LSVR K. Patel
Internet-Draft Arrcus, Inc.
Intended status: Informational A. Lindem
Expires: January 27, 2021 Cisco Systems
S. Zandi
G. Dawra
Linkedin
July 26, 2020

Usage and Applicability of Link State Vector Routing in Data Centers
draft-ietf-lsvr-applicability-06

Abstract

This document discusses the usage and applicability of Link State Vector Routing (LSVR) extensions in data center networks utilizing CLOS or Fat-Tree topologies. The document is intended to provide a simplified guide for the deployment of LSVR extensions.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

This document complements [I-D.ietf-lsvr-bgp-spf] by discussing the applicability of the technology in a simple and fairly common deployment scenario, which is described in Section 4.

After describing the deployment scenario, Section 5 will describe the reasons for BGP modifications for such deployments.

Once the control plane routing protocol requirements are described, Section 6 will cover the LSVR protocol enhancements to BGP to meet these requirements and their applicability to Data Center CLOS networks.

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.

3. Recommended Reading

This document assumes knowledge of existing data center networks and data center network topologies [CLOS]. This document also assumes knowledge of data center routing protocols like BGP [RFC4271], BGP-SPF [I-D.ietf-lsvr-bgp-spf], OSPF [RFC2328], as well as, data center OAM protocols like LLDP [RFC4957] and BFD [RFC5580].

4. Common Deployment Scenario

Within a Data Center, servers are commonly interconnected the CLOS topology [CLOS]. The CLOS topology is fully non-blocking and the topology is realized using Equal Cost Multi-Path (ECMP). In a CLOS topology, the minimum number of parallel paths between two servers is determined by the width of a tier-1 stage as shown in the figure 1.

The following example illustrates multi-stage CLOS topology.

```
                                      Tier-1
                                     +-----+
                                     |NODE |
                                  +->| 12  |--+
                                  |  +-----+  |
                          Tier-2  |           |   Tier-2
                         +-----+  |  +-----+  |  +-----+
           +------------>|NODE |--+->|NODE |--+--|NODE |-------------+
           |       +-----|  9  |--+  | 10  |  +--| 11  |-----+       |
           |       |     +-----+     +-----+     +-----+     |       |
           |       |                                         |       |
           |       |     +-----+     +-----+     +-----+     |       |
           | +-----+---->|NODE |--+  |NODE |  +--|NODE |-----+-----+ |
           | |     | +---|  6  |--+->|  7  |--+--|  8  |---+ |     | |
           | |     | |   +-----+  |  +-----+  |  +-----+   | |     | |
           | |     | |            |           |            | |     | |
         +-----+ +-----+          |  +-----+  |          +-----+ +-----+
         |NODE | |NODE | Tier-3   +->|NODE |--+   Tier-3 |NODE | |NODE |
         |  1  | |  2  |             |  3  |             |  4  | |  5  |
         +-----+ +-----+             +-----+             +-----+ +-----+
           | |     | |                                     | |     | |
           A O     B O            <- Servers ->            Z O     O O
```

Figure 1: Illustration of the basic CLOS

5. Justification for BGP SPF Extension

In order to simplify layer-3 routing and operations [RFC7938], many data centers use BGP as a routing protocol to create both an underlay and overlay network for their CLOS Topologies. However, BGP is a path-vector routing protocol. Since it does not create a fabric topology, it uses hop-by-hop EBGP peering to facilitate hop-by-hop routing to create the underlay network and to resolve any overlay next hops. The hop-by-hop BGP peering paradigm imposes several restrictions within a CLOS. It severely prohibits a deployment of Route Reflectors/Route Controllers as the EBGP sessions are congruent with the data path. The BGP best-path algorithm is prefix-based and it prevents announcements of prefixes to other BGP speakers until the best-path decision process has been performed for the prefix at each intermediate hop. These restrictions significantly delay the overall convergence of the underlay network within a CLOS network.

The LSVR SPF modifications allow BGP to overcome these limitations. Furthermore, using the BGP-LS NLRI format [RFC7752] allows the LSVR data to be advertised for nodes, links, and prefixes in the BGP routing domain and used for SPF computations.

6. LSVR Applicability to CLOS Networks

With the BGP SPF extensions [I-D.ietf-lsvr-bgp-spf], the BGP best-path computation and route computation are replaced with OSPF-like algorithms [RFC2328] both to determine whether an BGP-LS SPF NLRI has changed and needs to be re-advertised and to compute the BGP routes. These modifications will significantly improve convergence of the underlay while affording the operational benefits of a single routing protocol [RFC7938].

Data center controllers typically require visibility to the BGP topology to compute traffic-engineered paths. These controllers learn the topology and other relevant information via the BGP-LS address family [RFC7752] which is totally independent of the underlay address families (usually IPv4/IPv6 unicast). Furthermore, in traditional BGP underlays, all the BGP routers will need to advertise their BGP-LS information independently. With the BGP SPF extensions, controllers can learn the topology using the same BGP advertisements used to compute the underlay routes. Furthermore, these data center controllers can avail the convergence advantages of the BGP SPF extensions. The placement of controllers can be outside of the forwarding path or within the forwarding path.

Alternatively, as each and every router in the BGP SPF domain will have a complete view of the topology, the operator can also choose to configure BGP sessions in hop-by-hop peering model described in [RFC7938] along with BFD [RFC5580]. In doing so, while the hop-by-hop peering model lacks the inherent benefits of the controller-based model, BGP updates need not be serialized by BGP best-path algorithm in either of these models. This helps overall network convergence.

