Shortest Path Routing Extensions for BGP Protocol

draft-ietf-lsvr-bgp-spf-05

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 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 January 23, 2020.

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.

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 unnumber 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-SFP (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.

```
+---+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      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.

```
+---+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   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.

<table>
<thead>
<tr>
<th>TBD Type</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>BGP SPF Status</td>
<td></td>
</tr>
</tbody>
</table>

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 out 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 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.
4. 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 fail 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]


11.2. Information References


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

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 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 April 2, 2020.

Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.
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 .................................... 7
       4.1.1. Node NLRI Attribute SPF Capability TLV ........ 7
       4.1.2. BGP-LS Node NLRI Attribute SPF Status TLV .... 8
   4.2. Link NLRI Usage .................................... 9
       4.2.1. BGP-LS Link NLRI Attribute Prefix-Length TLVs .... 9
       4.2.2. BGP-LS Link NLRI Attribute SPF Status TLV ...... 10
   4.3. Prefix NLRI Usage ................................ 10
       4.3.1. BGP-LS Prefix NLRI Attribute SPF Status TLV .... 11
   4.4. BGP-LS Attribute Sequence-Number TLV ................ 11
5. Decision Process with SPF Algorithm ........................ 12
   5.1. Phase-1 BGP NLRI Selection .......................... 13
   5.2. Dual Stack Support .................................. 14
   5.3. SPF Calculation based on BGP-LS NLRI ................ 14
   5.4. NEXT_HOP Manipulation ................................ 17
   5.5. IPv4/IPv6 Unicast Address Family Interaction ....... 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 an
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 unnumber 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

The BGP Node NLRI will be advertised unconditionally by all routers in the BGP SPF routing domain.

4.1.1. Node NLRI Attribute SPF Capability TLV

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 in the Shortest Path Tree (SPT).

4.1.2. BGP-LS Node NLRI Attribute SPF Status TLV

A BGP-LS Attribute TLV to BGP-LS Node NLRI is defined to indicate the status of the node with respect to the BGP SPF calculation. This will be used to rapidly take a node out of service or to indicate the node is not to be used for transit (i.e., non-local) traffic. If the SPF Status TLV is not included with the Node NLRI, the node is considered to be up and is available for transit traffic.
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.
4.2.2. BGP-LS Link NLRI Attribute 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 SPF Status TLV is not included with the Link NLRI, the link is considered up and available.

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.

4.3. 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.
4.3.1. 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                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| SPF Status    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

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

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.

```
  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              Type             |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                Sequence Number (High-Order 32 Bits)           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                Sequence Number (Low-Order 32 Bits)            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

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 NLRIs 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. If the BGP-LS Node attribute includes an SPF Status TLV (Section 4.1.2) indicating the node is unreachable, the Node NLRI is ignored and the next lowest cost Node NLRI is selected from candidate list. 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 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.

* If the current Node NLRI attributes includes the SPF status TLV (Section 4.1.2) and the status indicates that the Node doesn’t support transit, the next link for the current node is processed.

* 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 attribute includes an 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 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 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 SPF Status TLV Section 4.3.1 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 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 five attribute TLVs for BGP-LS NLRI. We request IANA to assign types for the SPF capability TLV, Sequence Number TLV, IPv4 Link Prefix-Length TLV, IPv6 Link Prefix-Length TLV, and SPF Status TLV 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. Thanks to Chaitanya Yadlapalli and Pushpals Sarkar for discussions on preventing a BGP SPF Router from being used for non-local traffic (i.e., transit traffic).

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

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]

11.2. Information References


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
Layer 3 Discovery and Liveness
draft-ietf-lsvr-l3dl-02

Abstract

In Massive Data Centers, BGP-SPF and similar routing protocols are used to build topology and reachability databases. These protocols need to discover IP Layer 3 attributes of links, such as logical link IP encapsulation abilities, IP neighbor address discovery, and link liveness. This Layer 3 Discovery and Liveness protocol collects these data, which may then be disseminated using BGP-SPF and similar protocols.

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 January 8, 2020.
# Table of Contents

1. Introduction ...................................................... 3  
2. Terminology ...................................................... 4  
3. Background ...................................................... 5  
4. Top Level Overview ............................................... 5  
5. Inter-Link Protocol Overview ................................... 6  
   5.1. L3DL Ladder Diagram ....................................... 7  
6. Transport Layer .................................................. 8  
7. The Checksum .................................................... 10  
8. TLV PDUs ......................................................... 12  
9. Logical Link Endpoint Identifier ............................... 13  
10. HELLO .......................................................... 14  
11. OPEN ........................................................... 15  
12. ACK ............................................................ 17  
   12.1. Retransmission ............................................. 18  
13. The Encapsulations .............................................. 18  
   13.1. The Encapsulation PDU Skeleton ............................ 19  
   13.2. Encapsulation Flags ....................................... 20  
   13.3. IPv4 Encapsulation ....................................... 21  
   13.4. IPv6 Encapsulation ....................................... 21  
   13.5. MPLS Label List .......................................... 22  
   13.6. MPLS IPv4 Encapsulation ................................ 22  
   13.7. MPLS IPv6 Encapsulation ................................ 23  
14. VENDOR - Vendor Extensions .................................... 24  
15. KEEPALIVE - Layer 2 Liveness .................................. 25  
16. Layers 2.5 and 3 Liveness ..................................... 26  
17. The North/South Protocol ...................................... 26  
   17.1. Use BGP-LS as Much as Possible .......................... 26  
   17.2. Extensions to BGP-LS ..................................... 26  
18. Discussion ...................................................... 27  
   18.1. HELLO Discussion .......................................... 27  
   18.2. HELLO versus KEEPALIVE .................................. 27
1. Introduction

The Massive Data Center (MDC) environment presents unusual problems of scale, e.g. O(10,000) forwarding 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 [Clos0][Clos1] and similar environments. But BGP-SPF and similar higher level device-spanning protocols, e.g. [I-D.malhotra-bess-evpn-lsoe], need logical link state and addressing data from the network to build the routing topology. They also need prompt but prudent reaction to (logical) link failure.

Layer 3 Discovery and Liveness (L3DL) provides brutally simple mechanisms for devices to

- Discover each other’s unique endpoint identification,
- Discover mutually supported layer 3 encapsulations, e.g. IP/MPLS,
- Discover Layer 3 IP and/or MPLS addressing of interfaces of the encapsulations,
- 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,
- Enable layer 3 link liveness such as BFD, and finally
- Provide Layer 2 keep-alive messages for session continuity.
This protocol may be more widely applicable to a range of routing and similar protocols which need layer 3 discovery and characterisation.

2. Terminology

Even though it concentrates on the inter-device layer, this document relies heavily on routing terminology. The following attempts to clarify the use of 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 Shortest Path First 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 L3DL content of a single Layer 2 frame. A full L3DL PDU may be packaged in multiple Datagrams.

Encapsulation: Address Family Indicator and Subsequent Address Family Indicator (AFI/SAFI). I.e. classes of layer 2.5 and 3 addresses such as IPv4, IPv6, MPLS, etc.

Frame: A Layer 2 packet.

Link or Logical Link: A logical connection between two logical ports on two devices. E.g. two VLANs between the same two ports are two links.

LLEI: Logical Link Endpoint Identifier, the unique identifier of one end of a logical link, see Section 9.

MAC Address: 48-bit Layer 2 addresses are assumed since they are used by all widely deployed Layer 2 network technologies of interest, especially Ethernet. See [IEEE.802_2001].

MDC: Massive Data Center, commonly composed of thousands of Top of Rack Switches (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 L3DL 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 [RFC6286].

Session: An established, via OPEN PDUs, session between two L3DL capable link end-points.

SPF: Shortest Path First, an algorithm for finding the shortest paths between nodes in a graph; AKA Dijkstra’s algorithm.
System Identifier: An eight octet ISO System Identifier a la [RFC1629] System ID

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

L3DL is primarily designed for a Clos type datacenter scale and topology, but can accommodate richer topologies which contain potential cycles.

While L3DL is designed for the MDC, there are no inherent reasons it could not run on a WAN. The authentication and authorization needed to run safely on a WAN need to be considered, and the appropriate level of security options chosen.

L3DL assumes a new IEEE assigned EtherType (TBD).

The number of addresses of one Encapsulation type on an interface link may be quite large given a TOR with tens of servers, each server having a few hundred micro-services, resulting in an inordinate number of addresses. And highly automated micro-service migration can cause serious address prefix disaggregation, resulting in interfaces with thousands of disaggregated prefixes.

Therefore the L3DL protocol is session oriented and uses incremental announcement and withdrawal with session restart, a la BGP ([RFC4271]).

4. Top Level Overview

- Devices discover each other on logical links
- Logical Link Endpoint Identifiers are exchanged
- Layer 2 Liveness Checks may be started
- Encapsulation data are exchanged and IP-Level Liveness Checks enabled
- A BGP-like upper layer protocol is assumed to use these data to discover and build a topology database
There are two protocols, the inter-device per-link layer 3 discovery and the API to the upper level BGP-like routing protocol:

- Inter-device PDUs are used to exchange device and logical link identities and layer 2.5 and 3 identifiers (not payloads), e.g. device IDs, port identities, VLAN IDs, Encapsulations, and IP 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 L3DL protocols.

To simplify this document, Layer 2 framing is not shown. L3DL is about layer 3.

5. Inter-Link Protocol Overview

Two devices discover each other and their respective identities by sending multicast HELLO PDUs (Section 10). To assure discovery of new devices coming up on a multi-link topology, devices on such a topology send periodic HELLOs forever, see Section 18.1.
Once a new device is recognized, both devices attempt to negotiate and establish a session by sending unicast OPEN PDUs (Section 11). In an established session, the Encapsulations (Section 13) configured on an end point may be announced and modified. Note that these are only the encapsulation and addresses configured on the announcing interface; though a device’s loopback and overlay interface(s) may also be announced. 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. L3DL Ladder Diagram

The HELLO, Section 10, is a priming message. It is a small L3DL PDU encapsulated in an Ethernet multicast frame with the simple goal of discovering the identities of logical link endpoint(s) reachable from a Logical Link Endpoint, Section 9.

The HELLO and OPEN, Section 11, PDUs, which are used to discover and exchange detailed Logical Link Endpoint Identifiers, LLEIs, and the ACK/ERROR PDU, are mandatory; other PDUs are optional; though at least one encapsulation SHOULD be agreed at some point.

The following is a ladder-style diagram of the L3DL protocol exchanges:

