortest path algorithm as computed by the IGP protocol. Consistent with the deployed practice for link-state protocols, Algorithm 0 permits any node to overwrite the SPF path with a different path based on its local policy.

1 - Strict Shortest Path First (SPF) algorithm based on link metric. The algorithm is identical to Algorithm 0 but Algorithm 1 requires that all nodes along the path will honor the SPF routing decision. Local policy at the node claiming support for Algorithm 1 MUST NOT alter the SPF paths computed by Algorithm 1.

Note that usage of Strict Shortest Path First (SPF) algorithm is defined in the IGP algorithm registry but usage is restricted to [I-D.ietf-idr-bgpls-segment-routing-epe]. Hence, its usage for BGP-LS SPF is out of scope.

When computing the SPF for a given BGP routing domain, only BGP nodes advertising the SPF capability attribute will be included the Shortest Path Tree (SPT).

4.2. Link NLRI Usage

The criteria for advertisement of Link NLRI are discussed in Section 2.

Link NLRI is advertised with local and remote node descriptors as described above and unique link identifiers dependent on the addressing. For IPv4 links, the links local IPv4 (TLV 259) and remote IPv4 (TLV 260) addresses will be used. For IPv6 links, the local IPv6 (TLV 261) and remote IPv6 (TLV 262) addresses will be used. For unnumbered links, the link local/remote identifiers (TLV 258) will be used. For links supporting having both IPv4 and IPv6 addresses, both sets of descriptors may be included in the same Link NLRI. The link identifiers are described in table 5 of [RFC7752].

The link IGP metric attribute TLV (TLV 1095) as well as any others required for non-SPF purposes SHOULD be advertised. Algorithms such
as setting the metric inversely to the link speed as done in the OSPF MIB [RFC4750] MAY be supported. However, this is beyond the scope of this document.

4.2.1. BGP-LS Link NLRI Attribute Prefix-Length TLVs

Two BGP-LS Attribute TLVs to BGP-LS Link NLRI are defined to advertise the prefix length associated with the IPv4 and IPv6 link prefixes. The prefix length is used for the optional installation of prefixes corresponding to Link NLRI as defined in Section 5.3.

```
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      TBD IPv4 or IPv6 Type    |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Prefix-Length |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Prefix-length - A one-octet length restricted to 1-32 for IPv4 Link NLIR endpoint prefixes and 1-128 for IPv6 Link NLRI endpoint prefixes.

4.2.2. BGP-LS Link NLRI Attribute BGP SPF Status TLV

A BGP-LS Attribute TLV to BGP-LS Link NLRI is defined to indicate the status of the link with respect to the BGP SPF calculation. This will be used to expedite convergence for link failures as discussed in Section 5.6.1. If the BGP SPF Status TLV is not included with the Link NLRI, the link is considered up and available.

```
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| TBD Type | Length |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| BGP SPF Status |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

BGP Status Values: 0 - Reserved
1 - Link Unreachable with respect to BGP SPF
2-254 -Undefined
255 - Reserved
4.2.3. BGP-LS Prefix NLRI Attribute SPF Status TLV

A BGP-LS Attribute TLV to BGP-LS Prefix NLRI is defined to indicate the status of the prefix with respect to the BGP SPF calculation. This will be used to expedite convergence for prefix unreachability as discussed in Section 5.6.1. If the SPF Status TLV is not included with the Prefix NLRI, the prefix is considered reachable.

```
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   TBD Type    |                       Length                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| BGP SPF Status|                                     |
+-+-+-+-+-+-+-+-+-

BGP Status Values: 0 - Reserved
1 - Prefix down with respect to SPF
2-254 - Undefined
255 - Reserved
```

4.3. Prefix NLRI Usage

Prefix NLRI is advertised with a local node descriptor as described above and the prefix and length used as the descriptors (TLV 265) as described in [RFC7752]. The prefix metric attribute TLV (TLV 1155) as well as any others required for non-SPF purposes SHOULD be advertised. For loopback prefixes, the metric should be 0. For non-loopback prefixes, the setting of the metric is a local matter and beyond the scope of this document.

4.4. BGP-LS Attribute Sequence-Number TLV

A new BGP-LS Attribute TLV to BGP-LS NLRI types is defined to assure the most recent version of a given NLRI is used in the SPF computation. The TBD TLV type will be defined by IANA. The new BGP-LS Attribute TLV will contain an 8-octet sequence number. The usage of the Sequence Number TLV is described in Section 5.1.
Sequence Number

The 64-bit strictly increasing sequence number is incremented for every version of BGP-LS NLRI originated. BGP speakers implementing this specification MUST use available mechanisms to preserve the sequence number’s strictly increasing property for the deployed life of the BGP speaker (including cold restarts). One mechanism for accomplishing this would be to use the high-order 32 bits of the sequence number as a wrap/boot count that is incremented anytime the BGP router loses its sequence number state or the low-order 32 bits wrap.

When incrementing the sequence number for each self-originated NLRI, the sequence number should be treated as an unsigned 64-bit value. If the lower-order 32-bit value wraps, the higher-order 32-bit value should be incremented and saved in non-volatile storage. If by some chance the BGP Speaker is deployed long enough that there is a possibility that the 64-bit sequence number may wrap or a BGP Speaker completely loses its sequence number state (e.g., the BGP speaker hardware is replaced or experiences a cold-start), the phase 1 decision function (see Section 5.1) rules will insure convergence, albeit, not immediately.

5. Decision Process with SPF Algorithm

The Decision Process described in [RFC4271] takes place in three distinct phases. The Phase 1 decision function of the Decision Process is responsible for calculating the degree of preference for each route received from a BGP speaker’s peer. The Phase 2 decision function is invoked on completion of the Phase 1 decision function and is responsible for choosing the best route out of all those available for each distinct destination, and for installing each chosen route into the Loc-RIB. The combination of the Phase 1 and 2 decision functions is characterized as a Path Vector algorithm.

The SPF based Decision process replaces the BGP best-path Decision process described in [RFC4271]. This process starts with selecting only those Node NLRI whose SPF capability TLV matches with the local
BGP speaker’s SPF capability TLV value. Since Link-State NLRI always contains the local descriptor [RFC7752], it will only be originated by a single BGP speaker in the BGP routing domain. These selected Node NLRI and their Link/Prefix NLRI are used to build a directed graph during the SPF computation. The best paths for BGP prefixes are installed as a result of the SPF process.

When BGP-LS-SPF NLRI is received, all that is required is to determine whether it is the best-path by examining the Node-ID and sequence number as described in Section 5.1. If the received best-path NLRI had changed, it will be advertised to other BGP-LS-SPF peers. If the attributes have changed (other than the sequence number), a BGP SPF calculation will be scheduled. However, a changed NLRI MAY be advertised to other peers almost immediately and propagation of changes can approach IGP convergence times. To accomplish this, the MinRouteAdvertisementIntervalTimer and MinASOriginationIntervalTimer [RFC4271] are not applicable to the BGP-LS-SPF SAFI. Rather, SPF calculations SHOULD be triggered and dampened consistent with the SPF backoff algorithm specified in [RFC8405].

The Phase 3 decision function of the Decision Process [RFC4271] is also simplified since under normal SPF operation, a BGP speaker would advertise the NLRI selected for the SPF to all BGP peers with the BGP-LS/BGP-LS-SPF AFI/SAFI. Application of policy would not be prevented however its usage to best-path process would be limited as the SPF relies solely on link metrics.

5.1. Phase-1 BGP NLRI Selection

The rules for NLRI selection are greatly simplified from [RFC4271].

1. If the NLRI is received from the BGP speaker originating the NLRI (as determined by the comparing BGP Router ID in the NLRI Node identifiers with the BGP speaker Router ID), then it is preferred over the same NLRI from non-originators. This rule will assure that stale NLRI is updated even if a BGP-LS router loses its sequence number state due to a cold-start.

2. If the Sequence-Number TLV is present in the BGP-LS Attribute, then the NLRI with the most recent, i.e., highest sequence number is selected. BGP-LS NLRI with a Sequence-Number TLV will be considered more recent than NLRI without a BGP-LS Attribute or a BGP-LS Attribute that doesn’t include the Sequence-Number TLV.

3. The final tie-breaker is the NLRI from the BGP Speaker with the numerically largest BGP Router ID.
When a BGP speaker completely loses its sequence number state, i.e., due to a cold start, or in the unlikely possibility that that sequence number wraps, the BGP routing domain will still converge. This is due to the fact that BGP speakers adjacent to the router will always accept self-originated NLRI from the associated speaker as more recent (rule # 1). When BGP speaker reestablishes a connection with its peers, any existing session will be taken down and stale NLRI will be replaced by the new NLRI and stale NLRI will be discarded independent of whether or not BGP graceful restart is deployed, [RFC4724]. The adjacent BGP speaker will update their NLRI advertisements in turn until the BGP routing domain has converged.

The modified SPF Decision Process performs an SPF calculation rooted at the BGP speaker using the metrics from Link and Prefix NLRI Attribute TLVs [RFC7752]. As a result, any attributes that would influence the Decision process defined in [RFC4271] like ORIGIN, MULTI_EXIT_DISC, and LOCAL_PREF attributes are ignored by the SPF algorithm. Furthermore, the NEXT_HOP attribute value is preserved but otherwise ignored during the SPF or best-path.

5.2. Dual Stack Support

The SPF-based decision process operates on Node, Link, and Prefix NLRI that support both IPv4 and IPv6 addresses. Whether to run a single SPF instance or multiple SPF instances for separate AFs is a matter of a local implementation. Normally, IPv4 next-hops are calculated for IPv4 prefixes and IPv6 next-hops are calculated for IPv6 prefixes. However, an interesting use-case is deployment of [RFC5549] where IPv6 next-hops are calculated for both IPv4 and IPv6 prefixes. As stated in Section 1, support for Multiple Topology Routing (MTR) is an area for future study.

5.3. SPF Calculation based on BGP-LS NLRI

This section details the BGP-LS SPF local routing information base (RIB) calculation. The router will use BGP-LS Node, Link, and Prefix NLRI to populate the local RIB using the following algorithm. This calculation yields the set of intra-area routes associated with the BGP-LS domain. A router calculates the shortest-path tree using itself as the root. Variations and optimizations of the algorithm are valid as long as it yields the same set of routes. The algorithm below supports Equal Cost Multi-Path (ECMP) routes. Weighted Unequal Cost Multi-Path are out of scope. The organization of this section owes heavily to section 16 of [RFC2328].

The following abstract data structures are defined in order to specify the algorithm.
Local Route Information Base (RIB) - This is abstract contains reachability information (i.e., next hops) for all prefixes (both IPv4 and IPv6) as well as the Node NLRI reachability. Implementations may choose to implement this as separate RIBs for each address family and/or Node NLRI.

Link State NLRI Database (LSNDB) - Database of BGP-LS NLRI that facilitates access to all Node, Link, and Prefix NLRI as well as all the Link and Prefix NLRI corresponding to a given Node NLRI. Other optimization, such as, resolving bi-directional connectivity associations between Link NLRI are possible but of scope of this document.

Candidate List - This is a list of candidate Node NLRI with the lowest cost Node NLRI at the front of the list. It is typically implemented as a heap but other concrete data structures have also been used.

The algorithm is comprised of the steps below:

1. The current local RIB is invalidated. The local RIB is built again from scratch. The existing routing entries are preserved for comparison to determine changes that need to be installed in the global RIB.

2. The computing router’s Node NLRI is installed in the local RIB with a cost of 0 and as as the sole entry in the candidate list.

3. The Node NLRI with the lowest cost is removed from the candidate list for processing. The Node corresponding to this NLRI will be referred to as the Current Node. If the candidate list is empty, the SPF calculation has completed and the algorithm proceeds to step 6.

4. All the Prefix NLRI with the same Node Identifiers as the Current Node will be considered for installation. The cost for each prefix is the metric advertised in the Prefix NLRI added to the cost to reach the Current Node.

   * If the BGP-LS Prefix attribute includes an BGP-SPF Status TLV indicating the prefix is unreachable, the BGP-LS Prefix NLRI is considered unreachable and the next BGP-LS Prefix NLRI is examined.