6.1. Usage of BGP-LS SPF SAFI

The BGP SPF extensions [I-D.ietf-lsvr-bgp-spf] define a new BGP-LS SPF SAFI for announcement of BGP SPF link-state. The NLRI format and its associated attributes follow the format of BGP-LS for node, link, and prefix announcements. Whether the peering model within a CLOS follows hop-by-hop peering described in [RFC7938] or any controller-based or route-reflector peering, an operator can exchange BGP SPF SAFI routes over the BGP peering by simply configuring BGP SPF SAFI between the necessary BGP speakers.

The BGP-LS SPF SAFI can also co-exist with BGP IP Unicast SAFI which could exchange overlapping IP routes. The routes received by these SAFIs are evaluated, stored, and announced independently according to the rules of [RFC4760]. The tie-breaking of route installation is a matter of the local policies and preferences of the network operator.

Finally, as the BGP SPF peering is done following the procedures described in [RFC4271], all the existing transport security mechanisms including [RFC5925] are available for the BGP-LS SPF SAFI.

6.1.1. Relationship to Other BGP AFI/SAFI Tuples

Normally, the BGP-LS AFI/SAFI is used solely to compute the underlay and is given preference over other AFI/SAFIs. Other BGP SAFIs, e.g., IPv6/IPv6 Unicast VPN would use the BGP-SPF computed routes for next hop resolution. However, if BGP-LS NLRI is also being advertised for controller consumption, there is no need to replicate the Node, Link, and Prefix NLRI in BGP-NLRI. Rather, additional NLRI attributes can be advertised in the BGP-LS SPF AFI/SAFI as required (e.g., BGP-LS TE metric extensions [RFC8571] and BGP-LS segment routing extensions [I-D.ietf-idr-bgp-ls-segment-routing-ext]).

6.2. Peering Models

As previously stated, BGP SPF can be deployed using the existing peering model where there is a single-hop BGP session on each and every link in the data center fabric [RFC7938]. This provides for both the advertisement of routes and the determination of link and neighboring switch availability. With BGP SPF, the underlay will converge faster due to changes to the decision process that will allow NLRI changes to be advertised faster after detecting a change.

6.2.1. Sparse Peering Model

Alternately, BFD [RFC5580] can be used to swiftly determine the availability of links and the BGP peering model can be significantly sparser than the data center fabric. BGP SPF sessions only need to be established with enough peers to provide a bi-connected graph. If IEBGP is used, then the BGP routers at tier N-1 will act as route-reflectors for the routers at tier N.

The obvious usage of sparse peering is to avoid parallel sessions on links between the same two BGP speakers in the data center fabric. However, this use case is not very useful since parallel layer-3 links between the same two BGP routers are rare in CLOS or Fat-Tree topologies. Two more interesting scenarios are described below.

In current data center topologies, there is often a very dense mesh of links between levels, e.g., leaf and spine, providing 32-way, 64-way, or more Equal-Cost Multi-Path (ECMP) paths. In these topologies, it is desirable not to have a BGP session on every link and techniques such as the one described in Section 6.2.2 can be used establish sessions on some subset of northbound links. For example, in a Spine-Leaf topology, each leaf switch would only peer with a subset of the spines dependent on the flooding redundancy required to be reasonably certain that every node within the BGP-LS SPF routing domain has the complete topology.

Alternately, controller-based data center topologies are envisioned where BGP speakers within the data center only establish BGP sessions with two or more controllers. In these topologies, fabric nodes below the first tier (using [RFC7938] hierarchy) will establish BGP multi-hop sessions with the controllers. For the multi-hop sessions, determining the route to the controllers without depending on BGP would need to be through some other means beyond the scope of this document. However, the BGP discovery mechanisms described in Section 6.5 would be one possibility.

6.2.2. Bi-Connected Graph Heuristic

With this heuristic, discovery of BGP peers is assumed, e.g., as described in Section 6.5. Additionally, it assumed that the direction of the peering can be ascertained. In the context of a data center fabric, direction is either northbound (toward the spine), southbound (toward the Top-Of-Rack (TOR) switches) or east-west (same level in hierarchy. The determination of the direction is beyond the scope of this document. However, it would be reasonable to assume a technique where the TOR switches can be identified and the number of hops to the TOR is used to determine the direction.

In this heuristic, BGP speakers allow passive session establishment for southbound BGP sessions. For northbound sessions, BGP speakers will attempt to maintain two northbound BGP sessions with different switches (in data center fabrics there is normally a single layer-3 connection anyway). For east-west sessions, passive BGP session establishment is allowed. However, BGP speaker will never actively establish an east-west BGP session unless it cannot establish two northbound BGP sessions.

6.3. BGP Spine/Leaf Topology Policy

One of the advantages of using BGP SPF as the underlay protocol is that BGP policy can be applied at any level. For example, depending upon the topology, it may be possible to aggregate prefix advertisements using existing BGP policy. In Spine/Leaf topologies, it is not necessary to advertise BGP-LS NLRI received by leaves northbound to the spine nodes at the level above. If a common AS is used for the spine nodes, this can easily be accomplished with EBGP and a simple policy to filter advertisements from the leaves to the spine if the first AS in the AS path is the spine AS.