```
<table>
<thead>
<tr>
<th>HELLO</th>
<th>Logical Link Peer discovery</th>
</tr>
</thead>
<tbody>
<tr>
<td>HELLO</td>
<td>Mandatory</td>
</tr>
<tr>
<td>OPEN</td>
<td>MACs, IDs, etc.</td>
</tr>
<tr>
<td>ACK</td>
<td>Mandatory</td>
</tr>
<tr>
<td>OPEN</td>
<td>Interface IPv4 Addresses</td>
</tr>
<tr>
<td>ACK</td>
<td>Interface IPv4 Addresses</td>
</tr>
<tr>
<td>Interface IPv4 Addresses</td>
<td>Optional</td>
</tr>
</tbody>
</table>
```
<table>
<thead>
<tr>
<th>ACK</th>
<th>Interface IPv6 Addresses</th>
<th>Interface IPv6 Addresses</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ACK</td>
<td>ACK</td>
</tr>
<tr>
<td></td>
<td>Interface IPv6 Addresses</td>
<td>Interface IPv6 Addresses</td>
</tr>
<tr>
<td></td>
<td>ACK</td>
<td>ACK</td>
</tr>
<tr>
<td></td>
<td>Interface MPLSv4 Labels</td>
<td>Interface MPLSv4 Labels</td>
</tr>
<tr>
<td></td>
<td>ACK</td>
<td>ACK</td>
</tr>
<tr>
<td></td>
<td>Interface MPLSv4 Labels</td>
<td>Interface MPLSv4 Labels</td>
</tr>
<tr>
<td></td>
<td>ACK</td>
<td>ACK</td>
</tr>
<tr>
<td></td>
<td>Interface MPLSv6 Labels</td>
<td>Interface MPLSv6 Labels</td>
</tr>
<tr>
<td></td>
<td>ACK</td>
<td>ACK</td>
</tr>
<tr>
<td></td>
<td>Interface MPLSv6 Labels</td>
<td>Interface MPLSv6 Labels</td>
</tr>
<tr>
<td></td>
<td>ACK</td>
<td>ACK</td>
</tr>
<tr>
<td></td>
<td>L3DL KEEPALIVE</td>
<td>Layer 2 Liveness</td>
</tr>
<tr>
<td></td>
<td>L3DL KEEPALIVE</td>
<td>L3DL KEEPALIVE</td>
</tr>
</tbody>
</table>

6. Transport Layer

L3DL PDUs are carried by a simple transport layer which allows PDUs to occupy many Ethernet frames. An L3DL Ethernet frame is referred to as a Datagram.
The L3DL Transport Layer encapsulates each Datagram using a common transport header.

If a PDU does not fit in a single datagram, it is broken into multiple datagrams and reassembled by the receiver a la [RFC0791] Section 2.3 Fragmentation.

```
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 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Version   |L|                Datagram Number                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        Datagram Length        |            Checksum           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
˜                               |           Payload...          ˜
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
˜                               |           Payload...          ˜
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The fields of the L3DL Transport Header are as follows:

Version: Seven-bit Version number of the protocol, currently 0. Values other than 0 MUST BE treated as an error. The protocol version nees to be in one and only one place, so it is in the datagram as opposed to, for example, the PDU header.

L: A bit that set to one if this Datagram is the last Datagram of the PDU. For a PDU which fits in only one Datagram, it is set to one. Note that this is the inverse of the marking technique used by [RFC0791].

Datagram Number: A monotonically increasing 24-bit value which starts at zero for each PDU. This is used to reassemble frames into PDUs a la [RFC0791] Section 2.3. Note that this limits an L3DL PDU to 2^24 frames.

Datagram Length: Total number of octets in the Datagram including all payloads and fields. Note that this limits a datagram to 2^16 octets.

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

Payload: The PDU being transported or a fragment thereof.

To avoid the need for a receiver to reassemble two PDUs at the same time, a sender MUST NOT send a subsequent PDU when a PDU is already in flight and not yet acknowledged if it is an ACKed PDU Type.
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 prudent 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>
#include <stdint.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, 0x4f, 0xb3, 0x21, 0x15, 0x78,
  0x99, 0xb1, 0xaf, 0xf9, 0xe7, 0x2d, 0x4d, 0x8a, 0xce, 0x4c, 0xca, 0xe2,
  0x52, 0x95, 0xd9, 0x1e, 0x4e, 0x38, 0x44, 0x28, 0x0a, 0xdf, 0x02, 0xa0,
  0x17, 0xf1, 0x60, 0x68, 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,
  0xee, 0xb4, 0x1a, 0xea, 0xd0, 0x91, 0x2f, 0xb8, 0x55, 0xb9, 0xda, 0x85,
  0x3f, 0xc1, 0xbf, 0xe0, 0x5a, 0x58, 0x80, 0x5f, 0x66, 0xb0, 0xda, 0x80,
  0x35, 0xd5, 0xc0, 0xa7, 0x33, 0x06, 0x65, 0x69, 0x45, 0x00, 0x94, 0x56,
  0x8d, 0x98, 0xb9, 0x76, 0x97, 0xfc, 0xb2, 0xc2, 0xb0, 0xe0, 0xdb, 0x20,
  0xe1, 0x8b, 0xd6, 0xe4, 0xd4, 0x47, 0x4a, 0x1d, 0x42, 0xed, 0xe9, 0xe6,
  0x49, 0x3c, 0xc4, 0x32, 0x70, 0xd2, 0x07, 0xd4, 0xce, 0x77, 0x67, 0x18,
  0x89, 0xcb, 0x30, 0x1f, 0x84, 0xc6, 0x8f, 0x6a, 0xc8, 0x74, 0xdc, 0xc9,
  0x5d, 0x9c, 0x31, 0xa4, 0x70, 0x88, 0x61, 0x2c, 0x9f, 0x0d, 0x2b, 0x87,
  0x50, 0x82, 0x54, 0x64, 0xc6, 0x7d, 0x20, 0x40, 0x34, 0x4b, 0xc5, 0x73,
  0xda, 0xc4, 0xf0, 0x3b, 0xc0, 0xf0, 0x7f, 0xb8, 0xe6, 0x3e, 0x5b, 0xa5,
  0xda, 0x04, 0x23, 0x9c, 0x14, 0x51, 0x22, 0x0f, 0x29, 0x79, 0x71, 0x7e,
  0xff, 0x8c, 0xe0, 0x2c, 0xe0, 0xc0, 0xe0, 0xc8, 0x72, 0x75, 0x6f, 0x37, 0xa1,
  0xe0, 0xd3, 0x8e, 0x62, 0x8b, 0x86, 0x10, 0xe8, 0x08, 0x77, 0x11, 0xbe,
  0x92, 0x4f, 0x24, 0xc5, 0x32, 0x36, 0x9d, 0xcf, 0xf3, 0xa6, 0xbb, 0xac,
  0x5e, 0x6c, 0xa9, 0x13, 0x57, 0x25, 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 L3DL application layer PDU is a typical TLV (Type Length Value) PDU. It includes a signature to provide optional integrity and authentication. 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    PDU Type   |                 Payload Length                |               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  Payload ...                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

                |                  Payload ...                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

                |                  Payload ...                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

|    Sig Type   |        Signature Length       |               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|               |                        Signature |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

The fields of the basic L3DL header are as follows:

PDU Type: An integer differentiating PDU payload types. See Section 22.1.

Payload Length: Total number of octets in the Payload field.

Payload: The application layer content of the L3DL PDU.

Sig Type: The type of the Signature, see Section 22.2. Type 0, a null signature, is defined in this document.

Sig Type 0 indicates a null Signature. For a trivial PDU such as KEEPALIVE, the underlying Datagram checksum may be sufficient for integrity, though it lacks authentication.

Other Sig Types may be defined in other documents.

Signature Length: The length of the Signature, possibly including padding, in octets. If Sig Type is 0, Signature Length MUST BE 0.

Signature: The result of running the signature algorithm specified in Sig Type over all octets of the PDU except for the Signature itself.
9. Logical Link Endpoint Identifier

L3DL discovers neighbors on logical links and establishes sessions between the two ends of all consenting discovered logical links. A logical link is described by a pair of Logical Link Endpoint Identifiers, LLEIs.