   * If the prefix is in the local RIB and the cost is greater than the Current route’s metric, the Prefix NLRI does not contribute to the route and is ignored.
* If the prefix is in the local RIB and the cost is less than
  the current route’s metric, the Prefix is installed with the
  Current Node’s next-hops replacing the local RIB route’s next-
  hops and the metric being updated.

* If the prefix is in the local RIB and the cost is same as the
  current route’s metric, the Prefix is installed with the
  Current Node’s next-hops being merged with local RIB route’s
  next-hops.

5. All the Link NLRI with the same Node Identifiers as the Current
   Node will be considered for installation. Each link will be
   examined and will be referred to in the following text as the
   Current Link. The cost of the Current Link is the advertised
   metric in the Link NLRI added to the cost to reach the Current
   Node.

* Optionally, the prefix(es) associated with the Current Link
  are installed into the local RIB using the same rules as were
  used for Prefix NLRI in the previous steps.

* The Current Link’s endpoint Node NLRI is accessed (i.e., the
  Node NLRI with the same Node identifiers as the Link
  endpoint). If it exists, it will be referred to as the
  Endpoint Node NLRI and the algorithm will proceed as follows:

  + If the BGP-LS Link NLRI includes an BGP-SPF Status TLV
    indicating the link is down, the BGP-LS Link NLRI is
    considered down and the next BGP-LS Link NLRI is examined.

  + All the Link NLRI corresponding the Endpoint Node NLRI will
    be searched for a back-link NLRI pointing to the current
    node. Both the Node identifiers and the Link endpoint
    identifiers in the Endpoint Node’s Link NLRI must match for
    a match. If there is no corresponding Link NLRI
    corresponding to the Endpoint Node NLRI, the Endpoint Node
    NLRI fails the bi-directional connectivity test and is not
    processed further.

  + If the Endpoint Node NLRI is not on the candidate list, it
    is inserted based on the link cost and BGP Identifier (the
    latter being used as a tie-breaker).

  + If the Endpoint Node NLRI is already on the candidate list
    with a lower cost, it need not be inserted again.
+ If the Endpoint Node NLRI is already on the candidate list with a higher cost, it must be removed and reinserted with a lower cost.

* Return to step 3 to process the next lowest cost Node NLRI on the candidate list.

6. The local RIB is examined and changes (adds, deletes, modifications) are installed into the global RIB.

5.4. NEXT_HOP Manipulation

A BGP speaker that supports SPF extensions MAY interact with peers that don't support SPF extensions. If the BGP-LS address family is advertised to a peer not supporting the SPF extensions described herein, then the BGP speaker MUST conform to the NEXT_HOP rules specified in [RFC4271] when announcing the Link-State address family routes to those peers.

All BGP peers that support SPF extensions would locally compute the Loc-RIB next-hops as a result of the SPF process. Consequently, the NEXT_HOP attribute is always ignored on receipt. However, BGP speakers SHOULD set the NEXT_HOP address according to the NEXT_HOP attribute rules specified in [RFC4271].

5.5. IPv4/IPv6 Unicast Address Family Interaction

While the BGP-LS SPF address family and the IPv4/IPv6 unicast address families install routes into the same device routing tables, they will operate independently much the same as OSPF and IS-IS would operate today (i.e., "Ships-in-the-Night" mode). There will be no implicit route redistribution between the BGP address families. However, implementation specific redistribution mechanisms SHOULD be made available with the restriction that redistribution of BGP-LS SPF routes into the IPv4 address family applies only to IPv4 routes and redistribution of BGP-LS SPF route into the IPv6 address family applies only to IPv6 routes.

Given the fact that SPF algorithms are based on the assumption that all routers in the routing domain calculate the precisely the same SPF tree and install the same set of routes, it is RECOMMENDED that BGP-LS SPF IPv4/IPv6 routes be given priority by default when installed into their respective RIBs. In common implementations the prioritization is governed by route preference or administrative distance with lower being more preferred.
5.6. NLRI Advertisement and Convergence

5.6.1. Link/Prefix Failure Convergence

A local failure will prevent a link from being used in the SPF calculation due to the IGP bi-directional connectivity requirement. Consequently, local link failures should always be given priority over updates (e.g., withdrawing all routes learned on a session) in order to ensure the highest priority propagation and optimal convergence.

An IGP such as OSPF [RFC2328] will stop using the link as soon as the Router-LSA for one side of the link is received. With normal BGP advertisement, the link would continue to be used until the last copy of the BGP-LS Link NLRI is withdrawn. In order to avoid this delay, the originator of the Link NLRI will advertise a more recent version of the BGP-LS Link NLRI including the BGP-SPF Status TLV Section 4.2.2 indicating the link is down with respect to BGP-SPF. After some configurable period of time, e.g., 2-3 seconds, the BGP-LS Link NLRI can be withdrawn with no consequence. If the link becomes available in that period, the originator of the BGP-LS LINK NLRI will simply advertise a more recent version of the BGP-LS Link NLRI without the BGP-SPF status TLV in the BGP-LS Link Attributes.

Similarly, when a prefix becomes unreachable, a more recent version of the BGP-LS Prefix NLRI will be advertised with the BGP-SPF status TLV Section 4.2.3 indicating the prefix is unreachable in the BGP-LS Prefix Attributes and the prefix will be considered unreachable with respect to BGP SPF. After some configurable period of time, e.g., 2-3 seconds, the BGP-LS Prefix NLRI can be withdrawn with no consequence. If the prefix becomes reachable in that period, the originator of the BGP-LS Prefix NLRI will simply advertise a more recent version of the BGP-LS Prefix NLRI without the BGP-SPF status TLV in the BGP-LS Prefix Attributes.

5.6.2. Node Failure Convergence

With BGP without graceful restart [RFC4724], all the NLRI advertised by node are implicitly withdrawn when a session failure is detected. If fast failure detection such as BFD is utilized and the node is on the fastest converging path, the most recent versions of BGP-LS NLRI may be withdrawn while these versions are in-flight on longer paths. This will result the older version of the NLRI being used until the new versions arrive and, potentially, unnecessary route flaps. Therefore, BGP-LS SPF NLRI SHOULD always be retained before being implicitly withdrawn for a brief configurable interval, e.g., 2-3 seconds. This will not delay convergence since the adjacent nodes will detect the link failure and advertise a more recent NLRI.
indicating the link is down with respect to BGP SPF Section 5.6.1 and the BGP-SPF calculation will failure the bi-directional connectivity check.

5.7. Error Handling

When a BGP speaker receives a BGP Update containing a malformed SPF Capability TLV in the Node NLRI BGP-LS Attribute [RFC7752], it MUST ignore the received TLV and the Node NLRI and not pass it to other BGP peers as specified in [RFC7606]. When discarding a Node NLRI with malformed TLV, a BGP speaker SHOULD log an error for further analysis.

6. IANA Considerations

This document defines an AFI/SAFI for BGP-LS SPF operation and requests IANA to assign the BGP-LS/BGP-LS-SPF (AFI 16388 / SAFI TBD1) as described in [RFC4750].

This document also defines four attribute TLVs for BGP LS NLRI. We request IANA to assign TLVs for the SPF capability, Sequence Number, IPv4 Link Prefix-Length, and IPv6 Link Prefix-Length from the "BGP-LS Node Descriptor, Link Descriptor, Prefix Descriptor, and Attribute TLVs" Registry.

7. Security Considerations

This extension to BGP does not change the underlying security issues inherent in the existing [RFC4271], [RFC4724], and [RFC7752].

8. Management Considerations

This section includes unique management considerations for the BGP-LS SPF address family.

8.1. Configuration

In addition to configuration of the BGP-LS SPF address family, implementations SHOULD support the configuration of the INITIAL_SPF_DELAY, SHORT_SPF_DELAY, LONG_SPF_DELAY, TIME_TO_LEARN, and HOLDDOWN_INTERVAL as documented in [RFC8405].

8.2. Operational Data

In order to troubleshoot SPF issues, implementations SHOULD support an SPF log including entries for previous SPF computations. Each SPF log entry would include the BGP-LS NLRI SPF triggering the SPF, SPF scheduled time, SPF start time, SPF end time, and SPF type if
different types of SPF are supported. Since the size of the log will
be finite, implementations SHOULD also maintain counters for the
total number of SPF computations of each type and the total number of
SPF triggering events. Additionally, to troubleshoot SPF scheduling
and backoff [RFC8405], the current SPF backoff state, remaining time-
to-learn, remaining holddown, last trigger event time, last SPF time,
and next SPF time should be available.

9. Acknowledgements

The authors would like to thank Sue Hares, Jorge Rabadan, Boris
Hassanov, Dan Frost, and Fred Baker for their review and comments.

The authors extend special thanks to Eric Rosen for fruitful
discussions on BGP-LS SPF convergence as compared to IGPs.

10. Contributors

In addition to the authors listed on the front page, the following
co-authors have contributed to the document.

Derek Yeung
Arrcus, Inc.
derek@arrcus.com

Gunter Van De Velde
Nokia
gunter.van_de_velde@nokia.com

Abhay Roy
Cisco Systems
akr@cisco.com

Venu Venugopal
Cisco Systems
venuv@cisco.com

11. References

11.1. Normative References

[I-D.ietf-idr-bgpls-segment-routing-epe]
Previdi, S., Filsfils, C., Patel, K., Ray, S., and J.
Dong, "BGP-LS extensions for Segment Routing BGP Egress
Peer Engineering", draft-ietf-idr-bgpls-segment-routing-
epe-15 (work in progress), March 2018.
11.2. Information References

[I-D.ietf-lsvr-applicability]
Authors’ Addresses

Keyur Patel
Arrcus, Inc.

Email: keyur@arrcus.com

Acee Lindem
Cisco Systems
301 Midenhall Way
Cary, NC  27513
USA

Email: acee@cisco.com

Shawn Zandi
Linkedin
222 2nd Street
San Francisco, CA  94105
USA

Email: szandi@linkedin.com

Wim Henderickx
Nokia
Antwerp
Belgium

Email: wim.henderickx@nokia.com
Abstract

Used in Massive Data Centers (MDCs), BGP-SPF and similar protocols need link neighbor discovery, link encapsulation data, and Layer 2 liveness. The Link State Over Ethernet protocol provides link discovery, exchanges supported encapsulations (IPv4, IPv6, ...), discovers encapsulation addresses (Layer 3 / MPLS identifiers) over raw Ethernet, and provides layer 2 liveness checking. The interface data are pushed directly to a BGP API (for LSVR), obviating the need for centralized topology distribution architectures. This protocol is intended to be more widely applicable to other upper layer routing protocols which need link discovery and characterisation.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

Status of This Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

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."

This Internet-Draft will expire on August 22, 2019.
Copyright Notice

Copyright (c) 2019 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 (https://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. Introduction .................................................. 3
2. Terminology .................................................... 4
3. Background ...................................................... 4
4. Top Level Overview .............................................. 5
5. Ethernet to Ethernet Protocols ................................. 6
   5.1. Inter-Link Ether Protocol Overview ..................... 7
6. Transport Layer ................................................ 8
7. The Checksum ................................................... 9
8. TLV PDUs ........................................................ 11
9. HELLO .......................................................... 11
10. OPEN .......................................................... 12
11. ACK ............................................................ 14
   11.1. Retransmission ........................................... 15
12. The Encapsulations ............................................. 15
   12.1. The Encapsulation PDU Skeleton .......................... 15
   12.2. Prim/Loop Flags .......................................... 17
   12.3. IPv4 Encapsulation ....................................... 17
   12.4. IPv6 Encapsulation ....................................... 17
   12.5. MPLS Label List .......................................... 18
   12.6. MPLS IPv4 Encapsulation ................................ 18
   12.7. MPLS IPv6 Encapsulation ................................ 19
13. KEEPALIVE - Layer 2 Liveness ................................. 19
14. VENDOR - Vendor Extensions .................................. 20
15. Layers 2.5 and 3 Liveness .................................. 20
16. The North/South Protocol .................................... 21
   16.1. Use BGP-LS as Much as Possible ....................... 21
   16.2. Extensions to BGP-LS ................................... 21
17. Discussion .................................................... 22
   17.1. HELLO Discussion ........................................ 22
   17.2. HELLO versus KEEPALIVE ................................. 22
18. VLANs/SVIs/Sub-interfaces ................................... 22
1. Introduction

The Massive Data Center (MDC) environment presents unusual problems of scale, e.g. O(10,000) devices, while its homogeneity presents opportunities for simple approaches. Approaches such as Jupiter Rising [JUPITER] use a central controller to deal with scaling, while BGP-SPF [I-D.ietf-lsvr-bgp-spf] provides massive scale-out without centralization using a tried and tested scalable distributed control plane, offering a scalable routing solution in Clos and similar environments. But BGP-SPF and similar higher level device-spanning protocols need link state and addressing data from the network to build the routing topology.