In the figure below, the leaves would not advertise any NLRI with AS 64512 as the first AS in the AS path.

```
                +--------+    +--------+             +--------+
    AS 64512    |        |    |        |             |        |
    for Spine   | Spine 1+----+ Spine 2+- ......... -+ Spine N|
    Nodes at    |        |    |        |             |        |
    this Level  +-+-+-+-++    ++-+-+-+-+             +-+-+-+-++
           +------+ | | |      | | | |                 | | | |
           |  +-----|-|-|------+ | | |                 | | | |
           |  |  +--|-|-|--------+-|-|-----------------+ | | |
           |  |  |  | | |    +---+ | |                   | | |
           |  |  |  | | |    |  +--|-|-------------------+ | |
           |  |  |  | | |    |  |  | |              +------+ +----+
           |  |  |  | | |    |  |  | +--------------|----------+  |
           |  |  |  | | |    |  |  +-------------+  |          |  |
           |  |  |  | | +----|--|----------------|--|--------+ |  |
           |  |  |  | +------|--|--------------+ |  |        | |  |
           |  |  |  +------+ |  |              | |  |        | |  |
          ++--+--++      +-+-+--++            ++-+--+-+     ++-+--+-+
          | Leaf 1|~~~~~~| Leaf 2|  ........  | Leaf X|     | Leaf Y|
          +-------+      +-------+            +-------+     +-------+
```

Figure 2: Spine/Leaf Topology Policy

6.4. BGP Peer Discovery Requirements

The most basic requirement is to be able to discover the address of a single-hop peer without pre-configuration. This is being accomplished today with using IPv6 Router Advertisements (RA) [RFC4861] and assuming that a BGP sessions is desired with any discovered peer. Beyond the basic requirement, it is useful to have to following information relating to the BGP session:

- o Autonomous System (AS) and BGP Identifier of a potential peer. The latter can be used for debugging and to decrease the likelihood of BGP session establishment collisions.

- o Security capabilities supported and for cryptographic authentication, the security capabilities and possibly a key-chain [RFC8177] to be used.

- o Session Policy Identifier - A group number or name used to associate common session parameters with the peer. For example, in a data center, BGP sessions with a Top of Rack (ToR) device could have parameters than BGP sessions between leaf and spine.

In a data center fabric, it is often useful to know whether a peer is southbound (towards the servers) or northbound (towards the spine or super-spine), e.g., Section 6.2.2. A potential requirement would be to determine this dynamically. One mechanism, without specifying all the details, might be for the ToR switches to be identified when installed and for the others switches in the fabric to determine their level based on the distance from the closest ToR switch.

If there are multiple links between BGP speakers or the links between BGP speakers are unnumbered, it is also useful to be able to establish multi-hop sessions using the loopback addresses. This will often require the discovery protocol to install route(s) toward the potential peer loopback addresses prior to BGP session establishment.

Finally, a simple BGP discovery protocol could also be used to establish a multi-hop session with one or more controllers by advertising connectivity to one or more controllers. However, once the multi-hop session actually traverses multiple nodes, it is bordering a distance-vector routing protocol and possibly this is not a good requirement for the discovery protocol.

6.5. BGP Peer Discovery

6.5.1. BGP Peer Discovery Alternatives

While BGP peer discovery is not part of [I-D.ietf-lsvr-bgp-spf], there are, at least, three proposals for BGP peer discovery. At least one of these mechanisms will be adopted and will be applicable to deployments other than the data center. It is strongly RECOMMENDED that the accepted mechanism be used in conjunction with BGP SPF in data centers. The BGP discovery mechanism should discovery both peer addresses and endpoints for BFD discovery. Additionally, it would be great if there were a heuristic for determining whether the peer is at a tier above or below the discovering BGP speaker (refer to Section 6.2.2).

The BGP discovery mechanisms under consideration are [I-D.acee-idr-lldp-peer-discovery], [I-D.xu-idr-neighbor-autodiscovery], and [I-D.ietf-lsvr-l3dl].

6.5.2. BGP IPv6 Simplified Peering

In order to conserve IPv4 address space and simplify operations, BGP-LS SPF routers in CLOS/Fat-Tree deployments can use IPv6 addresses as peer address. For IPv4 address families, IPv6 peering as specified in [RFC5549] can be deployed to avoid configuring IPv4 addresses on BGP-LS SPF router interfaces. When this is done, dynamic discovery mechanisms, as described in Section 6.5, can used to learn the global or link-local IPv6 peer addresses and IPv4 addresses need not be configured on these interfaces. If IPv6 link-local peering is used, then configuration of IPv6 global addresses is also not required and these IPv6 link-local addresses must then be advertised in the BGP-LS Link Descriptor IPv6 Address TLV (262) [RFC7752].

6.5.3. BGP-LS SPF Topology Visibility for Management

Irrespective of whether or not BGP-LS SPF is used for route calculation, the BGP-LS SPF route advertisements can be used to periodically construct the CLOS/FAT Tree topology. This is especially useful in deployments where an IGP is not used and the base BGP-LS routes [RFC7752] are not available. The resultant topology visibility can then be used for troubleshooting and consistency checking. This would normally be done on a central controller or other management tool which could also be used for fabric data path verification. The precise algorithms and heuristics, as well as, the complete set of management applications is beyond the scope of this document.

6.5.4. Data Center Interconnect (DCI) Applicability

Since BGP SPF is to be used for the routing underlay and DCI gateway boxes typically have direct or very simple connectivity, BGP external sessions would typically not include the BGP SPF SAFI.

7. Non-CLOS/FAT Tree Topology Applicability

The BGP SPF extensions [I-D.ietf-lsvr-bgp-spf] can be used in other topologies and avail the inherent convergence improvements. Additionally, sparse peering techniques may be utilized Section 6.2. However, determining whether or to establish a BGP session is more complex and the heuristic described in Section 6.2.2 cannot be used. In such topologies, other techniques such as those described in [I-D.ietf-lsr-dynamic-flooding] may be employed. One potential deployment would be the underlay for a Service Provider (SP) backbone where usage of a single protocol, i.e., BGP, is desired.