An LLEI is a variable length descriptor which could be an ASN, a classic RouterID, a catenation of the two, an eight octet ISO System Identifier [RFC1629], or any other identifier unique to a single logical link endpoint in the topology.

An L3DL deployment will choose and define an LLEI which suits its needs, simple or complex. Two extremes are as follows:

A simplistic view of a link between two devices is two ports, identified by unique MAC addresses, carrying a layer 3 protocol conversation. In this case, the MAC addresses might suffice for the LLEIs.

Unfortunately, things can get more complex. Multiple VLANs can run between those two MAC addresses. In practice, many real devices use the same MAC address on multiple ports and/or sub-interfaces.

Therefore, in the general circumstance, a fully described LLEI might be as follows:

```
  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----------------------------------------------+
|                                               |
| System Identifier                             |
|                                               |
|-----------------------------------------------+
| ifIndex                                       |
|                                               |
+------------------------------------------------
```

System Identifier, a la [RFC1629], is an eight octet identifier unique in the entire operational space. Routers and switches usually have internal MAC Addresses which can be padded with high order zeros and used if no System ID exists on the device. If no unique identifier is burned into a device, the local L3DL configuration SHOULD create and assign a unique one by configuration.

ifIndex is the SNMP identifier of the (sub-)interface, see [RFC1213]. This uniquely identifies the port.
For a layer 3 tagged sub-interface or a VLAN/SVI interface, Ifindex is that of the logical sub-interface, so no further disambiguation is needed.

L3DL 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.

LLEIs are big-endian.

10. HELLO

The HELLO PDU is unique in that it is encapsulated in a multicast Ethernet frame. It solicits response(s) from other LLEI(s) on the link. See Section 18.1 for why multicast is used. The destination multicast MAC Addresses 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)). This SHOULD BE used when the link is known to be a simple point to point link.

To Be Assigned: When a switch receives a frame with a multicast destination MAC it does not recognize, it forwards to all ports. This destination MAC is to be sent when the interface is known to be connected to a switch. See Section 23. This SHOULD BE used when the link may be a multi-point link.

All other L3DL PDUs are encapsulated in unicast frames, as the peer’s destination MAC address is known after the HELLO exchange.

When an interface is turned up on a device, it SHOULD issue a HELLO if it is to participate in L3DL sessions.

If a constrained Nearest Bridge destination address is configured for a point-to-point interface, see above, then the HELLO SHOULD NOT be repeated once a session has been created by an exchange of OPENs.

If the configured destination address is one that is propagated by switches, the HELLO SHOULD be repeated at a configured interval, with a default of 60 seconds. This allows discovery by new devices which come up on the layer-2 mesh.
If more than one device responds, one adjacency is formed for each unique source LLEI response. L3DL treats each adjacency as a separate logical link.

When a HELLO is received from a source MAC address with which there is no established L3DL session, the receiver SHOULD respond with an OPEN PDU. The two devices establish an L3DL session by exchanging OPEN PDUs.

The Payload Length is zero as there is no payload.

HELLO PDUs can not be signed as keying material has yet to be exchanged. Hence the signature MUST always be the null type.

11. OPEN

Each device has learned the other’s MAC Address from the HELLO exchange, see Section 10. Therefore the OPEN and subsequent PDUs MUST BE unicast, as opposed to the HELLO’s multicast frame.
The Payload Length is the number of octets in all fields of the PDU from the Nonce through the Serial Number, not including the three final signature fields.

The Nonce enables detection of a duplicate OPEN PDU. It SHOULD be either a random number or a high resolution timestamp. It is needed to prevent session closure due to a repeated OPEN caused by a race or a dropped or delayed ACK.

My LLEI is the sender’s LLEI, see Section 9.

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

A node may have zero or more operator-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 Type is the Signature algorithm suite, see Section 8.

Key Length is a 16-bit field denoting the length in octets of the Key itself, not including the Auth Type or the Key Length. If there is no Key, the Auth Type and key Length MUST both be zero.

The Key is specific to the operational environment. A failure to authenticate is a failure to start the L3DL session, an ERROR PDU MUST BE sent (Error Code 2), and HELLOs MUST be restarted.

The Serial Number is that of the last received and processed PDU. This allows a receiver sending an OPEN to tell the sender that the receiver wants to resume a session and the sender only needs to send data more recent than the Serial Number. If this OPEN is not trying to restart a lost session, the Serial Number MUST BE set to zero.

The Signature fields are described in Section 8 and in an asymmetric key environment serve as a proof of possession of the signing auth data by the sender.

Once two logical link endpoints know each other, and have ACKed each other’s OPEN PDUs, Layer 2 KEEPALIVEs (see Section 15) MAY be started to ensure Layer 2 liveness and keep the session semantics alive. The timing and acceptable drop of KEEPALIVE PDUs are discussed in Section 15.
If a sender of OPEN does not receive an ACK of the OPEN PDU, 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 an LLEI with which the receiving logical link endpoint believes it already has an L3DL session (OPENs have already been exchanged), and the Serial Number in the OPEN is non-zero, the receiver SHOULD establish a new session by sending an OPEN with the Serial Number of the last data it received. Each party MUST resume sending encapsulations etc. subsequent to the other party’s Sequence Number. And each MUST retain all previously discovered encapsulation and other data.

If a properly authenticated OPEN arrives with a new Nonce from an LLEI with which the receiving logical link endpoint believes it already has an L3DL session (OPENs have already been exchanged), and the Serial Number in the OPEN is zero, then the receiver MUST assume that the sending LLEI or entire device has been reset. All previously discovered encapsulation data MUST NOT be kept and MUST be withdrawn via the BGP-LS API and the recipient MUST respond with a new OPEN.

12. ACK

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

```
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
|  PDU Type = 3 |               Payload Length = 5              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
| ACKed PDU | EType |       Error Code      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
|           Error Hint          |    Sig Type   |Signature Leng.|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
|                 Signature ...                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
```

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

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

The Error Codes, noting protocol failures listed in this document, are listed in Section 22.4. Someone stuck in the 1990s might think the catenation of EType and Error Code as an echo of 0x1zzz, 0x2zzz, etc. They might be right; or not.

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.

The Signature fields are described in Section 8.

12.1. 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), and the interface is live at layer 2, 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 MAY BE considered closed in case of this ACK failure.

If the link is broken at layer 2, retransmission MAY BE retried when the link is restored.

13. The Encapsulations

Once the devices know each other’s LLEIs, know each other’s upper layer identities, have means to ensure link state, etc., the L3DL session is considered established, and the devices SHOULD exchange L3 interface encapsulations, L3 addresses, and L2.5 labels.
The Encapsulation types the peers exchange may be IPv4 (Section 13.3), IPv6 (Section 13.4), MPLS IPv4 (Section 13.6), MPLS IPv6 (Section 13.7), 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 12) 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) 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 logical 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 IP subnet.

13.1. The Encapsulation PDU Skeleton

The header for all encapsulation PDUs is as follows:

```
+---------------------------------------------+---------------------------------------------+
|    PDU Type   |                 Payload Length                |
+---------------------------------------------+---------------------------------------------+
|                          Count              |
+---------------------------------------------+---------------------------------------------+
|                         Serial Number         |
+---------------------------------------------+---------------------------------------------+
|                     Encapsulation List...    |    Sig Type   |
+---------------------------------------------+---------------------------------------------+
|       Signature Length       |         Signature ...         |
+---------------------------------------------+---------------------------------------------+
```

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

The 24-bit Count is the number of Encapsulations in the Encapsulation list.

The Serial Number is a monotonically increasing 32-bit value representing the sender’s state in time. It may be an integer, a
timestamp, etc. On session restart (new OPEN), a receiver MAY send
the last received Session Number to tell the sender to only send
newer data.

If a sender has multiple links on the same interface, separate state:
data, ACKs, etc. must be kept for each peer session.

Over time, multiple Encapsulation PDUs may be sent for an interface
as configuration changes.

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 Signature fields are described in Section 8.

The Receiver MUST acknowledge the Encapsulation PDU with a Type=3,
ACK PDU (Section 12) with the Encapsulation Type being that of the
encapsulation being announced, see Section 12.

If the Sender does not receive an ACK in a configurable interval
(default one second), and the interface is live at layer 2, they
SHOULD retransmit. After a user configurable number of failures, the
L3DL session should be considered dead and the OPEN process SHOULD be
restarted.

If the link is broken at layer 2, retransmission MAY BE retried if
data have not changed in the interim.

13.2.  Encapsulation Flags

```
    0  1  2  3  4  ...  7
   +---------------+---------------+---------------+---------------+---------------+
   | Ann/With      | Primary       | Under/Over    | Loopback      | Reserved ..   |
   +---------------+---------------+---------------+---------------+---------------+
```

Each encapsulation in an Encapsulation PDU of Type T may announce new
and/or withdraw old encapsulations of Type T. It indicates this with
the Ann/With Encapsulation Flag, Announce == 1, Withdraw == 0.

Each Encapsulation interface address in an Encapsulation PDU is
either a new encapsulation be announced (Ann/With == 1) (yes, a la
BGP) or requests one be withdrawn (Ann/With == 0). Adding an
encapsulation which already exists SHOULD raise an Announce/Withdraw
Error (see Section 22.4); the EType SHOULD be 2, suggesting a session
restart (see Section 12 so all encapsulations will be resent.

If an LLEI has multiple addresses for an encapsulation type, one and
only one address SHOULD be configured to be marked as primary
(Primary Flag == 1). Only one address on an interface MAY be marked as primary for a particular encapsulation type.