Link State Over Ethernet (LSoE) provides brutally simple mechanisms for devices to

- Discover each other’s Layer 2 (MAC) Addresses,
- Run Layer 2 keep-alive messages for session continuity,
- Discover each other’s unique IDs (ASN, RouterID, ...),
- Discover mutually supported encapsulations, e.g. IP/MPLS,
- Discover Layer 3 and/or MPLS addressing of interfaces of the link encapsulations,
- Enable layer 3 link liveness such as BFD, and finally
- Present these data, using a very restricted profile of a BGP-LS [RFC7752] API, to BGP-SPF which computes the topology and builds routing and forwarding tables.

This protocol may be more widely applicable to a range of routing and similar protocols which need link discovery and characterisation.
2. Terminology

Even though it concentrates on the Ethernet layer, this document relies heavily on routing terminology. The following are some possibly confusing terms:

ASN: Autonomous System Number [RFC4271], a BGP identifier for an originator of Layer 3 routes, particularly BGP announcements.

BGP-LS: A mechanism by which link-state and TE information can be collected from networks and shared with external components using the BGP routing protocol. See [RFC7752].

BGP-SPF: A hybrid protocol using BGP transport but a Dijkstra SPF decision process. See [I-D.ietf-lsvr-bgp-spf].

Clos: A hierarchic subset of a crossbar switch topology commonly used in data centers.

Datagram: The LSoE content of a single Ethernet frame. A full LSoE PDU may be packaged in multiple Datagrams.

Encapsulation: Address Family Indicator and Subsequent Address Family Indicator (AFI/SAFI). I.e. classes of addresses such as IPv4, IPv6, MPLS, ...

Frame: An Ethernet Layer 2 packet.

MAC Address: Media Access Control Address, essentially an Ethernet address, six octets. See [IEEE.802_2001].

MDC: Massive Data Center, commonly thousands of TORs.

MTU: Maximum Transmission Unit, the size in octets of the largest packet that can be sent on a medium, see [RFC1122] 1.3.3.

PDU: Protocol Data Unit, an LSoE application layer message. A PDU may need to be broken into multiple Datagrams to make it through MTU or other restrictions.

RouterID: An 32-bit identifier unique in the current routing domain, see [RFC4271] updated by [RFC6286].

Session: An established, via OPEN PDUs, session between two LSoE capable devices,

SPF: Shortest Path First, an algorithm for finding the shortest paths between nodes in a graph; AKA Dijkstra’s algorithm.

TOR: Top Of Rack switch, aggregates the servers in a rack and connects to aggregation layers of the Clos tree, AKA the Clos spine.

ZTP: Zero Touch Provisioning gives devices initial addresses, credentials, etc. on boot/restart.

3. Background

LSoE assumes a datacenter scale and topology, but can accommodate richer topologies which contain potential cycles.
While LSoE is designed for the MDC, there are no inherent reasons it could not run on a WAN; though, as it is simply a discovery protocol, it is not clear that this would be useful. The authentication and authorisation needed to run safely on the WAN are not provided in detail in this version of the protocol, although future versions/extensions could expand on them.

LSoE assumes a new IEEE assigned EtherType (TBD).

As encapsulations may have an inordinate number of addresses, and security will further add to the length of PDUs, LLDP’s limitation to 1,500 octets is judged to be too limiting.

4. Top Level Overview
   - Devices discover each other on Ethernet links
   - MAC addresses and Link State are exchanged over Ethernet
   - Layer 2 Liveness Checks are begun
   - Encapsulation data are exchanged and IP-Level Liveness Checks done
   - A BGP-like protocol is assumed to use these data to discover and build a topology database
There are two protocols, the Ethernet discovery and the interface to the upper level BGP-like protocol:

- Layer 2 Ethernet protocols are used to exchange Layer 2 data, i.e. MAC addresses, and layer 2.5 and 3 identifiers (not payloads), i.e. ASNs, Encapsulations, and interface addresses.

- A Link Layer to BGP API presents these data up the stack to a BGP protocol or an other device-spanning upper layer protocol, presenting them using the BGP-LS BGP-like data format.

The upper layer BGP family routing protocols cross all the devices, though they are not part of these LSoE protocols.

To simplify this document, Layer 2 Ethernet framing is not shown.

5. Ethernet to Ethernet Protocols

Two devices discover each other and their respective MAC addresses by sending multicast HELLO PDUs (Section 9). To allow discovery of new devices coming up on a multi-link topology, devices send periodic HELLOs forever, see Section 17.1.
Once a new device is recognized, both devices attempt to negotiate and establish peering by sending unicast OPEN PDUs (Section 10). In an established peering, Encapsulations (Section 12) may be announced and modified. When two devices on a link have compatible Encapsulations and addresses, i.e. the same AFI/SAFI and the same subnet, the link is announced via the BGP-LS API.

5.1. Inter-Link Ether Protocol Overview

The HELLO, Section 9, is a priming message. It is an Ethernet multicast frame with a small LSoE PDU with the simple goal of discovering the Ethernet MAC address(es) of devices reachable via an interface.

The HELLO and OPEN, Section 10, PDUs, which are used to discover and exchange MAC address and IDs, are mandatory; other PDUs are optional; though at least one encapsulation MUST be agreed at some point.

The following is a ladder-style sketch of the Ethernet protocol exchanges:

```
<table>
<thead>
<tr>
<th>HELLO</th>
<th>MAC Address discovery</th>
</tr>
</thead>
<tbody>
<tr>
<td>HELLO</td>
<td>Mandatory</td>
</tr>
<tr>
<td>OPEN</td>
<td>MACs, IDs, and Capabilities</td>
</tr>
<tr>
<td>OPEN</td>
<td>Mandatory</td>
</tr>
<tr>
<td>Interface IPv4 Addresses</td>
<td>Interface IPv4 Addresses</td>
</tr>
<tr>
<td>ACK</td>
<td>Optional</td>
</tr>
<tr>
<td>Interface IPv4 Addresses</td>
<td></td>
</tr>
<tr>
<td>ACK</td>
<td></td>
</tr>
<tr>
<td>Interface IPv6 Addresses</td>
<td>Interface IPv6 Addresses</td>
</tr>
<tr>
<td>ACK</td>
<td>Optional</td>
</tr>
</tbody>
</table>
```
6. Transport Layer

LSoE PDU are carried by a simple transport layer which allows long PDUs to occupy multiple Ethernet frames. The LSoE data in each frame is referred to as a Datagram.

The LSoE Transport Layer encapsulates each Datagram using a common transport header.
The fields of the LSoE Transport Header are as follows:

Version: Version number of the protocol, currently 0. Values other than 0 are treated as errors.

L: A bit that set to 1 if this Datagram is the last Datagram of the PDU. For a PDU which fits in only one Datagram, it is set to one.

Datagram Number: 0..127, a monotonically increasing value, modulo 128, see [RFC1982].

Datagram Length: Total number of octets in the Datagram including all payloads and fields.

Checksum: A 32 bit hash over the Datagram to detect bit flips, see Section 7.

7. The Checksum

There is a reason conservative folk use a checksum in UDP. And as many operators stretch to jumbo frames (over 1,500 octets) longer checksums are the conservative approach.

For the purpose of computing a checksum, the checksum field itself is assumed to be zero.

The following code describes the suggested algorithm.

Sum up 32-bit unsigned ints in a 64-bit long, then take the high-order section, shift it right, rotate, add it in, repeat until zero.
# include <stddef.h>

/* The F table from Skipjack, and it would work for the S-Box */
static const uint8_t sbox[256] = {
  0xa3, 0xd7, 0x09, 0x83, 0xf8, 0x48, 0xf6, 0xf4, 0xf4, 0xb3, 0x21, 0x15, 0x78,
  0x99, 0xb1, 0xad, 0xf9, 0xe7, 0x2d, 0x4d, 0x8a, 0xce, 0x4c, 0xca, 0x2e,
  0x52, 0x95, 0xd9, 0x1e, 0x4e, 0x38, 0x44, 0x28, 0x0a, 0xdf, 0x02, 0xa0,
  0x17, 0xf1, 0x60, 0x68, 0x12, 0xb7, 0x7a, 0xc3, 0xe9, 0xfa, 0x3d, 0x53,
  0x96, 0x84, 0x6b, 0xba, 0xf2, 0x63, 0x9a, 0x19, 0x7c, 0xae, 0xe5, 0xf5,
  0xf7, 0x16, 0x6a, 0xa2, 0x39, 0xb6, 0x7b, 0x0f, 0xc1, 0x93, 0x81, 0x1b,
  0x0e, 0xb4, 0x1a, 0xea, 0xd0, 0x91, 0x2f, 0xb8, 0x55, 0xb9, 0xda, 0x85,
  0x3f, 0xb1, 0xbf, 0xe0, 0x5a, 0x58, 0x80, 0x5f, 0x66, 0x0b, 0xd8, 0x90,
  0x35, 0xd5, 0xc0, 0xa7, 0x33, 0x06, 0x65, 0x69, 0x45, 0x00, 0x94, 0x56,
  0x9d, 0x98, 0x9b, 0x76, 0x97, 0xfc, 0xb2, 0xc2, 0xb0, 0xe0, 0xdb, 0x20,
  0xe1, 0xeb, 0xd6, 0xe4, 0xd4, 0x47, 0x4a, 0x1d, 0x42, 0xed, 0x9e, 0xe6,
  0x49, 0x3c, 0xcd, 0x43, 0x27, 0xda, 0x07, 0xd4, 0xde, 0xc7, 0x67, 0x18,
  0x89, 0x9f, 0x30, 0x1f, 0x8d, 0xc6, 0x8f, 0xaa, 0xc8, 0x74, 0xdc, 0xc9,
  0x5d, 0x5c, 0x31, 0xa4, 0x70, 0x88, 0x61, 0x2c, 0x9f, 0xd0, 0x2b, 0x87,
  0x50, 0x82, 0x54, 0x64, 0x26, 0x7d, 0x03, 0x40, 0x34, 0x4b, 0x1c, 0x73,
  0xd1, 0xc4, 0xdf, 0x3b, 0xcc, 0xfb, 0x7f, 0xab, 0xe6, 0x3e, 0x5b, 0xa5,
  0xad, 0x04, 0x23, 0x9c, 0xe1, 0x14, 0x51, 0xe2, 0xf0, 0x29, 0x79, 0x71, 0xe7,
  0xff, 0x8c, 0x9e, 0xce, 0xf0, 0xbc, 0x72, 0x75, 0x6f, 0x37, 0xa1,
  0xec, 0xd3, 0xa0, 0xf2, 0x88, 0xb0, 0xe2, 0x10, 0xe8, 0x80, 0x77, 0x11, 0xbe,
  0x92, 0x4f, 0x24, 0xc5, 0x32, 0x36, 0x99, 0xcf, 0x3f, 0xa6, 0xbb, 0xac,
  0xc6, 0xa9, 0x31, 0x57, 0xb5, 0xb5, 0xe3, 0xbd, 0xa8, 0x3a, 0x01,
  0x05, 0x59, 0x2a, 0x46
};

/* non-normative example C code, constant time even */

uint32_t sbox_checksum_32(const uint8_t *b, const size_t n)
{
  uint32_t sum[4] = {0, 0, 0, 0};
  uint64_t result = 0;
  for (size_t i = 0; i < n; i++)
    sum[i & 3] += sbox[*b++];
  for (int i = 0; i < sizeof(sum)/sizeof(*sum); i++)
    result = (result << 8) + sum[i];
  result = (result >> 32) + (result & 0xFFFFFFFF);
  result = (result >> 32) + (result & 0xFFFFFFFF);
  return (uint32_t) result;
}
8. TLV PDUs

The basic LSoE application layer PDU is a typical TLV (Type Length Value) PDU. It may be broken into multiple Datagrams, see Section 6

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      Type     |           PDU Length          |               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+               +
|                           Value ...                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

The fields of the basic LSoE header are as follows:

Type: An integer differentiating PDU payload types

0 - HELLO
1 - OPEN
2 - KEEPALIVE
3 - ACK
4 - IPv4 Announcement
5 - IPv6 Announcement
6 - MPLS IPv4 Announcement
7 - MPLS IPv6 Announcement
8-254 Reserved
255 - VENDOR

PDU Length: Total number of octets in the PDU including all payloads and fields

Value: Any application layer content of the LSoE PDU beyond the type.