8. Non-Transit Node Capability

In certain scenarios, a BGP node wishes to participate in the BGP SPF topology but never be used for transit traffic. These in include situations where a server wants to make application services available to clients homed at subnets throughout the BGP SPF domain but does not ever want to be used as a router (i.e., carry transit traffic). Another specific instance is where a controller is resident on a server and direct connectivity to the controller is required throughout the entire domain. This can readily be accomplished using the BGP-LS Node NLRI Attribute SPF Status TLV as described in [I-D.ietf-lsvr-bgp-spf].

9. BGP Policy Applicability

Existing BGP policy including aggregation and prefix filtering may be used in conjunction with the BGP-LS SPF SAFI. When aggregation policy is used, BGP-LS SPF prefix NLRI will be originated for the aggregate prefix and BGP-LS SPF prefix NLRI for components will be filtered. Additionally, link and node NLRI may be filtered and the abstracted by the prefix NLRI.

When BGP policy is used with the BGP-LS SPF SAFI, BGP speakers in the BGP-LS SPF routing domain will not all have the same set of NLRI and will compute a different BGP local routing table. Consequently, care must be taken to assure routing is consistent and blackholes or routing loops do not ensue. However, this is no different than if tradition BGP routing using the IPv4 and IPv6 address families were used.

10. IANA Considerations

No IANA updates are requested by this document.

11. Security Considerations

This document introduces no new security considerations above and beyond those already specified in the [RFC4271] and [I-D.ietf-lsvr-bgp-spf].

12. Acknowledgements

The authors would like to thank Alvaro Retana, Yan Filyurin, and Boris Hassanov for their review and comments.

13. References

13.1. Normative References

[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-09 (work in progress), May 2020.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

13.2. Informative References

[CLOS] "A Study of Non-Blocking Switching Networks", The Bell System Technical Journal, Vol. 32(2), DOI 10.1002/j.1538-7305.1953.tb01433.x, March 1953.

[I-D.acee-idr-lldp-peer-discovery] Lindem, A., Patel, K., Zandi, S., Haas, J., and X. Xu, "BGP Logical Link Discovery Protocol (LLDP) Peer Discovery", draft-acee-idr-lldp-peer-discovery-07 (work in progress), June 2020.

[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-lsr-dynamic-flooding] Li, T., Psenak, P., Ginsberg, L., Chen, H., Przygienda, T., Cooper, D., Jalil, L., Dontula, S., and G. Mishra, "Dynamic Flooding on Dense Graphs", draft-ietf-lsr-dynamic-flooding-07 (work in progress), June 2020.

[I-D.ietf-lsvr-l3dl] Bush, R., Austein, R., and K. Patel, "Layer 3 Discovery and Liveness", draft-ietf-lsvr-l3dl-05 (work in progress), May 2020.

[I-D.xu-idr-neighbor-autodiscovery] Xu, X., Talaulikar, K., Bi, K., Tantsura, J., and N. Triantafillis, "BGP Neighbor Discovery", draft-xu-idr-neighbor-autodiscovery-12 (work in progress), November 2019.

[RFC2328] Moy, J., "OSPF Version 2", STD 54, RFC 2328, DOI 10.17487/RFC2328, April 1998, <https://www.rfc-editor.org/info/rfc2328>.

[RFC4271] Rekhter, Y., Ed., Li, T., Ed., and S. Hares, Ed., "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, DOI 10.17487/RFC4271, January 2006, <https://www.rfc-editor.org/info/rfc4271>.

[RFC4760] Bates, T., Chandra, R., Katz, D., and Y. Rekhter, "Multiprotocol Extensions for BGP-4", RFC 4760, DOI 10.17487/RFC4760, January 2007, <https://www.rfc-editor.org/info/rfc4760>.

[RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, DOI 10.17487/RFC4861, September 2007, <https://www.rfc-editor.org/info/rfc4861>.

[RFC4957] Krishnan, S., Ed., Montavont, N., Njedjou, E., Veerepalli, S., and A. Yegin, Ed., "Link-Layer Event Notifications for Detecting Network Attachments", RFC 4957, DOI 10.17487/RFC4957, August 2007, <https://www.rfc-editor.org/info/rfc4957>.

[RFC5549] Le Faucheur, F. and E. Rosen, "Advertising IPv4 Network Layer Reachability Information with an IPv6 Next Hop", RFC 5549, DOI 10.17487/RFC5549, May 2009, <https://www.rfc-editor.org/info/rfc5549>.

[RFC5580] Tschofenig, H., Ed., Adrangi, F., Jones, M., Lior, A., and B. Aboba, "Carrying Location Objects in RADIUS and Diameter", RFC 5580, DOI 10.17487/RFC5580, August 2009, <https://www.rfc-editor.org/info/rfc5580>.

[RFC5925] Touch, J., Mankin, A., and R. Bonica, "The TCP Authentication Option", RFC 5925, DOI 10.17487/RFC5925, June 2010, <https://www.rfc-editor.org/info/rfc5925>.

[RFC7752] Gredler, H., Ed., Medved, J., Previdi, S., Farrel, A., and S. Ray, "North-Bound Distribution of Link-State and Traffic Engineering (TE) Information Using BGP", RFC 7752, DOI 10.17487/RFC7752, March 2016, <https://www.rfc-editor.org/info/rfc7752>.