An Encapsulation interface address in an Encapsulation PDU MAY be marked as a loopback, in which case the Loopback bit is set. 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 L3DL speaking external interfaces, e.g. for iBGP peering. They SHOULD be marked as such, Loopback Flag == 1.

Each Encapsulation interface address in an Encapsulation PDU is that of the direct ‘underlay interface (Under/Over == 1), or an ‘overlay’ address (Under/Over == 0), likely that of a VM or container guest bridged or configured on to the interface already having an underlay address.

13.3. 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
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  PDU Type = 4 |                 Payload Length                ˜
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
˜               |                     Count                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Serial Number                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Encaps Flags  |                  IPv4 Address                 ˜
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
˜               |   PrefixLen   |    more ...   |    Sig Type   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        Signature Length       |         Signature ...         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The 24-bit Count is the number of IPv4 Encapsulations being announced and/or withdrawn.

13.4. IPv6 Encapsulation

The IPv6 Encapsulation describes a logical link’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.
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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  PDU Type = 5 |                 Payload Length                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
          |                     Count                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
          |                         Serial Number                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
          | Encaps Flags |                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
          |                        IPv6 Address                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
          |               |   PrefixLen   |    more ...   |    Sig Type   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
          |        Signature Length       |         Signature ...         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

The 24-bit Count is the number of IPv6 Encapsulations being announced and/or withdrawn.

13.5.  MPLS Label List

As an MPLS enabled interface may have a label stack, see [RFC3032], a variable length list of labels is needed. These are the labels the sender will accept for the prefix to which the list is attached.

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Label Count   |                 Label                 | Exp |S|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
          |                 Label                 | Exp |S|    more ...   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
          |                        Signature Length                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
          |                        Signature ...                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

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

13.6.  MPLS IPv4 Encapsulation

The MPLS IPv4 Encapsulation describes a logical link’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 which will be accepted for each address.
The 24-bit Count is the number of MPLSv4 Encapsulation being announced and/or withdrawns.

13.7. MPLS IPv6 Encapsulation

The MPLS IPv6 Encapsulation describes a logical link’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 which will be accepted for each address.
The 24-bit Count is the number of MPLSv6 Encapsulations being announced and/or withdrawn.

14. VENDOR - Vendor Extensions

Vendors or enterprises may define TLVs beyond the scope of L3DL standards. This is done using a Private Enterprise Number [IANA-PEN]
followed by Enterprise Data in a format defined for that Enterprise Number and Ent Type.

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. KEEPALIVE - Layer 2 Liveness

L3DL devices SHOULD beacon frequent Layer 2 KEEPALIVE PDUs to ensure session continuity. A receiver may choose to ignore KEEPALIVE PDUs.

An operational deployment MUST BE configured whether to use KEEPALIVEs or not, either globally, or down to per-link granularity. Disagreement MAY result in repeated session break and reestablishment.

KEE PALIVEs SHOULD be beaconed at a configured frequency. One per second is the default. Layer 3 liveness, such as BFD, may be more (or less) aggressive.

When a sender transmits a PDU which is not a KEEPALIVE, the sender SHOULD reset the KEEPALIVE timer. I.e. sending any PDU acts as a keepalive. Once the last fragment has been sent, the KEEPALIVE timer SHOULD BE restarted. Do not wait for the ACK.

If a KEEPALIVE or other PDUs have not been received from a peer with which a receiver has an open session for a configurable time (default 30 seconds), the link SHOULD BE presumed down. The devices MAY keep configuration state and restore it without retransmission if no data have changed. Otherwise, a new session SHOULD BE established and new Encapsulation PDUs exchanged.
16. Layers 2.5 and 3 Liveness

Layer 2 liveness may be continuously tested by KEEPALIVE PDUs, see Section 15. As layer 2.5 or layer 3 connectivity could still break, liveness above layer 2 MAY be frequently tested using BFD ([RFC5880]) or a similar technique.

This protocol assumes that one or more Encapsulation addresses may be used to ping, run BFD, or whatever the operator configures.

17. The North/South Protocol

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

The devices know their unique LLEIs and know the unique peer LLEIs and Encapsulations on each logical link interface.

Full topology discovery is not appropriate at the L3DL layer, so Dijkstra a la IS-IS etc. is assumed to be done by higher level protocols such as BGP-SPF.

Therefore the LLEIs, 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.

17.1. Use BGP-LS as Much as Possible

BGP-LS [RFC7752] defines BGP-like Datagrams describing logical 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.

17.2. 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-bgpls-segment-routing-epe]. The local and remote node descriptors for all NLRI are the IDs described in Section 11. 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.

18. Discussion

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

18.1. HELLO Discussion

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 logical interface (LLEI), I, on an L3DL 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, on multi-link interfaces MUST continue to send HELLOs as long as they are turned up.

18.2. HELLO versus KEEPALIVE

Both HELLO and KEEPALIVE are periodic. KEEPALIVE might be eliminated in favor of keeping only HELLOs. But KEEPALIVEs are unicast, and thus less noisy on the network, especially if HELLO is configured to transit layer-2-only switches, see Section 18.1.

19. VLANs/SVIs/Sub-interfaces

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

As the upper routing layer must view VLAN topologies as separate graphs, L3DL treats VLAN ports as separate links.

L3DL 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 LLIEs, they act as separate interfaces, forming their own links.
20. Implementation Considerations

An implementation SHOULD provide the ability to configure a logical interface as L3DL speaking or not.

An implementation SHOULD provide the ability to configure whether HELLOs on an L3DL enabled interface send Nearest Bridge or the MAC which is propagated by switches from that interface; see Section 10.

An implementation SHOULD provide the ability to distribute one or more loopback addresses or interfaces into L3DL on an external L3DL speaking interface.

An implementation SHOULD provide the ability to distribute one or more overlay and/or underlay addresses or interfaces into L3DL on an external L3DL speaking interface.

An implementation SHOULD provide the ability to configure one of the addresses of an encapsulation as primary on an L3DL speaking interface. If there is only one address for a particular encapsulation, the implementation MAY mark it as primary by default.

An implementation MAY allow optional configuration which updates the local forwarding table with overlay and underlay data both learned from L3DL peers and configured locally.

21. Security Considerations

The protocol as is MUST NOT be used outside a datacenter or similarly closed environment due to lack of formal definition of the authentication and authorization mechanism. Sufficient mechanisms may be described in separate documents.

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. In the case of L3DL, Authentication and Integrity as provided in [draft-ymbk-l3dl-signing] is strongly recommended.

It is generally unwise to assume that on the wire Layer 2 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, form malicious sessions, etc. to divert, intercept, or drop traffic.

Similarly, malicious nodes/devices could mis-announce addressing.
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.

If the KEEPALIVE PDU is not signed (as suggested in Section 8) to save computation, then a MITM could fake a session being alive.

22. IANA Considerations

22.1. PDU Types

This document requests the IANA create a registry for L3DL PDU Type, which may range from 0 to 255. The name of the registry should be L3DL-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 Announcement</td>
</tr>
<tr>
<td>5</td>
<td>IPv6 Announcement</td>
</tr>
<tr>
<td>6</td>
<td>MPLS IPv4 Announcement</td>
</tr>
<tr>
<td>7</td>
<td>MPLS IPv6 Announcement</td>
</tr>
<tr>
<td>8-254</td>
<td>Reserved</td>
</tr>
<tr>
<td>255</td>
<td>VENDOR</td>
</tr>
</tbody>
</table>

22.2. Signature Type

This document requests the IANA create a registry for L3DL Signature Type, AKA Sig Type, which may range from 0 to 255. The name of the registry should be L3DL-Signature-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>Number</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Null</td>
</tr>
<tr>
<td>1-255</td>
<td>Reserved</td>
</tr>
</tbody>
</table>
22.3. Flag Bits

This document requests the IANA create a registry for L3DL PL Flag Bits, which may range from 0 to 7. The name of the registry should be L3DL-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>Announce/Withdraw (ann == 0)</td>
</tr>
<tr>
<td>1</td>
<td>Primary</td>
</tr>
<tr>
<td>2</td>
<td>Underlay/Overlay (under == 0)</td>
</tr>
<tr>
<td>3</td>
<td>Loopback</td>
</tr>
<tr>
<td>4-7</td>
<td>Reserved</td>
</tr>
</tbody>
</table>

22.4. Error Codes

This document requests the IANA create a registry for L3DL Error Codes, a 16 bit integer. The name of the registry should be L3DL-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>No Error</td>
</tr>
<tr>
<td>1</td>
<td>Logical Link Addressing Conflict</td>
</tr>
<tr>
<td>2</td>
<td>Authorization Failure in OPEN</td>
</tr>
<tr>
<td>3</td>
<td>Signature Failure in PDU</td>
</tr>
<tr>
<td>4</td>
<td>Announce/Withdraw Error</td>
</tr>
</tbody>
</table>

23. IEEE Considerations

This document requires a new EtherType.

This document requires a new multicast MAC address that will be broadcast through a switch.

24. Acknowledgments

The authors thank Cristel Pelsser for multiple reviews, Harsha Kovuru for comments during implementation, 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,
25.  References

25.1.  Normative References

[I-D.ietf-idr-bgp-ls-segment-routing-ext]
Previdi, S., Talaulikar, K., Filsfils, C., Gredler, H.,
and M. Chen, "BGP Link-State extensions for Segment
Routing", draft-ietf-idr-bgp-ls-segment-routing-ext-16
(work in progress), June 2019.