9. HELLO

The HELLO PDU is unique in that it is a multicast Ethernet frame. It solicits response(s) from other device(s) on the link. See Section 17.1 for why multicast is used. The multicast MACs to be used MUST be one of the following, See Clause 9.2.2 of [IEEE802-2014]:

01-80-C2-00-00-0E: Nearest Bridge = Propagation constrained to a single physical link; stopped by all types of bridges (including MPRs (media converters)).
01-80-C2-00-00-03: Nearest non-TPMR Bridge = Propagation constrained by all bridges other than TPMRs; intended for use within provider bridged networks.
All other LSoE PDUs are unicast Ethernet frames, as the peer’s MAC Address is known after the HELLO exchange.

When an interface is turned up on a device, it SHOULD issue a HELLO periodically. The interval is set by configuration with a default of 60 seconds.

If more than one device responds, one adjacency is formed for each unique (source MAC address) response. LSoE treats the adjacencies as separate links.

When a HELLO is received from a source MAC Address with which there is no established LSoE adjacency, the receiver SHOULD respond with an OPEN PDU. The two devices establish an LSoE adjacency by exchanging OPEN PDUs.

The PDU Length is the octet count of the entire PDU, including the Type and the Datagram Length field itself.

10. OPEN

Each device has learned the other’s MAC address from the HELLO exchange, see Section 9. Therefore the OPEN and subsequent PDUs are unicast, as opposed to the HELLO’s multicast, Ethernet frames.

```
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Type = 1 | PDU Length |                   |               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     Nonce                     |               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+               +
|                                                               |
|                             My ID                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   AttrCount   |       Attribute List ...      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Auth Length     |                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Authentication Data ... |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
The Nonce enables detection of a duplicate OPEN PDU. It SHOULD be either a random number or time of day. It is needed to prevent session closure due to a repeated OPEN caused by a race or a dropped or delayed ACK.

My ID can be an ASN with high order bits zero, a classic RouterID with high order bits zero, a catenation of the two, a 80-bit ISO System-ID, or any other identifier unique to a single device in the topology. While a link is uniquely identified by a MAC pair, the same ID pair MAY occur on multiple links between the same two devices. IDs are big-endian.

AttrCount is the number of attributes in the Attribute List. Attributes are single octets whose semantics are user-defined.

A node may have zero or more user-defined attributes, e.g. spine, leaf, backbone, route reflector, arabica, ...

Attribute syntax and semantics are local to an operator or datacenter; hence there is no global registry. Nodes exchange their attributes only in the OPEN PDU.

Auth Length is a 16-bit field denoting the length in octets of the Authentication Data, not including the Auth Length itself. If there are no Authentication Data, the Auth Length MUST BE zero.

The Authentication Data are specific to the operational environment. A failure to authenticate is a failure to start the LSoE session, an ERROR PDU is sent (Error Code 2), and HELLOs MUST be restarted.

Once two devices know each other’s MAC addresses, and have ACKed each other’s OPEN PDUs, Layer 2 KEEPALIVEs (see Section 13) SHOULD be started to ensure Layer 2 liveness and keep the session semantics alive. The timing and acceptable drop of KEEPALIVE PDUs is discussed in Section 13.

If a sender of OPEN does not receive an ACK of the OPEN PDU Type, then they MUST resend the same OPEN PDU, with the same Nonce.

Resending an unacknowledged OPEN PDU, like other ACKed PDUs, SHOULD use exponential back-off, see [RFC1122].

If a properly authenticated OPEN arrives with a new Nonce from a device with which the receiving device believes it already has an LSoE session (OPENs have already been exchanged), the receiver MUST assume that the sending device has been reset. All discovered encapsulation date SHOULD be withdrawn via the BGP-LS API and the recipient MUST respond with a new OPEN. Then encapsulations SHOULD
NOT be kept because while the new OPEN is likely to be followed by new encapsulation PDUs of the same data, the old session might have an encapsulation type not in the new session.

11. ACK

The ACK PDU acknowledges receipt of a PDU and reports any error condition which might have been raised.

```
+-------------------+-------------------+
| Type = 3           | PDU Length = 8    |
| +-------------------+-------------------+
| EType             | Error Code        |
+-------------------+-------------------+
```

The ACK acknowledges receipt of an OPEN, Encapsulation, VENDOR PDU, etc.

The PDU Type is the Type of the PDU being acknowledged, e.g., OPEN or one of the Encapsulations.

If there was an error processing the received PDU, then the EType is non-zero. If the EType is zero, Error Code and Error Hint MUST also be zero.

A non-zero EType is the receiver’s way of telling the PDU’s sender that the receiver had problems processing the PDU. The Error Code and Error Hint will tell the sender more detail about the error.

The decimal value of EType gives a strong hint how the receiver sending the ACK believes things should proceed:

0 - No Error, Error Code and Error Hint MUST be zero
1 - Warning, something not too serious happened, continue
2 - Session should not be continued, try to restart
3 - Restart is hopeless, call the operator
4-15 - Reserved

Someone stuck in the 1990s might think of the error codes as 0x1zzz, 0x2zzz, etc. They might be right. Or not.

The Error Code indicates the type of error.

The Error Hint is any additional data the sender of the error PDU thinks will help the recipient or the debugger with the particular error.
11.1. Retransmission

If a PDU sender expects an ACK, e.g. for an OPEN, an Encapsulation, a VENDOR PDU, etc., and does not receive the ACK for a configurable time (default one second), the sender resends the PDU using exponential back-off, see [RFC1122]. This cycle MAY be repeated a configurable number of times (default three) before it is considered a failure. The session is considered closed in case of this ACK failure.

12. The Encapsulations

Once the devices know each other’s MAC addresses, know each other’s upper layer identities, have means to ensure link state, etc., the LSoE session is considered established, and the devices SHOULD announce their interface encapsulations, addresses, (and labels).

The Encapsulation types the peers exchange may be IPv4 Announcement (Section 12.3), IPv6 Announcement (Section 12.4), MPLS IPv4 Announcement (Section 12.6), MPLS IPv6 Announcement (Section 12.7), and/or possibly others not defined here.

The sender of an Encapsulation PDU MUST NOT assume that the peer is capable of the same Encapsulation Type. An ACK (Section 11) merely acknowledges receipt. Only if both peers have sent the same Encapsulation Type is it safe to assume that they are compatible for that type.

A receiver of an encapsulation might recognize an addressing conflict, such as both ends of the link trying to use the same address. In this case, the receiver SHOULD respond with an ERROR (Error Code 1) instead of an ACK. As there may be other usable addresses or encapsulations, this error might log and continue, letting an upper layer topology builder deal with what works.

Further, to consider a link of a type to formally be established so that it may be pushed up to upper layer protocols, the addressing for the type must be compatible, e.g. on the same IPvX subnet.

12.1. The Encapsulation PDU Skeleton

The header for all encapsulation PDUs is as follows:
The 16-bit Count is the number of Encapsulations in the Encapsulation list.

If the length of an Encapsulation PDU exceeds the Datagram size limit on media, the PDU is broken into multiple Datagrams. See Section 8.

The Receiver MUST acknowledge the Encapsulation PDU with a Type=3, ACK PDU (Section 11) with the Encapsulation Type being that of the encapsulation being announced, see Section 11.

If the Sender does not receive an ACK in a configurable interval (default one second), they SHOULD retransmit. After a user configurable number of failures, the LSoE session should be considered dead and the OPEN process SHOULD be restarted.

An Encapsulation PDU describes zero or more addresses of the encapsulation type.

An Encapsulation PDU of Type T replaces all previous encapsulations of Type T.

To remove all encapsulations of Type T, the sender uses a Count of zero.

If an interface has multiple addresses for an encapsulation type, one and only one address SHOULD be configured to be marked as primary, see Section 12.2.

Loopback addresses are generally not seen directly on an external interface. One or more loopback addresses MAY be exposed by configuration on one or more LSoE speaking external interfaces, e.g. for iBGP peering. They SHOULD be marked as such, see Section 12.2.

If there is exactly one non-loopback address for an encapsulation type on an interface, it SHOULD be marked as primary.

If a sender has multiple links on the same interface, separate data, ACKs, etc. must be kept for each peer.
Over time, multiple Encapsulation PDUs may be sent for an interface as configuration changes.

12.2. Prim/Loop Flags

```
  0 1 2 3 4 5 6 7
+---------------+---------------+---------------+---------------+
| Primary       | Loopback      | Reserved ...  |               |
+---------------+---------------+---------------+---------------+
```

Each Encapsulation interface address MAY be marked as a primary address, and/or a loopback, in which case the respective bit is set to one.

Only one address MAY be marked as primary for an encapsulation type.

12.3. IPv4 Encapsulation

The IPv4 Encapsulation describes a device’s ability to exchange IPv4 packets on one or more subnets. It does so by stating the interface’s addresses and the corresponding prefix lengths.

```
  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Type = 4 | PDU Length | Count |
+---+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| ... | PrimLoop Flags | IPv4 Address |
+---+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| ... | PrefixLen | more ... |
+------------------------------------------------------------------+
```

The 16-bit Count is the number of IPv4 Encapsulations.

12.4. IPv6 Encapsulation

The IPv6 Encapsulation describes a device’s ability to exchange IPv6 packets on one or more subnets. It does so by stating the interface’s addresses and the corresponding prefix lengths.
### 12.5. MPLS Label List

As an MPLS enabled interface may have a label stack, see [RFC3032], a variable length list of labels is needed.

```
<table>
<thead>
<tr>
<th>Label Count</th>
<th>Label</th>
<th>Exp</th>
<th>S</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

A Label Count of zero is an implicit withdraw of all labels for that prefix on that interface.

### 12.6. MPLS IPv4 Encapsulation

The MPLS IPv4 Encapsulation describes a device’s ability to exchange labeled IPv4 packets on one or more subnets. It does so by stating the interface’s addresses the corresponding prefix lengths, and the corresponding labels.

```
<table>
<thead>
<tr>
<th>PrefixLen</th>
<th>more</th>
</tr>
</thead>
</table>
```

The 16-bit Count is the number of IPv6 Encapsulations.
The 16-bit Count is the number of MPLSv6 Encapsulations.

12.7.  MPLS IPv6 Encapsulation

The MPLS IPv6 Encapsulation describes a device’s ability to exchange labeled IPv6 packets on one or more subnets. It does so by stating the interface’s addresses, the corresponding prefix lengths, and the corresponding labels.

The 16-bit Count is the number of MPLSv6 Encapsulations.

13.  KEEPALIVE - Layer 2 Liveness

LSOE devices MUST beacon occasional Layer 2 KEEPALIVE PDUs to ensure session continuity.
They SHOULD be beaconed at a configured frequency. One per second is the default. Layer 3 liveness, such as BFD, will likely be more aggressive.

If a KEEPALIVE is not received from a peer with which a receiver has an open session for a configurable time (default one minute), the session SHOULD BE presumed closed. The devices MAY keep configuration state until a new session is established and new Encapsulation PDUs are received.

```
   0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3
+-----------------------------------------------+
|    Type = 2    |           Length = 3           |
+-----------------------------------------------+
```

14. VENDOR - Vendor Extensions

```
   0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----------------------------------------------+
|   Type = 255   |             Length             |      ...    |
+-----------------------------------------------+
|               Enterprise Number               |    Ent Type   |
+-----------------------------------------------+
|                      Enterprise Data ...                  |
+-----------------------------------------------+
```

Vendors or enterprises may define TLVs beyond the scope of LSoE standards. This is done using a Private Enterprise Number [IANA-PEN] followed by free form data.

Ent Type allows a VENDOR PDU to be sub-typed in the event that the vendor/enterprise needs multiple PDU types.

As with Encapsulation PDUs, a receiver of a VENDOR PDU MUST respond with an ACK or an ERROR PDU. Similarly, a VENDOR PDU MUST only be sent over an open session.