[RFC7938] Lapukhov, P., Premji, A., and J. Mitchell, Ed., "Use of BGP for Routing in Large-Scale Data Centers", RFC 7938, DOI 10.17487/RFC7938, August 2016, <https://www.rfc-editor.org/info/rfc7938>.

[RFC8177] Lindem, A., Ed., Qu, Y., Yeung, D., Chen, I., and J. Zhang, "YANG Data Model for Key Chains", RFC 8177, DOI 10.17487/RFC8177, June 2017, <https://www.rfc-editor.org/info/rfc8177>.

[RFC8571] Ginsberg, L., Ed., Previdi, S., Wu, Q., Tantsura, J., and C. Filsfils, "BGP - Link State (BGP-LS) Advertisement of IGP Traffic Engineering Performance Metric Extensions", RFC 8571, DOI 10.17487/RFC8571, March 2019, <https://www.rfc-editor.org/info/rfc8571>.

Authors' Addresses

Keyur Patel
Arrcus, Inc.
2077 Gateway Pl
San Jose, CA 95110
USA

Email: keyur@arrcus.com

Acee Lindem
Cisco Systems
301 Midenhall Way
Cary, NC 95110
USA

Email: acee@cisco.com

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

Email: szandi@linkedin.com

Gaurav Dawra
Linkedin
222 2nd Street
San Francisco, CA 94105
USA

Email: gdawra@linkedin.com

Network Working Group K. Patel
Internet-Draft Arrcus, Inc.
Intended status: Standards Track A. Lindem
Expires: January 27, 2021 Cisco Systems
S. Zandi
LinkedIn
W. Henderickx
Nokia
July 26, 2020

Shortest Path Routing Extensions for BGP Protocol
draft-ietf-lsvr-bgp-spf-10

Abstract

Many Massively Scaled Data Centers (MSDCs) have converged on simplified layer 3 routing. Furthermore, requirements for operational simplicity have led 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

Copyright Notice

Table of Contents

1. Introduction

Many Massively Scaled Data Centers (MSDCs) have converged on simplified layer 3 routing. Furthermore, requirements for operational simplicity have led 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 number of sessions as with the previous peering model unless there are parallel links between switches in the fabric.

2.3. BGP Peering in Route-Reflector or Controller Topology

In this model, BGP speakers peer solely with one or more Route Reflectors [RFC4456] or controllers. As in the previous model, direct connection discovery and liveliness detection for those connections are done outside the BGP protocol. More specifically, the Liveliness detection is done using BFD protocol described in [RFC5880]. For the purposes of BGP SPF, Link NLRI is advertised as long as the corresponding link is up and considered operational.

This peering model, known as sparse peering, allows for many fewer BGP sessions and, consequently, instances of the same NLRI received from multiple peers. It is discussed in greater detail in [I-D.ietf-lsvr-applicability].

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

In order to replace the Phase 1 and 2 decision functions of the existing Decision Process with an SPF-based Decision Process and streamline the Phase 3 decision functions in a backward compatible manner, this draft introduces the BGP-LS-SFP SAFI for BGP-LS SPF operation. The BGP-LS-SPF (AFI 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].

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              Type             |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| SPF Algorithm |
+-+-+-+-+-+-+-+-+
```

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).

4.1.2. 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.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   TBD Type    |                  Length                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| SPF Status    |
+-+-+-+-+-+-+-+-+

 BGP Status Values: 0 - Reserved
                    1 - Node Unreachable with respect to BGP SPF
                    2 - Node does not support transit with respect
                        to BGP SPF
                3-254 - Undefined
                  255 - Reserved
```

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. The metric value in this TLV is variable length dependent on specific protocol usage (refer to section 3.3.2.4 in [RFC7752]). For simplicity, the BGP-LS

SPF metric length will be 4 octets. Algorithms such as setting the metric inversely to the link speed as done in the OSPF MIB [RFC4750] MAY be supported. However, this is beyond the scope of this document.

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

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

```
     0                   1                   2                   3
     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
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |      TBD IPv4 or IPv6 Type    |             Length            |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    | Prefix-Length |
    +-+-+-+-+-+-+-+-+

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

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.

```
     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
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |   TBD Type    |                Length                         |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    | SPF Status    |
    +-+-+-+-+-+-+-+-+

     BGP Status Values: 0 - Reserved
                        1 - Link Unreachable with respect to BGP 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.

4.3.1. 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
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   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
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              Type             |             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 back-off 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.

- o 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.

- o 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.

- o 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 rebuilt during the course of the SPF computation. 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. 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 [RFC4760].

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 back-off [RFC8405], the current SPF back-off 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, Matt Anderson, Fred Baker, and Lukas Krattiger for their review and comments. Thanks to Pushpasis 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 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
Arrcus, Inc.
abhay@arrcus.com

Venu Venugopal
Cisco Systems
venuv@cisco.com

Chaitanya Yadlapalli
AT&T
cy098d@att.com

11. References

11.1. Normative References

[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-segment-routing-epe-19 (work in progress), May 2019.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC4271] Rekhter, Y., Ed., Li, T., Ed., and S. Hares, Ed., "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, DOI 10.17487/RFC4271, January 2006, <https://www.rfc-editor.org/info/rfc4271>.

[RFC7606] Chen, E., Ed., Scudder, J., Ed., Mohapatra, P., and K. Patel, "Revised Error Handling for BGP UPDATE Messages", RFC 7606, DOI 10.17487/RFC7606, August 2015, <https://www.rfc-editor.org/info/rfc7606>.