[I-D.ietf-idr-bgpls-segment-routing-epe]
Previdi, S., Talaulikar, K., Filsfils, C., Patel, K., Ray,
S., and J. Dong, "BGP-LS extensions for Segment Routing
BGP Egress Peer Engineering", draft-ietf-idr-bgpls-

[I-D.ietf-lsvr-bgp-spf]
Patel, K., Lindem, A., Zandi, S., and W. Henderickx,
"Shortest Path Routing Extensions for BGP Protocol",
draft-ietf-lsvr-bgp-spf-04 (work in progress), December
2018.

[IANA-PEN]
"IANA Private Enterprise Numbers",

[IEEE.802_2001]
IEEE, "IEEE Standard for Local and Metropolitan Area
Networks: Overview and Architecture", IEEE 802-2001,
DOI 10.1109/ieeestd.2002.93395, July 2002,

[IEEE802-2014]
Institute of Electrical and Electronics Engineers, "Local
and Metropolitan Area Networks: Overview and

for Network Management of TCP/IP-based internets: MIB-II",
STD 17, RFC 1213, DOI 10.17487/RFC1213, March 1991,
25.2. Informative References


Authors' Addresses

Randy Bush
Arrcus & Internet Initiative Japan
5147 Crystal Springs
Bainbridge Island, WA  98110
US

Email: randy@psg.com

Rob Austein
Arrcus, Inc

Email: sra@hactrn.net

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

Email: keyur@arrcus.com
Abstract

In Massive Data Centers, BGP-SPF and similar routing protocols are used to build topology and reachability databases. These protocols need to discover IP Layer 3 attributes of links, such as logical link IP encapsulation abilities, IP neighbor address discovery, and link liveness. This Layer 3 Discovery and Liveness protocol collects these data, which may then be disseminated using BGP-SPF and similar protocols.

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 May 5, 2020.
Internet-Draft       Layer 3 Discovery and Liveness        November 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 ................................................ 5
4. Top Level Overview ....................................... 5
5. Inter-Link Protocol Overview ............................... 7
   5.1. L3DL Ladder Diagram ................................. 7
6. Transport Layer ........................................... 9
7. The Checksum ............................................. 10
8. TLV PDUs .................................................. 12
9. Logical Link Endpoint Identifier .......................... 13
10. HELLO .................................................... 14
11. OPEN ....................................................... 15
12. ACK ........................................................ 18
   12.1. Retransmission ...................................... 19
13. The Encapsulations ........................................ 19
   13.1. The Encapsulation PDU Skeleton ....................... 20
   13.2. Encapsulation Flags .................................. 21
   13.3. IPv4 Encapsulation .................................. 21
   13.4. IPv6 Encapsulation .................................. 22
   13.5. MPLS Label List ..................................... 23
   13.6. MPLS IPv4 Encapsulation .............................. 23
   13.7. MPLS IPv6 Encapsulation .............................. 24
14. VENDOR - Vendor Extensions ................................. 24
15. KEEPALIVE - Layer 2 Liveness ............................... 25
16. Layers 2.5 and 3 Liveness ................................ 26
17. The North/South Protocol .................................. 26
   17.1. Use BGP-LS as Much as Possible ...................... 27
   17.2. Extensions to BGP-LS ................................. 27
18. Discussion ............................................... 27
   18.1. HELLO Discussion .................................... 27
   18.2. HELLO versus KEEPALIVE .............................. 28
1. Introduction

The Massive Data Center (MDC) environment presents unusual problems of scale, e.g. O(10,000) forwarding 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 [Clos0] [Clos1] and similar environments. But BGP-SPF and similar higher level device-spanning protocols, e.g. [I-D.malhotra-bess-evpn-lsoe], need logical link state and addressing data from the network to build the routing topology. They also need prompt but prudent reaction to (logical) link failure.

Layer 3 Discovery and Liveness (L3DL) provides brutally simple mechanisms for devices to

- Discover each other’s unique endpoint identification,
- Discover mutually supported layer 3 encapsulations, e.g. IP/MPLS,
- Discover Layer 3 IP and/or MPLS addressing of interfaces of the encapsulations,
- 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,
- Enable Layer 3 link liveness such as BFD,
- Provide Layer 2 keep-alive messages for session continuity, and finally
o Provide for authenticity verification of protocol messages.

This protocol may be more widely applicable to a range of routing and similar protocols which need layer 3 discovery and characterisation.

2. Terminology

Even though it concentrates on the inter-device layer, this document relies heavily on routing terminology. The following attempts to clarify the use of 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 Shortest Path First 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 L3DL content of a single Layer 2 frame, sans Ethernet framing. A full L3DL PDU may be packaged in multiple Datagrams.

Encapsulation: Address Family Indicator and Subsequent Address Family Indicator (AFI/SAFI). I.e. classes of layer 2.5 and 3 addresses such as IPv4, IPv6, MPLS, etc.

Frame: A Layer 2 Ethernet packet.

Link or Logical Link: A logical connection between two logical ports on two devices. E.g. two VLANs between the same two ports are two links.

LLEI: Logical Link Endpoint Identifier, the unique identifier of one end of a logical link, see Section 9.

MAC Address: 48-bit Layer 2 addresses are assumed since they are used by all widely deployed Layer 2 network technologies of interest, especially Ethernet. See [IEEE.802_2001].

MDC: Massive Data Center, commonly composed of thousands of Top of Rack Switches (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 L3DL application layer message. A PDU’s content 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 [RFC6286].
Session: An established, via OPEN PDUs, session between two L3DL capable link end-points,

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

System Identifier: An eight octet ISO System Identifier a la [RFC1629] System ID

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

L3DL is primarily designed for a Clos type datacenter scale and topology, but can accommodate richer topologies which contain potential cycles.

While L3DL is designed for the MDC, there are no inherent reasons it could not run on a WAN. The authentication and authorization needed to run safely on a WAN need to be considered, and the appropriate level of security options chosen.

L3DL assumes a new IEEE assigned EtherType (TBD).

The number of addresses of one Encapsulation type on an interface link may be quite large given a TOR with tens of servers, each server having a few hundred micro-services, resulting in an inordinate number of addresses. And highly automated micro-service migration can cause serious address prefix disaggregation, resulting in interfaces with thousands of disaggregated prefixes.

Therefore the L3DL protocol is session oriented and uses incremental announcement and withdrawal with session restart, a la BGP ([RFC4271]).

4. Top Level Overview

- Devices discover each other on logical links
- Logical Link Endpoint Identifiers (LLEIs) are exchanged
- Layer 2 Liveness checks may be started
- Encapsulation data are exchanged and IP-Level Liveness checks enabled
A BGP-like upper layer protocol is assumed to use the identifiers and encapsulation data to discover and build a topology database.

There are two protocols, the inter-device (left-right in the diagram) per-link layer 3 discovery and the API to the upper level BGP-like routing protocol (up-down in the above diagram):

- Inter-device PDUs are used to exchange device and logical link identities and layer 2.5 (MPLS) and 3 identifiers (not payloads), e.g. device IDs, port identities, VLAN IDs, Encapsulations, and IP 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 L3DL protocols.

To simplify this document, Layer 2 framing is not shown. L3DL is about layer 3.
5. Inter-Link Protocol Overview

Two devices discover each other and their respective identities by sending multicast HELLO PDUs (Section 10). To assure discovery of new devices coming up on a multi-link topology, devices on such a topology, and only on a multi-link topology, send periodic HELLOs forever, see Section 18.1.

Once a new device is recognized, both devices attempt to negotiate and establish a session by sending unicast OPEN PDUs (Section 11) to the source MAC addresses (plus VIDs if VLANs) of the received HELLOs. Once a session is established through the OPEN exchange, the Encapsulations (Section 13) configured on an end point may be announced and modified. Note that these are only the encapsulation and addresses configured on the announcing interface; though a device’s loopback and overlay interface(s) may also be announced. 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. L3DL Ladder Diagram

The HELLO, Section 10, is a priming message sent on all configured logical links. It is a small L3DL PDU encapsulated in an Ethernet multicast frame with the simple goal of discovering the identities of logical link endpoint(s) reachable from a Logical Link Endpoint, Section 9.

The HELLO and OPEN, Section 11, PDUs, which are used to discover and exchange detailed Logical Link Endpoint Identifiers, LLEIs, and the ACK/ERROR PDU, are mandatory; other PDUs are optional; though at least one encapsulation SHOULD be agreed at some point.

The following is a ladder-style diagram of the L3DL protocol exchanges:

```
<table>
<thead>
<tr>
<th>HELLO</th>
<th>Logical Link Peer discovery</th>
</tr>
</thead>
<tbody>
<tr>
<td>HELLO</td>
<td>Mandatory</td>
</tr>
<tr>
<td>OPEN</td>
<td>MACs, IDs, etc.</td>
</tr>
<tr>
<td>ACK</td>
<td>Mandatory</td>
</tr>
<tr>
<td>OPEN</td>
<td></td>
</tr>
</tbody>
</table>
```
<--------------
   ACK
-------------->

Interface IPv4 Addresses
-------------->

Interface IPv4 Addresses
<--------------
   ACK
-------------->

Interface IPv6 Addresses
-------------->

Interface IPv6 Addresses
<--------------
   ACK
-------------->

Interface MPLSv4 Labels
-------------->

Interface MPLSv4 Labels
<--------------
   ACK
-------------->

Interface MPLSv6 Labels
-------------->

Interface MPLSv6 Labels
<--------------
   ACK
-------------->

Interface MPLSv6 Labels
<--------------
   ACK
-------------->

Optional
6. Transport Layer

L3DL PDUs are carried by a simple transport layer which allows long PDUs to occupy many Ethernet frames. The L3DL content of a single Ethernet frame, exclusive of Ethernet framing data, is referred to as a Datagram.