15. Layers 2.5 and 3 Liveness

Ethernet liveness is continuously tested by KEEPALIVE PDUs, see Section 13. As layer 2.5 or layer 3 connectivity could still break, liveness above layer 2 SHOULD be frequently tested using BFD ([RFC5880]) or a similar technique.

This protocol assumes that one or more Encapsulation addresses will be used to ping, BFD, or whatever the operator configures.
16. The North/South Protocol

Thus far, a one-hop point-to-point link discovery protocol has been defined.

The nodes know the unique node identifiers (ASNs, RouterIDs, ...) and Encapsulations on each link interface.

Full topology discovery is not appropriate at the Ethernet layer, so Dijkstra a la IS-IS etc. is assumed to be done by higher level protocols.

Therefore the node identifiers, link Encapsulations, and state changes are pushed North via a small subset of the BGP-LS API. The upper layer routing protocol(s), e.g. BGP-SPF, learn and maintain the topology, run Dijkstra, and build the routing database(s).

For example, if a neighbor’s IPv4 Encapsulation address changes, the devices seeing the change push that change Northbound.

16.1. Use BGP-LS as Much as Possible

BGP-LS [RFC7752] defines BGP-like Datagrams describing link state (links, nodes, link prefixes, and many other things), and a new BGP path attribute providing Northbound transport, all of which can be ingested by upper layer protocols such as BGP-SPF; see Section 4 of [I-D.ietf-lsvr-bgp-spf].

For IPv4 links, TLVs 259 and 260 are used. For IPv6 links, TLVs 261 and 262. If there are multiple addresses on a link, multiple TLV pairs are pushed North, having the same ID pairs.

16.2. Extensions to BGP-LS

The Northbound protocol needs a few minor extensions to BGP-LS. Luckily, others have needed the same extensions.

Similarly to BGP-SPF, the BGP protocol is used in the Protocol-ID field specified in table 1 of [I-D.ietf-idr-bgppls-segment-routing-epe]. The local and remote node descriptors for all NLRI are the ID’s described in Section 10. This is equivalent to an adjacency SID or a node SID if the address is a loopback address.

Label Sub-TLVs from [I-D.ietf-idr-bgp-.ls-segment-routing-ext] Section 2.1.1, are used to associate one or more MPLS Labels with a link.
17. Discussion

This section explores some trade-offs taken and some considerations.

17.1. HELLO Discussion

There is the question of whether to allow an intermediate switch to be transparent to discovery. We consider that an interface on a device is a Layer 2 or a Layer 3 interface. In theory it could be a Layer 3 interface with no encapsulation or Layer 3 addressing currently configured.

A device with multiple Layer 2 interfaces, traditionally called a switch, may be used to forward frames and therefore packets from multiple devices to one interface, I, on an LSoE speaking device. Interface I could discover a peer J across the switch. Later, a prospective peer K could come up across the switch. If I was not still sending and listening for HELLOs, the potential peering with K could not be discovered. Therefore, interfaces MUST continue to send HELLOs as long as they are turned up.

17.2. HELLO versus KEEPALIVE

Both HELLO and KEEPALIVE are periodic. KEEPALIVE might be eliminated in favor of keeping only HELLOs. But currently KEEPALIVE is unicast, and thus less noisy on the network, especially if HELLO is configured to transit layer-2-only switches.

This warrants discussion.

18. VLANs/SVIs/Sub-interfaces

One can think of the protocol as an instance (i.e. state machine) which runs on each link of a device.

As the upper routing layer must view VLAN topologies as separate graphs, LSoE treats VLAN ports as separate links.

LSoE PDUs learned over VLAN-ports may be interpreted by upper layer-3 routing protocols as being learned on the corresponding layer-3 SVI interface for the VLAN.

As Sub-Interfaces each have their own MAC, they act as separate interfaces, forming their own links.
19. Implementation Considerations

An implementation SHOULD provide the ability to configure an interface as LSoE speaking or not.

An implementation SHOULD provide the ability to configure whether HELLOs on an LSoE enabled interface send Nearest Bridge or Nearest non-TPMR Bridge multicast frames from that interface; see Section 9.

An implementation SHOULD provide the ability to distribute one or more loopback addresses or interfaces into LSoE on an external LSoE speaking interface.

An implementation SHOULD provide the ability to configure one of the addresses of an encapsulation as primary on an LSoE speaking interface. If there is only one address for a particular encapsulation, the implementation MAY mark it as primary by default.

20. Security Considerations

The protocol as it is MUST NOT be used outside a datacenter or similarly closed environment due to lack of formal definition of the authentication and authorisation mechanism. These will be worked on in a later effort, likely using credentials configured using ZTP or similar configuration automation.

Many MDC operators have a strange belief that physical walls and firewalls provide sufficient security. This is not credible. All MDC protocols need to be examined for exposure and attack surface.

It is generally unwise to assume that on the wire Ethernet is secure. Strange/unauthorized devices may plug into a port. Mis-wiring is very common in datacenter installations. A poisoned laptop might be plugged into a device’s port.

Malicious nodes/devices could mis-announce addressing, form malicious sessions, etc.

If OPENs are not being authenticated, an attacker could forge an OPEN for an existing session and cause the session to be reset.

For these reasons, the OPEN PDU’s authentication data exchange SHOULD be used. [A mandatory to implement authentication is in development.]
21. IANA Considerations

This document requests the IANA create a registry for LSoE PDU Type, which may range from 0 to 255. The name of the registry should be LSoE-PDU-Type. The policy for adding to the registry is RFC Required per [RFC5226], either standards track or experimental. The initial entries should be the following:

<table>
<thead>
<tr>
<th>PDU Code</th>
<th>PDU Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>HELLO</td>
</tr>
<tr>
<td>1</td>
<td>OPEN</td>
</tr>
<tr>
<td>2</td>
<td>KEEPALIVE</td>
</tr>
<tr>
<td>3</td>
<td>ACK</td>
</tr>
<tr>
<td>4</td>
<td>IPv4 Announce / Withdraw</td>
</tr>
<tr>
<td>5</td>
<td>IPv6 Announce / Withdraw</td>
</tr>
<tr>
<td>6</td>
<td>MPLS IPv4 Announce / Withdraw</td>
</tr>
<tr>
<td>7</td>
<td>MPLS IPv6 Announce / Withdraw</td>
</tr>
<tr>
<td>8-254</td>
<td>Reserved</td>
</tr>
<tr>
<td>255</td>
<td>VENDOR</td>
</tr>
</tbody>
</table>

This document requests the IANA create a registry for LSoE PL Flag Bits, which may range from 0 to 7. The name of the registry should be LSoE-PL-Flag-Bits. The policy for adding to the registry is RFC Required per [RFC5226], either standards track or experimental. The initial entries should be the following:

<table>
<thead>
<tr>
<th>Bit</th>
<th>Bit Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Primary</td>
</tr>
<tr>
<td>1</td>
<td>Loopback</td>
</tr>
<tr>
<td>2-7</td>
<td>Reserved</td>
</tr>
</tbody>
</table>

This document requests the IANA create a registry for LSoE Error Codes, a 16 bit integer. The name of the registry should be LSoE-Error-Codes. The policy for adding to the registry is RFC Required per [RFC5226], either standards track or experimental. The initial entries should be the following:

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Error Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Reserved</td>
</tr>
<tr>
<td>1</td>
<td>Link Addressing Conflict</td>
</tr>
<tr>
<td>2</td>
<td>Authorisation Failure in OPEN</td>
</tr>
</tbody>
</table>
22. IEEE Considerations

This document requires a new EtherType.

23. Acknowledgments

The authors thank Cristel Pelsser for multiple reviews, Jeff Haas for review and comments, Joe Clarke for a useful review, John Scudder for deeply serious review and comments, Larry Kreeger for a lot of layer 2 clue, Martijn Schmidt for his contribution, Neeraj Malhotra for review, Russ Houssley for checksum discussion and sBox, and Steve Bellovin for checksum advice.

24. References

24.1. Normative References

[I-D.ietf-idr-bgp-ls-segment-routing-ext]

[I-D.ietf-idr-bgpls-segment-routing-epe]

[I-D.ietf-lsvr-bgp-spf]

[IANA-PEN]

[IEEE.802_2001]
[IEEE802-2014]
Institute of Electrical and Electronics Engineers, "Local and Metropolitan Area Networks: Overview and Architecture", IEEE Std 802-2014, 2014.


24.2. Informative References


Authors’ Addresses

Randy Bush
Arrcus & IIJ
5147 Crystal Springs
Bainbridge Island, WA  98110
United States of America

Email: randy@psg.com

Rob Austein
Arrcus, Inc

Email: sra@hactrn.net

Keyur Patel
Arrcus
2077 Gateway Place, Suite #400
San Jose, CA  95119
United States of America

Email: keyur@arrcus.com
Abstract

In an EVPN network, EVPN PEs provide VPN bridging and routing service to connected CE devices based on BGP EVPN control plane. At present, there is no PE-CE control plane defined for an EVPN PE to learn CE MAC, IP, and any other routes from a CE that may be distributed in EVPN control plane to enable unicast flows between CE devices. As a result, EVPN PEs rely on data plane based gleaning of source MACs for CE MAC learning, ARP/ND snooping for CE IPv4/IPv6 learning, and in some cases, local configuration for learning prefix routes behind a CE. A PE-CE control plane alternative to this traditional learning approach, where applicable, offers certain distinct advantages that in turn result in simplified EVPN operation.

This document defines a PE-CE control plane as an optional alternative to traditional non-control-plane based PE-CE learning in an EVPN network. It defines PE-CE control plane procedures and TLVs based on LSoE as the base protocol, enumerates advantages that may be achieved by using this PE-CE control plane, and discusses in detail EVPN use cases that are simplified as a result.

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".
Table of Contents

1 Introduction ........................................... 4
1.1 Terminology ......................................... 5
2. PE <-> CE Control Plane Overview .................. 7
3. TLVs .................................................. 9
  3.1 Overlay IPv4 Encapsulation PDU .................... 9
  3.2 Overlay IPv6 Encapsulation PDU .................... 10
  3.3 Overlay IPv4 Prefix Encapsulation PDU .......... 12
  3.4 Overlay IPv6 Prefix Encapsulation PDU .......... 13
4. CE MAC/IP Learning on a PE AC ....................... 14
  4.1 PE <-> CE LSoE Session Establishment .......... 14
  4.2 CE MAC/IP Learning ................................. 14
5. PE Any-cast GW MAC/IP Learning on CE .............. 15
6. Remote CE MAC/IP Learning on CE .................... 15
7. PE <-> CE Control Plane with EVPN All-active Multi-Homing ... 16
  7.1 All-active Multi-Homing Mode ..................... 16
  7.2 Source MAC ......................................... 17
  7.3 CE MAC/IP Learning with EVPN All-active Multi-Homing ... 17
  7.4 LAG Member Link Failure .......................... 18
    7.4.1 Session Re-establishment ...................... 18
    7.4.2 TLV Retention .................................. 18
  7.4 LAG Failure ........................................ 18
  7.5 Example PE <-> CE Control Plane Flow with All-active
Multi-Homing ........................................... 19
8. Software Neighbor Tables ................................. 21
9. MAC/IP Learning Conflict Resolution .................. 21
10. PE-CE Overlay Prefix Learning ....................... 22
11. Asymmetric EVPN-IRB .................................. 22
12. Centralized Gateway EVPN-IRB ......................... 22
13. Use Cases .............................................. 22
    13.1 Simplified EVPN Operations ....................... 22
        13.1.1 EVPN All-active Multi-Homing .............. 23
        13.1.2 Convergence on CE Host Moves .............. 24
            13.1.2.1 Silent Hosts .......................... 24
            13.1.2.2 Probing ............................. 25
        13.1.3 ARP Gleaning Latency ....................... 26
    13.2 Applicability to non-EVPN Use Cases .......... 26
14. Summary .............................................. 26
15. References ........................................... 28
    15.1 Normative References ............................. 28
    15.2 Informative References ........................... 28
16. Acknowledgements .................................... 29
Contributors ............................................. 29
Authors’ Addresses ..................................... 29
1 Introduction

In an EVPN network, CE devices typically connect to an EVPN PE via layer-2 interfaces that terminate in a BD on the PE. Multi-homed LAG interfaces together with EVPN all-active multi-homing procedures are used to achieve PE-CE link and PE node redundancy for fault-tolerance and load-balancing. PEs provide overlay bridging and, optionally, first-hop routing service for these CE devices based on an EVPN control plane that is used to distribute CE MAC, IP, and prefix reachability across PEs.