[RFC7752] Gredler, H., Ed., Medved, J., Previdi, S., Farrel, A., and S. Ray, "North-Bound Distribution of Link-State and Traffic Engineering (TE) Information Using BGP", RFC 7752, DOI 10.17487/RFC7752, March 2016, <https://www.rfc-editor.org/info/rfc7752>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8402] Filsfils, C., Ed., Previdi, S., Ed., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", RFC 8402, DOI 10.17487/RFC8402, July 2018, <https://www.rfc-editor.org/info/rfc8402>.

[RFC8405] Decraene, B., Litkowski, S., Gredler, H., Lindem, A., Francois, P., and C. Bowers, "Shortest Path First (SPF) Back-Off Delay Algorithm for Link-State IGPs", RFC 8405, DOI 10.17487/RFC8405, June 2018, <https://www.rfc-editor.org/info/rfc8405>.

11.2. Information References

[I-D.ietf-lsvr-applicability] Patel, K., Lindem, A., Zandi, S., and G. Dawra, "Usage and Applicability of Link State Vector Routing in Data Centers", draft-ietf-lsvr-applicability-05 (work in progress), March 2020.

[RFC2328] Moy, J., "OSPF Version 2", STD 54, RFC 2328, DOI 10.17487/RFC2328, April 1998, <https://www.rfc-editor.org/info/rfc2328>.

[RFC4456] Bates, T., Chen, E., and R. Chandra, "BGP Route Reflection: An Alternative to Full Mesh Internal BGP (IBGP)", RFC 4456, DOI 10.17487/RFC4456, April 2006, <https://www.rfc-editor.org/info/rfc4456>.

[RFC4724] Sangli, S., Chen, E., Fernando, R., Scudder, J., and Y. Rekhter, "Graceful Restart Mechanism for BGP", RFC 4724, DOI 10.17487/RFC4724, January 2007, <https://www.rfc-editor.org/info/rfc4724>.

[RFC4750] Joyal, D., Ed., Galecki, P., Ed., Giacalone, S., Ed., Coltun, R., and F. Baker, "OSPF Version 2 Management Information Base", RFC 4750, DOI 10.17487/RFC4750, December 2006, <https://www.rfc-editor.org/info/rfc4750>.

[RFC4760] Bates, T., Chandra, R., Katz, D., and Y. Rekhter, "Multiprotocol Extensions for BGP-4", RFC 4760, DOI 10.17487/RFC4760, January 2007, <https://www.rfc-editor.org/info/rfc4760>.

[RFC4790] Newman, C., Duerst, M., and A. Gulbrandsen, "Internet Application Protocol Collation Registry", RFC 4790, DOI 10.17487/RFC4790, March 2007, <https://www.rfc-editor.org/info/rfc4790>.

[RFC4915] Psenak, P., Mirtorabi, S., Roy, A., Nguyen, L., and P. Pillay-Esnault, "Multi-Topology (MT) Routing in OSPF", RFC 4915, DOI 10.17487/RFC4915, June 2007, <https://www.rfc-editor.org/info/rfc4915>.

[RFC5286] Atlas, A., Ed. and A. Zinin, Ed., "Basic Specification for IP Fast Reroute: Loop-Free Alternates", RFC 5286, DOI 10.17487/RFC5286, September 2008, <https://www.rfc-editor.org/info/rfc5286>.

[RFC5549] Le Faucheur, F. and E. Rosen, "Advertising IPv4 Network Layer Reachability Information with an IPv6 Next Hop", RFC 5549, DOI 10.17487/RFC5549, May 2009, <https://www.rfc-editor.org/info/rfc5549>.

[RFC5880] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD)", RFC 5880, DOI 10.17487/RFC5880, June 2010, <https://www.rfc-editor.org/info/rfc5880>.

[RFC7938] Lapukhov, P., Premji, A., and J. Mitchell, Ed., "Use of BGP for Routing in Large-Scale Data Centers", RFC 7938, DOI 10.17487/RFC7938, August 2016, <https://www.rfc-editor.org/info/rfc7938>.

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

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 30, 2021

R. Bush
Arrcus & Internet Initiative Japan
R. Austein
K. Patel
Arrcus

July 29, 2020

Layer 3 Discovery and Liveness
draft-ietf-lsvr-l3dl-06

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 neighbor IP addressing, logical link IP encapsulation abilities, 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

Copyright Notice

Table of Contents

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.

In this document, the use case for L3DL is for point to point links in a datacenter Clos in order to exchange the data needed for BGP-SPF [I-D.ietf-lsvr-bgp-spf] bootstrap and continuity. Once layer two connectivity has been leveraged to get layer three addressability and forwarding capabilities, normal layer three forwarding and routing can take over.

L3DL might be found to be more widely applicable to a range of routing and similar protocols which need layer three 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

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

```
   +-------------------+   +-------------------+   +-------------------+
   |      Device       |   |      Device       |   |      Device       |
   |                   |   |                   |   |                   |
   |+-----------------+|   |+-----------------+|   |+-----------------+|
   ||                 ||   ||                 ||   ||                 ||
   ||     BGP-SPF     <+---+>     BGP-SPF     <+---+>     BGP-SPF     ||
   ||                 ||   ||                 ||   ||                 ||
   |+--------^--------+|   |+--------^--------+|   |+--------^--------+|
   |         |         |   |         |         |   |         |         |
   |         |         |   |         |         |   |         |         |
   |+--------+--------+|   |+--------+--------+|   |+--------+--------+|
   ||  Encapsulations ||   ||  Encapsulations ||   ||  Encapsulations ||
   ||    Addresses    ||   ||    Addresses    ||   ||    Addresses    ||
   ||   L2 Liveness   ||   ||   L2 Liveness   ||   ||   L2 Liveness   ||
   |+--------^--------+|   |+--------^--------+|   |+--------^--------+|
   |         |         |   |         |         |   |         |         |
   |         |         |   |         |         |   |         |         |
   |+--------v--------+|   |+--------v--------+|   |+--------v--------+|
   ||                 ||   ||                 ||   ||                 ||
   ||Inter-Device PDUs<+---+>Inter-Device PDUs<+---+>Inter-Device PDUs||
   ||                 ||   ||                 ||   ||                 ||
   |+-----------------+|   |+-----------------+|   |+-----------------+|
   +-------------------+   +-------------------+   +-------------------+
```