The L3DL Transport Layer encapsulates each Datagram using a common transport header.

If a PDU does not fit in a single datagram, it is broken into multiple Datagrams and reassembled by the receiver a la [RFC0791] Section 2.3 Fragmentation.

Should a PDU need to be retransmitted, it MUST BE sent as the identical Datagram set as the original transmission. The Transmission Sequence Number informs the receiver that it is the same PDU.

```
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
|                Version                | Transmission Sequence Number |L|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
| Datagram Number | Datagram Length |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
|                      Checksum                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
|                   Payload...                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
```

The fields of the L3DL Transport Header are as follows:

Version: Seven-bit Version number of the protocol, currently 0. Values other than 0 MUST BE treated as an error. The protocol version needs to be in one and only one place, so it is in the datagram as opposed to, for example, the PDU header.

L: A bit that set to one if this Datagram is the last Datagram of the PDU. For a PDU which fits in only one Datagram, it is set to one. Note that this is the inverse of the marking technique used by [RFC0791].
Transmission Sequence Number: A 16-bit strictly increasing unsigned integer identifying this PDU, possibly across retransmissions, that wraps from $2^{16}-1$ to 0. The initial value is arbitrary. See [RFC1982] on DNS Serial Number Arithmetic for too much detail on comparison and incrementing a wrapping sequence number.

Datagram Number: A monotonically increasing 24-bit value which starts at zero for each PDU. This is used to reassemble frames into PDUs a la [RFC0791] Section 2.3. Note that this limits an L3DL PDU to $2^{24}$ frames.

Datagram Length: Total number of octets in the Datagram including all payloads and fields. Note that this limits a datagram to $2^{16}$ octets; though Ethernet framing is likely to impose a smaller limit.

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

If a Datagram fails checksum verification, the datagram is invalid and should be silently discarded. The sender will retransmit the PDU, and the receiver can assemble it.

Payload: The PDU being transported or a fragment thereof.

To avoid the need for a receiver to reassemble two PDUs at the same time, a sender MUST NOT send a subsequent PDU when a PDU is already in flight and not yet acknowledged; assuming it is an ACKed PDU Type.

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 prudent 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.
/* 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, 0xf4, 0xb3, 0x21, 0x15, 0x78,
  0x99, 0xb1, 0xaf, 0xf9, 0xe7, 0x2d, 0x4d, 0xa8, 0xce, 0x4c, 0xca, 0x2e,
  0x52, 0x95, 0xd9, 0x1e, 0xe4, 0x3e, 0x44, 0x28, 0x0a, 0xdf, 0x02, 0xa0,
  0x17, 0xf1, 0x60, 0x68, 0xb7, 0x7a, 0xc3, 0xe9, 0xfa, 0x3d, 0x53,
  0x96, 0x84, 0x6b, 0xfa, 0xf2, 0x63, 0x9a, 0x19, 0x7c, 0xae, 0xe5, 0xf5,
  0xf7, 0x16, 0x6a, 0xa2, 0x39, 0xb6, 0x7b, 0x0f, 0xc1, 0x93, 0x81, 0x1b,
  0xee, 0xb4, 0x1a, 0xea, 0xd0, 0x91, 0x2f, 0xb8, 0x55, 0xb9, 0xda, 0x85,
  0x3f, 0xe1, 0xbf, 0xe0, 0x5a, 0x58, 0x80, 0x5f, 0x66, 0x0b, 0xd8, 0x90,
  0x35, 0xd5, 0xc0, 0xa7, 0x33, 0xa6, 0x65, 0x69, 0x45, 0x00, 0x94, 0x56,
  0xe4, 0x98, 0x9b, 0x76, 0x97, 0xf0, 0x2c, 0xb0, 0x0f, 0x0d, 0xb2, 0x20,
  0xe1, 0xcb, 0x7d, 0x5e, 0x47, 0x4a, 0xd4, 0x91, 0x8d, 0xe9, 0xe6,
  0x49, 0x3c, 0xc7, 0x27, 0xd2, 0x07, 0xd4, 0xda, 0x7c, 0x67, 0x18,
  0x89, 0xcb, 0x30, 0x1f, 0x84, 0xc6, 0x8f, 0xaa, 0xc8, 0x74, 0xdc, 0xc9,
  0x5d, 0x5c, 0x31, 0xa4, 0x70, 0x88, 0x61, 0x2c, 0x9f, 0xd0, 0x92, 0x87,
  0x50, 0x82, 0x54, 0x64, 0x26, 0x7d, 0x30, 0x40, 0x34, 0x4b, 0xc1, 0x73,
  0xda, 0xc4, 0xdf, 0x3b, 0xcc, 0xfb, 0x7f, 0x9b, 0xe6, 0xe3, 0x25, 0xa5,
  0xad, 0x04, 0x23, 0x9c, 0x14, 0x51, 0x22, 0xf0, 0x29, 0x79, 0x71, 0x7e,
  0xff, 0x8c, 0xe0, 0xe2, 0x0c, 0xef, 0xbc, 0x72, 0x75, 0x6f, 0x37, 0xa1,
  0xec, 0xd3, 0x8e, 0x62, 0x8b, 0x86, 0x10, 0xe8, 0x08, 0x77, 0x11, 0xe8,
  0x92, 0xf4, 0xb4, 0xc5, 0x32, 0x36, 0x9d, 0xcf, 0xf3, 0xa6, 0xbb, 0xac,
  0x5e, 0x6c, 0xa9, 0x13, 0x57, 0x25, 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 L3DL application layer PDU is a typical TLV (Type Length Value) PDU. It includes a signature to provide optional integrity and authentication. 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
+-----------------------------------------------+-------------------+
|    PDU Type   |                 Payload Length                |
+-----------------------------------------------+-------------------+-------------------+
|                                Payload ...     |
+-----------------------------------------------+-------------------+-------------------+
|    Sig Type   |        Signature Length       |               |
+-----------------------------------------------+-------------------+-------------------+-------------------+
|                                Signature     |
+-----------------------------------------------+-------------------+-------------------+-------------------+
```

The fields of the basic L3DL header are as follows:

- **PDU Type**: An integer differentiating PDU payload types. See Section 22.1.
- **Payload Length**: Total number of octets in the Payload field.
- **Payload**: The application layer content of the L3DL PDU.
- **Sig Type**: The type of the Signature, see Section 22.2. Type 0, a null signature, is defined in this document.
- **Signature Length**: The length of the Signature, possibly including padding, in octets. If Sig Type is 0, Signature Length MUST BE 0.
- **Signature**: The result of running the signature algorithm specified in Sig Type over all octets of the PDU except for the Signature itself.
9. Logical Link Endpoint Identifier

L3DL discovers neighbors on logical links and establishes sessions between the two ends of all consenting discovered logical links. A logical link is described by a pair of Logical Link Endpoint Identifiers, LLEIs.

An LLEI is a variable length descriptor which could be an ASN, a classic RouterID, a catenation of the two, an eight octet ISO System Identifier [RFC1629], or any other identifier unique to a single logical link endpoint in the topology.

An L3DL deployment will choose and define an LLEI which suits its needs, simple or complex. Examples of two extremes follow:

A simplistic view of a link between two devices is two ports, identified by unique MAC addresses, carrying a layer 3 protocol conversation. In this case, the MAC addresses might suffice for the LLEIs.

Unfortunately, things can get more complex. Multiple VLANs can run between those two MAC addresses. In practice, many real devices use the same MAC address on multiple ports and/or sub-interfaces.

Therefore, in the general circumstance, a fully described LLEI might be as follows:

```
  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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            ifIndex                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
System Identifier, a la [RFC1629], is an eight octet identifier unique in the entire operational space. Routers and switches usually have internal MAC Addresses which can be padded with high order zeros and used if no System ID exists on the device. If no unique identifier is burned into a device, the local L3DL configuration SHOULD create and assign a unique one, likely by configuration.

ifIndex is the SNMP identifier of the (sub-)interface, see [RFC1213]. This uniquely identifies the port.
For a layer 3 tagged sub-interface or a VLAN/SVI interface, Ifindex is that of the logical sub-interface, so no further disambiguation is needed.

L3DL 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.

LLEIs are big-endian.

10. HELLO

The HELLO PDU is unique in that it is encapsulated in a multicast Ethernet frame. It solicits response(s) from other LLEI(s) on the link. See Section 18.1 for why multicast is used. The destination multicast MAC Addresses 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)). This SHOULD BE used when the link is known to be a simple point to point link.

To Be Assigned: When a switch receives a frame with a multicast destination MAC it does not recognize, it forwards to all ports. This destination MAC is to be sent when the interface is known to be connected to a switch. See Section 23. This SHOULD BE used when the link may be a multi-point link.

All other L3DL PDUs are encapsulated in unicast frames, as the peer’s destination MAC address is known after the HELLO exchange.

When an interface is turned up on a device, it SHOULD issue a HELLO if it is to participate in L3DL sessions.