At present, there is no PE-CE control plane defined for an EVPN PE to learn connected CE host MACs and IPs. As a result, EVPN PEs rely on:

- data plane based gleaning of source MAC for MAC learning,
- ARP snooping for IPv4 + MAC learning, and
- ND snooping for IPv6 + MAC learning.

A PE-CE control plane alternative to this traditional learning approach, where applicable, can offer some distinct advantages across various boot-up, mobility, and convergence scenarios:

- PE-CE learning is decoupled from non-deterministic hashing of data, ARP, and ND packets from CEs over all-active multi-homed LAG interfaces.
- PE-CE learning is decoupled from non-deterministic periodicity of data traffic from CEs or, in an extreme scenario, from CE device being silent for an extended period.
- PE-CE learning is decoupled from non-deterministic CE behavior with respect to unsolicited ARPs and NAs following boot-up and moves.
- PE-CE learning is decoupled from latencies associated with data packet triggered ARP and ND gleaning.

This in-turn results in simplification of certain EVPN operations such as aliasing, MAC and IP syncing across multi-homing PEs, and probing on MAC/IP moves. In addition, it helps achieve a deterministic convergence behavior across various boot-up, mobility, and failure scenarios.

A PE may also use local policy configuration for learning prefixes behind a CE that does not run a dynamic routing protocol. A PE-CE control plane can provide an operationally simpler alternative to local configuration for such use cases, where CE and PE devices are not under the same configuration management entity.

This document defines a new PE-CE control plane as an alternative to traditional data-plane and ARP/ND snooping based PE-CE host learning.
INTERNET DRAFT  LSoE-based PE-CE Control Plane for EVPN

and to local configuration-based PE-CE prefix learning. It defines PE-CE control plane procedures and TLVs based on [LSOE] as the base protocol, enumerates advantages that may be achieved by using this PE-CE control plane, and discusses in detail EVPN operations that are simplified as a result. Use of PE-CE control plane defined in this document is intended to be optional and backwards compatible with CEs that use traditional PE-CE learning within the same BD.

1.1 Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

The following terms are used in this document:

- LSoE: Link State over Ethernet Protocol defined in [LSOE]
- EVPN-IRB: A BGP-EVPN distributed control plane based integrated routing and bridging fabric overlay discussed in [EVPN-IRB]
- Underlay: IP or MPLS fabric core network that provides IP or MPLS routed reachability between EVPN PEs.
- Overlay: VPN or service layer network consisting of EVPN PEs OR VPN provider-edge (PE) switch-router devices that runs on top of an underlay routed core.
- EVPN PE: A PE switch-router in a data-center fabric that runs overlay BGP-EVPN control plane and connects to overlay CE host devices. An EVPN PE may also be the first-hop layer-3 gateway for CE/host devices. This document refers to EVPN PE as a logical function in a data-center fabric. This EVPN PE function may be physically hosted on a top-of-rack switching device (ToR) OR at layer(s) above the ToR in the Clos fabric. An EVPN PE is typically also an IP or MPLS tunnel end-point for overlay VPN flows.
- CE: A tenant host device that has layer 2 connectivity to an EVPN PE switch-router, either directly OR via intermediate switching device(s).
- Symmetric EVPN-IRB: An overlay fabric first-hop routing architecture as defined in [EVPN-IRB], wherein, overlay host-to-host routed inter-subnet flows are routed at both ingress and egress EVPN PEs.
- Asymmetric EVPN-IRB: An overlay fabric first-hop routing architecture as defined in [EVPN-IRB], wherein, overlay host-to-host routed inter-subnet flows are routed and bridged at ingress PE and bridged at egress PEs.
- Centralized EVPN-IRB: An overlay fabric first-hop routing architecture, wherein, overlay host-to-host routed inter-subnet...
flows are routed at a centralized gateway, typically at the one of the spine layers, and where EVPN PEs are pure bridging devices.
- ARP: Address Resolution Protocol [RFC 826].
- ND: IPv6 Neighbor Discovery Protocol [RFC 4861].
- Ethernet-Segment: physical Ethernet or LAG port that connects an access device to an EVPN PE, as defined in [RFC 7432].
- ESI: Ethernet Segment Identifier as defined in [RFC 7432].
- LAG: Layer-2 link-aggregation, also known as layer-2 bundle port-channel, or bond interface.
- EVPN all-active multi-homing: PE-CE all-active multi-homing achieved via a multi-homed layer-2 LAG interface on a CE with member links to multiple PEs and related EVPN procedures on the PEs.
- EVPN Aliasing: multi-homing procedure as defined in [RFC 7432].
- BD: Broadcast Domain.
- Bridge Table: An instantiation of a broadcast domain on a MAC-VRF.
- AC: A PE Attachment Circuit. This may be an access (untagged) or trunk (tagged) layer-2 interface that is a member of a local VLAN or a BD.
2. PE <-> CE Control Plane Overview

The Link State over Ethernet (LSoE) protocol is defined in [LSOE] as a protocol over Ethernet links to auto-discover connected neighbor’s layer 2, layer 3 attributes, and encapsulations for the purpose of bringing up upper layer routing protocols. This document leverages LSoE as a PE-CE protocol in an EVPN network fabric on access links between an EVPN PE and CE. Specifically,

- PE-CE control plane based on LSoE protocol is proposed for CE MAC learning as an alternative to data-plane based source MAC learning.
- PE-CE control plane based on LSoE protocol is proposed for CE MAC-IP adjacency learning as an alternative to MAC-IP learning based on ARP/ND snooping.
- PE-CE control plane based on LSoE is proposed for learning of IP Prefixes and associated overlay indexes, as an alternative to local configuration on the PE for use case defined in section 4.1 of [EVPN-PREFIX-ADV].

Note that any specification related to base LSoE protocol itself is considered out of scope for this document and will continue to be covered in the base protocol spec. This document will instead focus on procedures and TLV extensions needed to achieve the above learning on PE-CE links in an EVPN network. Any text that relates to the base protocol included in this document is simply background information in the context of use cases covered in this document. The reader should refer to the base LSoE protocol document for the exact LSoE protocol specification.

![Diagram of LSoE Session](image_url)
An LSoE session is established on layer-2 access interfaces between the EVPN PE and each connected CE host device. A session end-point is identified by a peer device MAC address on a layer-2 interface. LSoE HELLO messages are used for end-point discovery and OPEN messages are exchanged between two end-points to establish an LSoE peering. Once LSoE peering is established, encapsulation TLVs are exchanged for learning.

In the context of an EVPN network, CE Attachment Circuits (AC logical interfaces) typically terminate in a BD on the PE, with multi-homed LAG interfaces used for EVPN all-active multi-homing. CE hosts may be directly connected to EVPN PEs via access ports, or may be connected on trunk-ports via another switch. In a common EVPN-IRB design, EVPN PEs also function as distributed first-hop gateways for hosts in a BD. While symmetric and asymmetric IRB designs are possible as discussed in [EVPN IRB], procedures described in subsequent sections assume symmetric IRB with distributed any-cast gateways on EVPN PEs. Any deviations from these procedures for asymmetric IRB design or a centralized IRB design will be covered in future updates to this document.

The next few sections will focus on additional LSoE TLVs and procedures needed for PE-CE learning on EVPN PE ACs without and with all-active multi-homing.
3. TLVs

This section defines new TLVs that are used by PE-CE control plane defined in this document.

3.1 Overlay IPv4 Encapsulation PDU

A new encapsulation PDU type is defined for the purpose of carrying overlay IPv4 and MAC bindings. Alternatively, it may also be used to carry an overlay MAC with a NULL IPv4 address in a non-IRB use case.

A new LSoE PDU type (8) is requested for this PDU.
- The IPv4 Address is that of an overlay.
- MAC address carries the MAC binding for the particular IPv4 address if one is set in the PDU. If an IPv4 address is not set, it simply signals an overlay MAC address.
- EVPN flag ‘E’ indicates if this encapsulation is being sent on behalf of a remote host learnt via EVPN. Use of this flag is covered in a later section.

This PDU is used to carry PE’s any-cast gateway IPv4 address and MAC bindings to a CE host device. Optionally, it may also be used to relay a remote CE’s IPv4 address and MAC bindings to a local CE host within a subnet, as well as to send local CE IPv4 address and MAC binding to the PE. Procedures related to use of this PDU are
discussed in subsequent sections.

In comparison to IPv4 Encapsulation PDU defined in [LSOE], this PDU allows you to explicitly signal a MAC binding that MAY be different from the device MAC used to establish an LSoE peering via HELLO/OPEN messages exchange.

The encapsulation list in this PDU MUST follow full replace semantics as in the LSoE protocol specification.

3.2 Overlay IPv6 Encapsulation PDU

A new encapsulation PDU type is defined for the purpose of carrying overlay IPv6 and MAC bindings:

```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Type = 9   |           PDU Length          |     Count     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|               |                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          IPv6 Address                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|               |   PrefixLen   |E|R|O|   Rsvd  |               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+               +
|                          MAC Address                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|               |                    more ...                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

**Figure 3**

- A new LSoE PDU type (9) is requested for this PDU.
- The IPv6 Address is that of an overlay.
- MAC address carries the MAC binding for IPv6 address in the PDU.
- An EVPN flag ‘E‘ indicates if this encapsulation is being sent on behalf of a remote host learnt via EVPN. Usage of this flag is covered in a later section.
- A Router flag ‘R’ is used to carry "Router Flag" or "R-bit" as defined in [RFC4861]. Usage of this flag for the purpose of installing ND cache entries based on learning via this TLV is as defined in [RFC4861]
o An Override flag ‘O’ is used to carry "Override Flag" or "O-bit" as defined in [RFC4861]. Usage of this flag for the purpose of installing ND cache entries based on learning via this TLV is as defined in [RFC4861]

This PDU is used to carry PE’s any-cast gateway IPv6 address and MAC bindings to a CE host device. Optionally, it may also be used to relay a remote CE’s IPv6 address and MAC bindings to a local CE within a subnet, as well as to send local CE IPv6 address and MAC bindings to the PE. Procedures related to usage of this PDU are discussed in subsequent sections.

The encapsulation list contained in this PDU MUST follow full replace semantics as in the LSoE protocol specification.
3.3 Overlay IPv4 Prefix Encapsulation PDU

A new encapsulation PDU type is defined for the purpose of carrying overlay IPv4 prefix routes for prefixes behind a CE that does not run a dynamic routing protocol for use-case as defined in section 4.1 of [EVPN-PREFIX-ADV]:

```
+-----------------+-----------------+-----------------+-----------------+
| Type = 10       | PDU Length      | Count           |
| Prefix Count    | IPv4 Prefix     | PrefixLen       |
| IPv4 Prefix     | IPv4 Prefix     | PrefixLen       |
| PrefixLen       | More...         | GW IP           |
| Rsvd            | More...         |                 |
+-----------------+-----------------+-----------------+-----------------+-----------------+-----------------+-----------------+-----------------+
```

Figure 4

A CE device as defined in [EVPN-PREFIX-ADV], with prefixes behind it MAY use the above PDU to send these prefixes to an EVPN PE with itself as the GW. An EVPN PE MAY then advertise prefixes received via this PDU as RT-5, with TS as the GW, as defined in [EVPN-PREFIX-ADV].

- A new LSoE PDU type (10) is requested for this PDU.
- IPv4 Prefix is set to a prefix behind a CE.
- PrefixLen is set to IPv4 prefix length for the advertised prefix.
- GW-IP is set to the CE IPv4 address (advertised via Type 8 PDU).

Multiple prefixes may be set for a single GW IP. The encapsulation list contained in this PDU MUST follow full replace semantics as in the LSoE protocol specification.
3.4 Overlay IPv6 Prefix Encapsulation PDU

A new encapsulation PDU type is defined for the purpose of carrying overlay IPv6 prefix routes for prefixes behind a CE that does not run a dynamic routing protocol for use-case as defined in section 4.1 of [EVPN-PREFIX-ADV]:

A CE device as defined in [EVPN-PREFIX-ADV], with prefixes behind it
MAY use the above PDU to send these prefixes to an EVPN PE with itself as the GW. An EVPN PE MAY then advertise prefixes received via this PDU as RT-5, with TS as the GW, as defined in [EVPN-PREFIX-ADV].

- A new LSoE PDU type (11) is requested for this PDU.
- IPv6 Prefix is set to an IPv6 prefix behind a CE.
- PrefixLen is set to IPv6 prefix length for the advertised prefix.
- GW-IP is set to the CE IPv6 address (advertised via Type 9 PDU).

Multiple prefixes may be set for a single GW IP. The encapsulation list contained in this PDU MUST follow full replace semantics as in the LSoE protocol specification.

4. CE MAC/IP Learning on a PE AC

This section defines procedures for learning a connected CE MAC and IP on a PE local attachment circuit (AC).

4.1 PE <-> CE LSoE Session Establishment

On an EVPN PE,

- A HELLO and/or OPEN PDU sent from a CE host source MAC is received on a tagged or untagged interface that is member of a local BD, referred here to as an AC.
- OPEN messages are exchanged with the host on the AC.
- LSoE session is established to the host source MAC and bound to a local AC.

4.2 CE MAC/IP Learning

Overlay IPv4 and IPv6 encapsulation PDU types 8/9 from a CE are used for the purpose of CE MAC/IP learning on a PE:

- The EVPN flag ‘E’ MUST NOT be set in type 8/9 PDU from a CE.
- A MAC entry for the MAC received in a type 8/9 PDU MUST be installed in the MAC-VRF table pointing to the AC to which the session is bound.
- If an IPv4/IPv6 address is set in the PDU, an IPv4/IPv6 neighbor binding MUST be established for the IPv4/IPv6 address in the PDU to the MAC address in the PDU. In other words, a next-hop re-write for these IPv4/IPv6 neighbor entries MUST be installed using the MAC address in the PDU, and if required by forwarding logic, bound to the AC associated with the LSoE session.
- Note that an IPv4/IPv6 address MAY NOT be set in a type 8/9 PDU received from a CE, in which case this PDU is only used for MAC learning. This MAY be the case in a non-IRB EVPN network, wherein, an EVPN PE is not a first-hop router for the attached CEs.
5. PE Any-cast GW MAC/IP Learning on CE

If LSoE based host learning is enabled on a PE with a distributed any-cast gateway on the EVPN PE,

- EVPN PE MUST send type 8/9 Overlay Encapsulation PDUs on associated ACs with LSoE sessions toward CE hosts.
- Type 8/9 PDUs from an EVPN PE MUST be encoded with the any-cast gateway IPv4/IPv6 address and any-cast gateway MAC address.
- EVPN flag ‘E’ MUST NOT be set in this PDU.
- A CE MAY process type 8/9 PDUs to establish GW IP to MAC bindings and learn gateway MAC to LAG AC bindings, similar to handling of type 8/9 PDUs on the PE described above.

Handling of type 8/9 PDUs for the purpose of gateway learning on the host is desirable but optional. A CE MAY continue to use ARP and ND for this purpose.

6. Remote CE MAC/IP Learning on CE

For CE to CE intra-subnet flows across the overlay, CE needs to learn and install a neighbor IP to MAC binding for remote CEs. This is handled today either by flooding ARP/ND requests across the overlay bridge and optionally implementing an ARP/ND suppression cache on the PE that is populated via MAC+IP EVPN route-type 2. ARP/ND request frames are trapped on the PE that does a local ARP/ND reply on behalf of the remote CE. If LSoE based learning is enabled in the fabric, LSoE may be used for this purpose to avoid overlay ARP/ND flooding, data frame triggered ARP learning, and to avoid maintaining an ARP suppression cache on the PE.

- Remote MAC-IP routes learned via BGP EVPN route-type 2 that are imported to a local MAC-VRF MAY also be sent as type 8/9 PDUs on LSoE sessions to CEs over local ACs in that BD.
- EVPN flag ‘E’ MUST be set in this encapsulation in the PDU.
- A CE MAY install IPv4/IPv6 neighbor MAC bindings for remote CEs within a subnet based on ‘E’ flagged type 8/9 PDUs received from the PE.

Handling of type 8/9 PDUs for this purpose is optional but desirable to get full benefit of a fabric that is completely setup on boot-up, avoids overlay flooding, and is decoupled from latencies associated with data plane driven ARP and ND learning.
7. PE <-> CE Control Plane with EVPN All-active Multi-Homing

In an EVPN all-active multi-homing setup, a LAG interface on the CE includes member physical ports that connect to multiple PE devices. A subset of these member ports that terminate at a PE are configured as members of a local LAG interface at that PE. A LAG AC at the PE is a logical interface in a BD, identified by this LAG interface and optionally, an Ethernet Tag in case of trunk ports.

In order for LSoE based learning to work with EVPN all-active multi-homing, a separate LSoE peering MUST be established between the CE host and each PE device. For this reason, while an EVPN PE MAY form an LSoE peering to a CE host on its local LAG AC, the CE host MUST form an LSoE peering to a PE on a local LAG "member physical port".

A configurable All-active Multi-Homing mode is defined below in order to be able to bind an LSoE peering to a LAG member-port as opposed to a LAG interface.

7.1 All-active Multi-Homing Mode

When configured to run on a local LAG port in this mode,

- LSoE HELLO messages MUST be replicated on ALL LAG member ports.
- An LSoE OPEN message sent in response to a HELLO MUST be sent on the LAG member port on which the HELLO was received.
- An LSoE session MUST be bound to the local LAG member port on
which the OPEN message was received.
- LSoE encapsulation PDUs MUST be sent on the local LAG member port on which the session was bound.
- LSoE Keep-Alives MUST be sent on the local LAG member port on which the session was bound.

Note that this may result in a PE receiving multiple HELLO PDUs from a CE MAC. This however is harmless, as per the [LSOE] specification. A PE simply drops redundant HELLOs from a MAC that it has already replied to with an OPEN, within a retry time window.

7.2 Source MAC

LSoE relies on the source MAC address in the Ethernet frame to establish a peering. When running LSoE on a LAG port (in all-active multi-homing mode or regular mode), LSoE frames MUST use the LAG interface MAC as the source MAC address in the Ethernet frame.

7.3 CE MAC/IP Learning with EVPN All-active Multi-Homing

In order to accomplish MAC/IP learning of CE host devices multi-homed to EVPN fabric PEs via EVPN All-active Multi-Homing:

- A multi-homed CE device MUST be configured to run LSoE on a local LAG interfaces in All-active Multi-Homing mode defined above.
- EVPN PE MAY run LSoE on local LAG interfaces to multi-homed CE devices in regular mode.
- EVPN PEs that share the same Ethernet Segment MUST use unique source MACs (that of the local LAG) in HELLO/OPEN messages to establish separate LSoE sessions to a CE.

With the above rules in place,

- An LSoE session on the CE is bound to a local LAG member-port.
- An LSoE session on the PE is bound to a local LAG AC port.
- A single LSoE session is established at the PE to a CE on the local LAG AC.
- ‘N’ LSoE sessions are established at the CE, one to each PE on a local LAG member interface, where N = number of multi-homing PEs in an Ethernet Segment.

Once an LSoE session is established as above, all other host learning procedures defined earlier for CE MAC/IP learning on a PE’s AC port apply as is to a LAG AC in an EVPN all-active multi-homing setup.
7.4 LAG Member Link Failure

On a CE that is running in all-active multi-homing mode, an LSoE session to a PE is bound to a LAG member interface. If the link that the LSoE session is bound to fails, LSoE session will get torn down at the CE by virtue of the session interface going down. If the CE has additional active member link(s) to this PE, a new LSoE session must be established on one of the active member links via HELLO PDUs sent by the CE on its remaining active member links to the PE.

7.4.1 Session Re-establishment

LSoE session at the CE is torn down immediately following the session interface failure. While the LAG interface at the PE is still operationally UP, LSoE session at the PE is subject to Keep Alive PDUs received from the CE. Once the session expires at the PE because of missed Keep Alive PDUs from the CE, PE will respond to HELLO on one of the active member link with an OPEN to re-establish a new session. Note that the new session is still bound to the LAG AC at the PE and to a new member link at the CE.

7.4.2 TLV Retention

TLVs learnt from a CE over a failed session MUST be retained at the PE if the PE LAG AC is still operationally up following a member link failure because of active member link(s) in the LAG. TLV retention logic at the PE MAY be based on an age-out time, that is a local matter at the PE. TLV age-out time MUST be higher than the missed Keep Alive duration, after which the session is considered closed. Once a new LSoE session is established, PE MUST implement a mark and sweep logic to reconcile retained TLVs from the CE peer with the new set of TLVs received from this CE.

7.4 LAG Failure

When a LAG member link failure results in the LAG interface being operationally down, TLV age-out logic discussed above MUST NOT be in effect. LSoE session MAY be be considered as DOWN immediately on the LAG being down at the PE. This is so that, in the event of a total connectivity loss between a PE and CE, CE learnt routes can be withdrawn immediately.
7.5 Example PE <-> CE Control Plane Flow with All-active Multi-Homing

An example LSoE over all-active multi-homing session flow is discussed below for clarity.

![Diagram showing LSoE example](https://example.com/diagram)

**Figure 7**

Example topology with CE H1 multi-homed to PE2 and PE3 via EVPN all-active multi-homing LAG with four member ports to each PE:

- H1 member ports to PE2: i121, i122, i123, i124
- PE2 member ports to H1: i211, i212, i213, i214
- H1 member ports to PE3: i131, i132, i133, i134
- PE3 member ports to H1: i311, i312, i313, i314
- H1 LAG port to PE2/PE3: MLAG1
- PE2 LAG port to H1: LAG2
- PE3 LAG port to H1: LAG3
- H1 LAG MAC: LMAC1
- PE2 LAG MAC: LMAC2
- PE3 LAG MAC: LMAC3

- H1 running LSoE on MLAG1 in All-active Multi-Homing mode
- PE2 running LSoE on LAG2 in regular mode
- PE3 running LSoE on LAG3 in regular mode
<table>
<thead>
<tr>
<th>PE2</th>
<th>HELLOs</th>
<th>H1</th>
<th>HELLOs</th>
</tr>
</thead>
<tbody>
<tr>
<td>LAG2</td>
<td>&lt;--------</td>
<td>HELLOs</td>
<td>-------</td>
</tr>
<tr>
<td>LAG2</td>
<td>&lt;--------</td>
<td>HELLOs</td>
<td>-------</td>
</tr>
<tr>
<td>LAG2</td>
<td>&lt;--------</td>
<td>HELLOs</td>
<td>-------</td>
</tr>
<tr>
<td>LAG2</td>
<td>&lt;--------</td>
<td>HELLOs</td>
<td>-------</td>
</tr>
<tr>
<td></td>
<td>OPEN</td>
<td>&lt;------</td>
<td>OPEN</td>
</tr>
<tr>
<td></td>
<td>LAG2</td>
<td>i122</td>
<td>i132</td>
</tr>
<tr>
<td></td>
<td>OPEN</td>
<td>------</td>
<td>-------</td>
</tr>
<tr>
<td></td>
<td>LAG2</td>
<td>i122</td>
<td>i132</td>
</tr>
</tbody>
</table>

Session to  
LMAC1 on LAG2  
Session to  
LMAC2 on i122  
Session to  
LMAC3 on i132  
Session to  
LMAC1 on LAG3

<table>
<thead>
<tr>
<th>PE2</th>
<th>Encap-PDU</th>
<th>Encap-PDU</th>
</tr>
</thead>
<tbody>
<tr>
<td>LAG2</td>
<td>&lt;--------</td>
<td>Encap-PDU</td>
</tr>
<tr>
<td>LAG2</td>
<td>i122</td>
<td>i132</td>
</tr>
<tr>
<td></td>
<td>ACK</td>
<td>ACK</td>
</tr>
<tr>
<td></td>
<td>&lt;--------</td>
<td>ACK</td>
</tr>
</tbody>
</table>
|      | LAG3     | LAG3     

<table>
<thead>
<tr>
<th>PE2</th>
<th>Overlay-PDU</th>
<th>Overlay-PDU</th>
</tr>
</thead>
<tbody>
<tr>
<td>LAG2</td>
<td>&lt;--------</td>