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):

- o 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:

```
|             HELLO            | Logical Link Peer discovery
|----------------------------->|
```

```
   |            HELLO             | Mandatory
   |<-----------------------------|
   |                              |
   |                              |
   |            OPEN              | MACs, IDs, etc.
   |----------------------------->|
   |            ACK               |
   |<-----------------------------|
   |                              |
   |            OPEN              | Mandatory
   |<-----------------------------|
   |            ACK               |
   |----------------------------->|
   |                              |
   |                              |
   |   Interface IPv4 Addresses   | Interface IPv4 Addresses
   |----------------------------->| Optional
   |            ACK               |
   |<-----------------------------|
   |                              |
   |   Interface IPv4 Addresses   |
   |<-----------------------------|
   |            ACK               |
   |----------------------------->|
   |                              |
   |                              |
   |   Interface IPv6 Addresses   | Interface IPv6 Addresses
   |----------------------------->| Optional
   |            ACK               |
   |<-----------------------------|
   |                              |
   |   Interface IPv6 Addresses   |
   |<-----------------------------|
   |            ACK               |
   |----------------------------->|
   |                              |
   |                              |
   |   Interface MPLSv4 Labels    | Interface MPLSv4 Labels
   |----------------------------->| Optional
   |            ACK               |
   |<-----------------------------|
   |                              |
   |   Interface MPLSv4 Labels    | Interface MPLSv4 Labels
   |<-----------------------------| Optional
   |            ACK               |
   |----------------------------->|
   |                              |
   |                              |
```

```
 |   Interface MPLSv6 Labels   | Interface MPLSv6 Labels
 |---------------------------->| Optional
 |            ACK              |
 |<----------------------------|
 |                             |
 |   Interface MPLSv6 Labels   | Interface MPLSv6 Labels
 |<----------------------------| Optional
 |            ACK              |
 |---------------------------->|
 |                             |
 |                             |
 |        L3DL KEEPALIVE       | Layer 2 Liveness
 |---------------------------->| Optional
 |        L3DL KEEPALIVE       |
 |<----------------------------|
```

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.

This is not classic 'fragmentation', but rather decomposition at the origin to allow PDU payloads larger than the frame allows. There are no intermediate devices capable of further fragmentation or reassembly.

A PDU might need a large number of frames to be sent. As fragments are not ACK paced (as PDUs are), to avoid overwhelming bursts, the sender should pace fragments of a large PDU.

L3DL is carrying relatively small amounts of data on relatively high bandwidth links, and at a time when the link is not active with other data as it does not yet have layer three connectivity. So congestion is not considered a sufficiently significant risk to warrant additional complexity.

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
 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    |  Transmission Sequence Number |L| Dtgm Number ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~   Datagram Number (contd)     |        Datagram Length        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            Checksum                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           Payload...                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The fields of the L3DL Transport Header are as follows:

Version: Eight-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.

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 comparing and incrementing a wrapping sequence number.

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 23-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 a suggested algorithm. This specification avoids mandatory to implement, algorithm agility, etc. What matters is that the same algorithm is used consistently in any deployment.

Sum up 32-bit unsigned ints in a 64-bit long, then take the high-order section, shift it right filling on the left with zeros, rotate, add it in, repeat until the high order 32 bits are all zero.

```
<CODE BEGINS>
#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] = {
0xa3,0xd7,0x09,0x83,0xf8,0x48,0xf6,0xf4,0xb3,0x21,0x15,0x78,
0x99,0xb1,0xaf,0xf9,0xe7,0x2d,0x4d,0x8a,0xce,0x4c,0xca,0x2e,
0x52,0x95,0xd9,0x1e,0x4e,0x38,0x44,0x28,0x0a,0xdf,0x02,0xa0,
0x17,0xf1,0x60,0x68,0x12,0xb7,0x7a,0xc3,0xe9,0xfa,0x3d,0x53,
0x96,0x84,0x6b,0xba,0xf2,0x63,0x9a,0x19,0x7c,0xae,0xe5,0xf5,
0xf7,0x16,0x6a,0xa2,0x39,0xb6,0x7b,0x0f,0xc1,0x93,0x81,0x1b,
0xee,0xb4,0x1a,0xea,0xd0,0x91,0x2f,0xb8,0x55,0xb9,0xda,0x85,
0x3f,0x41,0xbf,0xe0,0x5a,0x58,0x80,0x5f,0x66,0x0b,0xd8,0x90,
0x35,0xd5,0xc0,0xa7,0x33,0x06,0x65,0x69,0x45,0x00,0x94,0x56,
0x6d,0x98,0x9b,0x76,0x97,0xfc,0xb2,0xc2,0xb0,0xfe,0xdb,0x20,
0xe1,0xeb,0xd6,0xe4,0xdd,0x47,0x4a,0x1d,0x42,0xed,0x9e,0x6e,
0x49,0x3c,0xcd,0x43,0x27,0xd2,0x07,0xd4,0xde,0xc7,0x67,0x18,
0x89,0xcb,0x30,0x1f,0x8d,0xc6,0x8f,0xaa,0xc8,0x74,0xdc,0xc9,
0x5d,0x5c,0x31,0xa4,0x70,0x88,0x61,0x2c,0x9f,0x0d,0x2b,0x87,
0x50,0x82,0x54,0x64,0x26,0x7d,0x03,0x40,0x34,0x4b,0x1c,0x73,
0xd1,0xc4,0xfd,0x3b,0xcc,0xfb,0x7f,0xab,0xe6,0x3e,0x5b,0xa5,
0xad,0x04,0x23,0x9c,0x14,0x51,0x22,0xf0,0x29,0x79,0x71,0x7e,
0xff,0x8c,0x0e,0xe2,0x0c,0xef,0xbc,0x72,0x75,0x6f,0x37,0xa1,
0xec,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 & 0xFFFFFFFFU);
  result = (result >> 32) + (result & 0xFFFFFFFFU);
  return (uint32_t) result;
}
<CODE ENDS>
```