If a constrained Nearest Bridge destination address has been configured for a point-to-point interface, see above, then the HELLO SHOULD NOT be repeated once a session has been created by an exchange of OPENs.

If the configured destination address is one that is propagated by switches, the HELLO SHOULD be repeated at a configured interval, with a default of 60 seconds. This allows discovery by new devices which come up on the layer-2 mesh.
If more than one device responds, one adjacency is formed for each unique source LLEI response. L3DL treats each adjacency as a separate logical link.

When a HELLO is received from a source MAC address (plus VID if VLAN) with which there is no established L3DL session, the receiver SHOULD respond by sending an OPEN PDU to the source MAC address (plus VID). The two devices establish an L3DL session by exchanging OPEN PDUs.

If a HELLO is received from a MAC address with which there is an established session, the HELLO should be dropped.

The Payload Length is zero as there is no payload.

HELLO PDUs can not be signed as keying material has yet to be exchanged. Hence the signature MUST always be the null type.

11. OPEN

Each device has learned the other’s MAC Address from the HELLO exchange, see Section 10. Therefore the OPEN and all subsequent PDUs MUST BE unicast, as opposed to the HELLO’s multicast frame.
The Payload Length is the number of octets in all fields of the PDU from the Nonce through the Serial Number, not including the three final signature fields.

The Nonce enables detection of a duplicate OPEN PDU. It SHOULD be either a random number or a high resolution timestamp. It is needed to prevent session closure due to a repeated OPEN caused by a race or a dropped or delayed ACK.

My LLEI is the sender’s LLEI, see Section 9.

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

A node may have zero or more operator-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 Type is the Signature algorithm suite, see Section 8.

Key Length is a 16-bit field denoting the length in octets of the Key itself, not including the Auth Type or the Key Length. If the Auth Type is zero, then the Key Length MUST also be zero, and there MUST BE no Key data.
The Key is specific to the operational environment. A failure to authenticate is a failure to start the L3DL session, an ERROR PDU MUST be sent (Error Code 3), and HELLOs MUST be restarted.

The Serial Number is that of the last received and processed PDU. This allows a receiver sending an OPEN to tell the sender that the receiver wants to resume a session and the sender only needs to send data more recent than the Serial Number. If this OPEN is not trying to restart a lost session, the Serial Number MUST BE set to zero.

The Signature fields are described in Section 8 and in an asymmetric key environment serve as a proof of possession of the signing auth data by the sender.

Once two logical link endpoints know each other, and have ACKed each other’s OPEN PDUs, Layer 2 KEEPALIVEs (see Section 15) MAY be started to ensure Layer 2 liveness and keep the session semantics alive. The timing and acceptable drop of KEEPALIVE PDUs are discussed in Section 15.

If a sender of OPEN does not receive an ACK of the OPEN PDU, 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 an LLEI with which the receiving logical link endpoint believes it already has an L3DL session (OPENs have already been exchanged), and the Serial Number in the OPEN is non-zero, the receiver SHOULD establish a new session by sending an OPEN with the Serial Number of the last data it received. Each party MUST resume sending encapsulations etc. subsequent to the other party’s Sequence Number. And each MUST retain all previously discovered encapsulation and other data.

If a properly authenticated OPEN arrives with a new Nonce from an LLEI with which the receiving logical link endpoint believes it already has an L3DL session (OPENs have already been exchanged), and the Serial Number in the OPEN is zero, then the receiver MUST assume that the sending LLEI or entire device has been reset. All previously discovered encapsulation data MUST NOT be kept and MUST BE withdrawn via the BGP-LS API and the recipient MUST respond with a new OPEN.
12. ACK

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

```
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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| PDU Type = 3 |               Payload Length = 5              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| ACKed PDU | EType |       Error Code      |                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Error Hint          |    Sig Type   |Signature Leng.  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               Signature ...                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

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

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

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

The Error Codes, noting protocol failures, are listed in Section 22.4. Someone stuck in the 1990s might think the catenation of EType and Error Code as an echo of 0x1zzz, 0x2zzz, etc. They might be right; or not.

The Error Hint, an arbitrary 16 bits, is any additional data the sender of the error PDU thinks will help the recipient or the debugger with the particular error.
The Signature fields are described in Section 8.

12.1. 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), and the interface is live at layer 2, 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 MAY BE considered closed in case of this ACK failure.

If the link is broken at layer 2, retransmission MAY BE retried when the link is restored.

13. The Encapsulations

Once the devices know each other’s LLEIs, know each other’s upper layer (L2.5 and L3) identities, have means to ensure link state, etc., the L3DL session is considered established, and the devices SHOULD exchange L3 interface encapsulations, L3 addresses, and L2.5 labels.

The Encapsulation types the peers exchange may be IPv4 (Section 13.3), IPv6 (Section 13.4), MPLS IPv4 (Section 13.6), MPLS IPv6 (Section 13.7), 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 12) merely acknowledges receipt. Only if both peers have sent the same Encapsulation Type is it safe for Layer 3 protocols 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 2) 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 logical 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 IP subnet.
13.1. The Encapsulation PDU Skeleton

The header for all encapsulation PDUs is as follows:

```
+-------------------------------+-------------------------------+-------------------------------+
<table>
<thead>
<tr>
<th>PDU Type</th>
<th>Payload Length</th>
<th>Count</th>
</tr>
</thead>
</table>
+-------------------------------+-------------------------------+-------------------------------+
| Encapsulation List...        | Sig Type                      |
| Signature Length             | Signature ...                 |
+-------------------------------+-------------------------------+-------------------------------+
```

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

The 24-bit Count is the number of Encapsulations in the Encapsulation list.

The Serial Number is a monotonically increasing 32-bit value representing the sender’s state in time. It may be an integer, a timestamp, etc. On session restart (new OPEN), a receiver MAY send the last received Session Number to tell the sender to only send newer data.

If a sender has multiple links on the same interface, separate state: data, ACKs, etc. must be kept for each peer session.

Over time, multiple Encapsulation PDUs may be sent for an interface as configuration changes.

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 Signature fields are described in Section 8.

The Receiver MUST acknowledge the Encapsulation PDU with a Type=3, ACK PDU (Section 12) with the Encapsulation Type being that of the encapsulation being announced, see Section 12.

If the Sender does not receive an ACK in a configurable interval (default one second), and the interface is live at layer 2, they SHOULD retransmit. After a user configurable number of failures.
Internet-Draft       Layer 3 Discovery and Liveness        November 2019

(default three), the L3DL session should be considered dead and the
OPEN process SHOULD be restarted.

If the link is broken at layer 2, retransmission MAY BE retried if
data have not changed in the interim.

13.2. Encapsulation Flags

The Encapsulation Flags are a sequence of bit fields as follows:

```
  0 1 2 3 4 5 6 7
+----------------+----------------+----------------+----------------+----------------+
| Ann/With | Primary | Under/Over | Loopback | Reserved ..|
+----------------+----------------+----------------+----------------+----------------+
```

Each encapsulation in an Encapsulation PDU of Type T may announce new
and/or withdraw old encapsulations of Type T. It indicates this with
the Ann/With Encapsulation Flag, Announce == 1, Withdraw == 0.

Each Encapsulation interface address in an Encapsulation PDU is
either a new encapsulation be announced (Ann/With == 1) (yes, a la
BGP) or requests one be withdrawn (Ann/With == 0). Adding an
encapsulation which already exists SHOULD raise an Announce/Withdraw
Error (see Section 22.4); the EType SHOULD be 2, suggesting a session
restart (see Section 12 so all encapsulations will be resent.

If an LLEI has multiple addresses for an encapsulation type, one and
only one address MAY be marked as primary (Primary Flag == 1) for
that Encapsulation Type.

An Encapsulation interface address in an Encapsulation PDU MAY be
marked as a loopback, in which case the Loopback bit is set.
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 L3DL speaking external interfaces, e.g.
for iBGP peering. They SHOULD be marked as such, Loopback Flag == 1.

Each Encapsulation interface address in an Encapsulation PDU is that
of the direct 'underlay interface (Under/Over == 1), or an 'overlay'
address (Under/Over == 0), likely that of a VM or container guest
bridged or configured on to the interface already having an underlay
address.

13.3. 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.
The 24-bit Count is the sum of the number of IPv4 Encapsulations being announced and/or withdrawn.

13.4. IPv6 Encapsulation

The IPv6 Encapsulation describes a logical link’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.
The 24-bit Count is the sum of the number of IPv6 Encapsulations being announced and/or withdrawn.

13.5. MPLS Label List

As an MPLS enabled interface may have a label stack, see [RFC3032], a variable length list of labels is needed. These are the labels the sender will accept for the prefix to which the list is attached.