<td>Overlay-PDU</td>
</tr>
<tr>
<td>LAG2</td>
<td>i122</td>
<td>i132</td>
</tr>
<tr>
<td></td>
<td>ACK</td>
<td>ACK</td>
</tr>
<tr>
<td></td>
<td>&lt;--------</td>
<td>ACK</td>
</tr>
</tbody>
</table>
|      | LAG3     | LAG3     

Figure 8

In an example flow shown above:

- H1: originates HELLO(SMAC=LMAC2) on all MLAG member ports
- PE2: Multiple HELLO(SMAC=LMAC2) copies received on port LAG2
- PE3: Multiple HELLO(SMAC=LMAC2) copies received on port LAG3
- PE2: A single OPEN(SMAC=LMAC2, DMAC=LMAC1) sent on port LAG2
- PE3: A single OPEN(SMAC=LMAC3, DMAC=LMAC1) sent on port LAG3
- PE2/PE3: duplicate HELLOs from same source LMAC2 are ignored
- H1: OPEN(SMAC=LMAC2, DMAC=LMAC1) received on member port i122
- H1: OPEN(SMAC=LMAC1, DMAC=LMAC2) sent on member port i122
- H1: Session established to LMAC2 on MLAG1 member port i122
8. Software Neighbor Tables

Some networking stack implementations rely on ARP and ND populated neighbor tables for software forwarding. In order to inter-work with such an implementation, an LSoE learned IPv4/IPv6 neighbor entry MAY also be installed in ARP and ND neighbor table as a static/permanent entry.

In addition,

- Pre-installing LSoE learned neighbor entries may help reduce potential conflict with ARP or ND learned neighbor entries.
- Pre-installing LSoE learned neighbor entries may help reduce reliance on data traffic triggered ARP requests/ND solicitations and associated learning latency.

With respect to installing IPv6 entries learnt via LSoE in IPv6 ND cache, Router flag (R-bit) and Override flag (O-bit) received in LSoE PDU should be handled as defined in [RFC4861].

9. MAC/IP Learning Conflict Resolution

If LSoE learned neighbor entries are not already installed as static entries in ARP/ND neighbor table, it is possible that a neighbor IPv4/IPv6 adjacency may be learned both via LSoE and ARP/ND. Even if LSoE learned entries were pre-installed in neighbor table, a race condition is still possible leading to a potential conflict between ARP/ND learned and LSoE learned neighbor IP adjacency. In such scenarios, LSoE learned entry should be preferred for the purpose of programming neighbor IP adjacencies in forwarding.

With respect to MAC-VRF entries, it is recommended that data plane learning be turned off when LSoE based learning is enabled. However, if it is not, data plane learned entries MUST be reconciled with LSoE learned entries in software and, in case of a conflict, LSoE learned entries preferred if LSoE based learning is enabled.
10. PE-CE Overlay Prefix Learning

[EVPN-PREFIX-ADV] section 4.1 defines a use case, wherein, a PE may advertise IP prefixes and subnets behind a CE. In this use case, CE device does not run a dynamic routing protocol. Instead, these prefixes are learnt on the PE via local policy or configuration. Prefixes are then advertised by PE as RT-5 with the CE as the GW.

PE-CE control plane defined in this document MAY be used to learn these prefixes from a CE as an alternative to local configuration on the PE. Once an LSoE session is established between a CE and a PE, as discussed earlier,

- A CE MAY send type 10/11 PDUs with these IPv4/IPv6 prefixes over an LSoE session to a PE with the CE IP as the GW IP.
- A PE MAY advertise prefixes learnt via type 10/11 PDUs as RT-5 with CE IP as the GW IP.

To summarize, A PE would advertise:

- RT-2 for the CE MAC-IP learnt via type 8/9 PDU
- RT-5 for Prefixes learnt via type 10/11 PDU with GW IP = CE IP

11. Asymmetric EVPN-IRB

Any deviations from the above procedures proposed in this document for asymmetric IRB design will be covered in subsequent updates to this document.

12. Centralized Gateway EVPN-IRB

Any deviations from the above procedures proposed in this document for centralized GW based IRB design will be covered in subsequent updates to this document.

13. Use Cases

13.1 Simplified EVPN Operations

This section will discuss in detail, benefits and simplifications that may be achieved in the context of an EVPN network, if one chooses to implement PE-CE control plane defined in this document as opposed to using traditional data-plane and ARP/ND snooping based PE-CE learning.
Data plane and ARP/ND snooping based MAC/IP learning on PE-CE all-active multi-homed LAG ports is subject to unpredictable hashing of ARP, ND, and data frames from host to PE. As an example, an ARP request for a connected host might originate at PE1 but the resulting ARP response from the host might be received at PE2. Redundant EVPN PEs in all-active multi-homing mode typically handle this unpredictability via combination of methods below:

- PEs can handle unsolicited ARP and ND response frames.
- PEs can implement additional mechanism to SYNC ARP, ND, and MAC tables across all PEs in a redundancy group for optimal forwarding to locally connected hosts.
- PEs can implement EVPN aliasing procedures discussed in [RFC 7432] OR re-originate SYNCed MAC-IP adjacencies as local RT-2 to achieve MAC ECMP across the overlay.
- PEs can also re-originate SYNCed MAC-IP adjacencies as local RT-2 to achieve IP ECMP across the overlay OR implement IP aliasing procedures discussed in [EVPN-IP-ALIASING].
- PEs can also ensure EVPN sequence number SYNC for local MAC entries for EVPN mobility procedures to work correctly, as discussed in [EVPN-IRB-MOBILITY].

The PE-CE control plane learning alternative defined in this document fully decouples MAC and IP learning over MLAG ports from unpredictable hashing of data, AR, ND frames on all-active multi-
homed LAG member links. As a result, above procedures that essentially result from data-plane PE-CE learning on all-active multi-homed LAGs can be simplified via the PE-CE control plane alternative defined in this document.

13.1.2 Convergence on CE Host Moves

Host mobility across EVPN PE switches is a common occurrence in a data center fabric for flexibility in work load placement across a DC. Further, a host move must result in minimal, if any, disruption to traffic flows / services to / from the device.

Data plane and ARP/ND snooping based PE-CE learning may result in unpredictable convergence times, following host moves for the following cases:

- A host may or may not send any data packet immediately following a move.
- A host may or may not send an unsolicited ARP following a move.

While probing procedures, discussed in the next sub-sections are typically used to minimize convergence time, certain scenarios discussed below may still result in extended convergence times and flooding.

13.1.2.1 Silent Hosts

If a host is silent for an extended period following a move from PE1 to PE2, any bridged traffic flow destined to this host will continue to be black-holed by PE1 until the MAC ages out at PE1. Once the the MAC ages out at PE1, any bridged traffic flow destined to the host is
INTERNET DRAFT  LSoE-based PE-CE Control Plane for EVPN

flooded across the overlay bridge. Flooding of unknown unicast traffic on the overlay is enabled for this purpose. In summary, PE-CE learning that is based on data-plane and AR/ND snooping may be subject to non-deterministic convergence time and flooding following host moves because of being heavily dependent on unpredictable CE behavior.

PE-CE control plane based learning defined in this document fully decouples convergence in such scenarios from non-deterministic data flows and unsolicited ARP/ND behavior on a CE.

13.1.2.2 Probing

ARP and ND probing procedures are typically used to achieve host re-learning and convergence following host moves across the overlay:

- Following a host move from PE1 to PE2, the host’s MAC is discovered at PE2 as a local MAC via a data frames received from the host. If PE2 has a prior REMOTE MAC-IP host route for this MAC from PE1, an ARP probe is typically triggered at PE2 to learn the MAC-IP as a local IP adjacency and triggers EVPN RT-2 advertisement for this MAC-IP across the overlay with new reachability via PE2.

- Following a host move from PE1 to PE2, once PE1 receives a MAC or MAC-IP route from PE2 with a higher sequence number, an ARP probe is triggered at PE1 to clear the stale local MAC-IP neighbor adjacency OR re-learn the local MAC-IP in case the host has moved back or is duplicate.

- Following a local MAC age-out, if there is a local IP adjacency with this MAC, an ARP probe is triggered for this IP to either re-learn the local MAC and maintain local l3 and l2 reachability to this host OR to clear the ARP entry in case the host is indeed no longer local. Note that clearing of stale ARP entries, following a move is required for traffic to converge in the event that the host was silent and not discovered at its new location. Once stale ARP entry for the host is cleared, routed traffic flow destined for the host can re-trigger ARP discovery for this host at the new location. ARP flooding on the overlay MUST also be done to enable ARP discovery via routed flows.

- Alternatively, ARP probing timer may be tuned to be smaller than the MAC aging timer to avoid MAC age-out.

PE-CE control plane learning alternative defined in this document decouples host learning following moves from unpredictable host behavior with respect to sending data traffic and unsolicited ARPs,
and as a result from ARP probing and MAC aging timer settings. Host move handling is hence greatly simplified to a very predictable and deterministic behavior.

13.1.3 ARP Gleaning Latency

If a CE’s ARP binding is not already learned on a PE via an unsolicited ARP sent by the CE following events such as boot-up, flaps, and moves, a data frame that needs to be routed to the CE triggers ARP or ND discovery process on the PE. On a typical hardware switching platform, an IP packet that does not resolve to a link layer re-write would be punted to host stack that delivers packets with incomplete link-layer resolution to ARP or ND for resolution. An ARP request / ND Solicitation is generated for the CE IP and an ARP response or NA results in installing a link-layer re-write for the CE IP. In an EVPN multi-homing environment, this procedure is further complicated as the response is only received by one of the PEs that may or may not be the one that generated the ARP or ND request. Learned neighbor binding is SYNCed to other PEs that share the multi-homed Ethernet Segment. Routed flows can now be forwarded to the host via all PEs. Latency associated with such data frame driven ARP discovery may result in significant initial convergence hit, following triggers that warrant re-gleaning of CE IP to MAC binding.

PE-CE control plane learning alternative defined in this document results in proactive host learning following these scenarios, potentially avoiding a convergence hit on initial data packets.

13.2 Applicability to non-EVPN Use Cases

While the LSoE based host learning procedure described in this document focuses on EVPN-IRB overlay fabric use case, it may also have benefits and applicability in non-EVPN use cases. Applicability of procedures described in this document to non-EVPN use cases is a topic for further study.

14. Summary

PE-CE control plane is proposed as an alternative to data plane and ARP/ND snooping based PE-CE host MAC/IP learning and for PE-CE prefix learning. With a PE-CE control plane, CE host MAC and IP are deterministically learned on host boot-up, on host configuration, across host moves, on convergence triggers such as link failures, flaps, and PE re-boots and on all-active multi-homing LAG links. A PE-CE control plane decouples CE MAC and IP learning from traffic flows sourced by a CE, from varying CE behavior with respect to sending unsolicited ARP/ND frames, and from hashing of CE sourced frames over all-active multi-homed LAG links. As a result, it helps
achieve a predictable and reliable convergence behavior across these triggers and helps simplify certain EVPN procedures that are otherwise needed with a data-plane and ARP/ND snooping based PE-CE learning. In addition, it may also be used for non-host learning use cases such as prefix learning.
15. References

15.1 Normative References


15.2 Informative References
INTERNET DRAFT  LSoE-based PE-CE Control Plane for EVPN

16. Acknowledgements

Authors would like to thank Randy Bush and Rob Austein for detailed review and feedback to ensure consistency with base LSOE protocol specification, as well as for helping build detailed LSOE flows included in this document.

Authors would like to thank Ali Sajassi and John Drake for detailed review and very valuable input on PE-CE protocol design for EVPN use cases as well as structuring this document for EVPN use cases.

Contributors

Randy Bush
Arrcus & IIJ
5147 Crystal Springs
Bainbridge Island, WA  98110
United States of America

Email: randy@psg.com

Authors’ Addresses

Neeraj Malhotra (Editor)
Arrcus
2077 Gateway Place, Suite #400
San Jose, CA  95119, USA

Email: neeraj.ietf@gmail.com

Keyur Patel
Arrcus
2077 Gateway Place, Suite #400
San Jose, CA  95119, USA

Email: keyur@arrcus.com

Jorge Rabadan
Nokia
777 E. Middlefield Road
Mountain View, CA 94043, USA

Email: jorge.rabadan@nokia.com