```
+---------------+----------+
| Label Count   | Label    |
+---------------+----------+
|                | Exp      |
+---------------+----------+
|                 | S        |
+---------------+----------+

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

13.6. MPLS IPv4 Encapsulation

The MPLS IPv4 Encapsulation describes a logical link’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 which will be accepted for each address.

```
+------------+-------------------+
| PDU Type = 6 | Payload Length    |
|             |                   |
+------------+-------------------+
| Count      |                   |
+------------+-------------------+
| Serial Number |
+------------+-------------------+
| Encaps Flags | MPLS Label List ...
|             |                   |
+------------+-------------------+
| IPv4 Address | PrefixLen |
| more ... | Sig Type |
|           | Signature Length |
+------------+-------------------+
| Signature  |
+------------+-------------------+
```

The 24-bit Count is the sum of the number of MPLSv4 Encapsulation being announced and/or withdrawns.
13.7.  MPLS IPv6 Encapsulation

The MPLS IPv6 Encapsulation describes a logical link’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 which will be accepted for each address.

```
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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  PDU Type = 7 |                 Payload Length                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     Count                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Serial Number                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Encaps Flags  |      MPLS Label List ...      |               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+               +
|                                                               |
|                                                               |
|                          IPv6 Address                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                               Prefix Len |
|                             more ...   |    Sig Type   |        Signature Length       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Signature ...                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The 24-bit Count is the sum of the number of MPLSv6 Encapsulations being announced and/or withdrawn.

14.  VENDOR - Vendor Extensions
Vendors or enterprises may define TLVs beyond the scope of L3DL standards. This is done using a Private Enterprise Number [IANA-PEN] followed by Enterprise Data in a format defined for that Enterprise Number and Ent Type.

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. KEEPALIVE - Layer 2 Liveness

L3DL devices SHOULD beacon frequent Layer 2 KEEPALIVE PDUs to ensure session continuity. The inter-KEEPALIVE interval is configurable, with a default of ten seconds. A receiver may choose to ignore KEEPALIVE PDUs.

An operational deployment MUST BE configured whether to use KEEPALIVEs or not, either globally, or as finely as to per-link granularity. Disagreement MAY result in repeated session failure and reestablishment.
KEEPALIVEs SHOULD be beaconed at a configured frequency. One per second is the default. Layer 3 liveness, such as BFD, may be more (or less) aggressive.

When a sender transmits a PDU which is not a KEEPALIVE, the sender SHOULD reset the KEEPALIVE timer. I.e. sending any PDU acts as a keepalive. Once the last fragment has been sent, the KEEPALIVE timer SHOULD BE restarted. Do not wait for the ACK.

If a KEEPALIVE or other PDUs have not been received from a peer with which a receiver has an open session for a configurable time (default 30 seconds), the link SHOULD BE presumed down. The devices MAY keep configuration state and restore it without retransmission if no data have changed. Otherwise, a new session SHOULD BE established and new Encapsulation PDUs exchanged.

16. Layers 2.5 and 3 Liveness

Layer 2 liveness may be continuously tested by KEEPALIVE PDUs, see Section 15. As layer 2.5 or layer 3 connectivity could still break, liveness above layer 2 MAY be frequently tested using BFD ([RFC5880]) or a similar technique.

This protocol assumes that one or more Encapsulation addresses may be used to ping, run BFD, or whatever the operator configures.

17. The North/South Protocol

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

The devices know their unique LLEIs and know the unique peer LLEIs and Encapsulations on each logical link interface.

Full topology discovery is not appropriate at the L3DL layer, so Dijkstra a la IS-IS etc. is assumed to be done by higher level protocols such as BGP-SPF.

Therefore the LLEIs, 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.
17.1. Use BGP-LS as Much as Possible

BGP-LS [RFC7752] defines BGP-like Datagrams describing logical 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.

17.2. 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-bgpls-segment-routing-epe]. The local and remote node descriptors for all NLRI are the IDs described in Section 11. 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.

18. Discussion

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

18.1. HELLO Discussion

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 logical interface (LLEI), I, on an L3DL 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, on multi-link interfaces, L3DL MUST continue to send HELLOs as long as they are turned up.
18.2. HELLO versus KEEPALIVE

Both HELLO and KEEPALIVE are periodic. KEEPALIVE might be eliminated in favor of keeping only HELLOs. But KEEPALIVES are unicast, and thus less noisy on the network, especially if HELLO is configured to transit layer-2-only switches, see Section 18.1.

19. VLANs/SVIs/Sub-interfaces

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

As the upper routing layer must view VLAN topologies as separate graphs, L3DL treats VLAN ports as separate links.

L3DL 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 LLIEs, they act as separate interfaces, forming their own links.

20. Implementation Considerations

An implementation SHOULD provide the ability to configure each logical interface as L3DL speaking or not.

An implementation SHOULD provide the ability to configure whether HELLOs on an L3DL enabled interface send Nearest Bridge or the MAC which is propagated by switches from that interface; see Section 10.

An implementation SHOULD provide the ability to distribute one or more loopback addresses or interfaces into L3DL on an external L3DL speaking interface.

An implementation SHOULD provide the ability to distribute one or more overlay and/or underlay addresses or interfaces into L3DL on an external L3DL speaking interface.

An implementation SHOULD provide the ability to configure one of the addresses of an encapsulation as primary on an L3DL speaking interface. If there is only one address for a particular encapsulation, the implementation MAY mark it as primary by default.

An implementation MAY allow optional configuration which updates the local forwarding table with overlay and underlay data both learned from L3DL peers and configured locally.
21. Security Considerations

The protocol as is MUST NOT be used outside a datacenter or similarly closed environment without authentication and authorisation mechanisms such as [I-D.ymbk-lsvr-l3dl-signing].

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. In the case of L3DL, Authentication and Integrity as provided in [I-D.ymbk-lsvr-l3dl-signing] is strongly recommended.

It is generally unwise to assume that on the wire Layer 2 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, form malicious sessions, etc. to divert, intercept, or drop traffic.

Similarly, malicious nodes/devices could mis-announce addressing.

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.

If the KEEPALIVE PDU is not signed (as suggested in Section 8) to save computation, then a MITM could fake a session being alive.

22. IANA Considerations

22.1. PDU Types

This document requests the IANA create a registry for L3DL PDU Type, which may range from 0 to 255. The name of the registry should be L3DL-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:
This document requests the IANA create a registry for L3DL Signature Type, AKA Sig Type, which may range from 0 to 255. The name of the registry should be L3DL-Signature-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>Number</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Null</td>
</tr>
<tr>
<td>1-255</td>
<td>Reserved</td>
</tr>
</tbody>
</table>

This document requests the IANA create a registry for L3DL PL Flag Bits, which may range from 0 to 7. The name of the registry should be L3DL-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>Announce/Withdraw (ann == 0)</td>
</tr>
<tr>
<td>1</td>
<td>Primary</td>
</tr>
<tr>
<td>2</td>
<td>Underlay/Overlay (under == 0)</td>
</tr>
<tr>
<td>3</td>
<td>Loopback</td>
</tr>
<tr>
<td>4-7</td>
<td>Reserved</td>
</tr>
</tbody>
</table>

This document requests the IANA create a registry for L3DL Error Codes, a 16 bit integer. The name of the registry should be L3DL-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>No Error</td>
</tr>
<tr>
<td>1</td>
<td>Checksum Error</td>
</tr>
<tr>
<td>2</td>
<td>Logical Link Addressing Conflict</td>
</tr>
<tr>
<td>3</td>
<td>Authorization Failure</td>
</tr>
<tr>
<td>4</td>
<td>Announce/Withdraw Error</td>
</tr>
</tbody>
</table>

23. IEEE Considerations

This document requires a new EtherType.

This document requires a new multicast MAC address that will be broadcast through a switch.

24. Acknowledgments

The authors thank Cristel Pelsser for multiple reviews, Harsha Kovuru for comments during implementation, 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, Nalinaksh Pai for transport discussions, Neeraj Malhotra for review, Paul Congdon for Ethernet hints, Russ Housley for checksum discussion and sBox, and Steve Bellovin for checksum advice.

25. References

25.1. Normative References

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

[I-D.ietf-idr-bgpls-segment-routing-epe]
Patel, K., Lindem, A., Zandi, S., and W. Henderickx,
"Shortest Path Routing Extensions for BGP Protocol",
draft-ietf-lsvr-bgp-spf-06 (work in progress), September 2019.

Bush, R. and R. Austein, "Layer 3 Discovery and Liveness Signing",
draft-ymbk-lsvr-l3dl-signing-00 (work in progress), October 2019.

"IANA Private Enterprise Numbers",

IEEE, "IEEE Standard for Local and Metropolitan Area Networks: Overview and Architecture",
IEEE 802-2001, DOI 10.1109/ieeestd.2002.93395, July 2002,

Institute of Electrical and Electronics Engineers, "Local and Metropolitan Area Networks: Overview and Architecture",

STD 17, RFC 1213, DOI 10.17487/RFC1213, March 1991,

Colella, R., Callon, R., Gardner, E., and Y. Rekhter,
"Guidelines for OSI NSAP Allocation in the Internet",
RFC 1629, DOI 10.17487/RFC1629, May 1994,

Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels",
BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997,

Rosen, E., Tappan, D., Fedorkow, G., Rekhter, Y.,
Farinacci, D., Li, T., and A. Conta, "MPLS Label Stack Encoding",
RFC 3032, DOI 10.17487/RFC3032, January 2001,


25.2. Informative References


[RFC0791] Postel, J., "Internet Protocol", STD 5, RFC 791,
DOI 10.17487/RFC0791, September 1981,

[RFC1122] Braden, R., Ed., "Requirements for Internet Hosts -
Communication Layers", STD 3, RFC 1122,
DOI 10.17487/RFC1122, October 1989,

DOI 10.17487/RFC1982, August 1996,

Authors’ Addresses

Randy Bush
Arrcus & Internet Initiative Japan
5147 Crystal Springs
Bainbridge Island, WA  98110
US

Email: randy@psg.com

Rob Austein
Arrcus, Inc

Email: sra@hactrn.net

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

Email: keyur@arrcus.com