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
 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   |                 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.

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 authenticity.

Other Sig Types may be defined in other documents, cf. [I-D.ymbk-lsvr-l3dl-signing].

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 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, 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 Addressees 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. In this multi-link scenario, the operator should be aware of the trade-off between timer tuning and network noise and adjust the inter-HELLO timer accordingly.

```
    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
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |  PDU Type = 0 |                Payload Length = 0             ~
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   ~               |  Sig Type = 0 |      Signature Length = 0     |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

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.

To ameliorate possible load spikes during bootstrap or event recovery, there SHOULD be a jittered delay between receipt of a HELLO and issue of the OPEN. The default delay range SHOULD BE zero to five seconds, and MUST be configurable.

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.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  PDU Type = 1 |                 Payload Length                ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~               |                     Nonce                     ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~               | LLEI Length   |            My LLEI            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-~
~                               |   AttrCount   |               ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~       Attribute List ...      |   Auth Type   |  Key Length   ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~               |                    Key ...                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Serial Number                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Sig Type   |        Signature Length       | Signature ... |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

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.

Although delay and jitter in responding with an OPEN were specified above, beware of load created by long strings of authentication failures and retries. A configurable failure count limit (default 8) SHOULD result in giving up on the connection attempt.

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 at L3DL speaker A with a new Nonce from an LLEI, speaker B, with which A believes it already has an L3DL session (OPENs have already been exchanged), and the Serial Number in the OPEN PDU is non-zero, speaker A SHOULD establish a new session by sending an OPEN with the Serial Number being the same as that of A's last sent and ACKed PDU. 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
 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 this 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:

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    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 (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. Encapsulaion Flags

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

```
 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 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.

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

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

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

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.

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

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
 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

```
    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
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   | PDU Type = 255|                 Payload Length                ~
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   ~               |                 Serial Number                 ~
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   ~               |               Enterprise Number               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |    Ent Type   |              Enterprise Data ...              ~
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   ~               |    Sig Type   |        Signature Length       |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                         Signature ...                         |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

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

```
    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
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |  PDU Type = 2 |               Payload Length = 0              ~
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   ~               |  Sig Type = 0 |      Signature Length = 0     |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

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 authorization 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:

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

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:

Number	Name
0	Null
1-255	Reserved

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:

Bit	Bit Name
0	Announce/Withdraw (ann == 0)
1	Primary
2	Underlay/Overlay (under == 0)
3	Loopback
4-7	Reserved

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:

Error Code	Error Name
0	No Error
1	Checksum Error
2	Logical Link Addressing Conflict
3	Authorization Failure
4	Announce/Withdraw Error

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, Joerg Ott for an early but deep transport review, 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]
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-segment-routing-epe-19 (work in progress), May 2019.

[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-10 (work in progress), July 2020.

[I-D.ymbk-lsvr-l3dl-signing]
Bush, R. and R. Austein, "Layer 3 Discovery and Liveness Signing", draft-ymbk-lsvr-l3dl-signing-01 (work in progress), May 2020.

[IANA-PEN]
"IANA Private Enterprise Numbers", <https://www.iana.org/assignments/enterprise-numbers/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, <http://ieeexplore.ieee.org/servlet/opac?punumber=7732>.

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

[RFC1213] McCloghrie, K. and M. Rose, "Management Information Base for Network Management of TCP/IP-based internets: MIB-II", STD 17, RFC 1213, DOI 10.17487/RFC1213, March 1991, <https://www.rfc-editor.org/info/rfc1213>.

[RFC1629] 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, <https://www.rfc-editor.org/info/rfc1629>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC3032] 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, <https://www.rfc-editor.org/info/rfc3032>.

[RFC4271] Rekhter, Y., Ed., Li, T., Ed., and S. Hares, Ed., "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, DOI 10.17487/RFC4271, January 2006, <https://www.rfc-editor.org/info/rfc4271>.

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", RFC 5226, DOI 10.17487/RFC5226, May 2008, <https://www.rfc-editor.org/info/rfc5226>.

[RFC5880] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD)", RFC 5880, DOI 10.17487/RFC5880, June 2010, <https://www.rfc-editor.org/info/rfc5880>.

[RFC6286] Chen, E. and J. Yuan, "Autonomous-System-Wide Unique BGP Identifier for BGP-4", RFC 6286, DOI 10.17487/RFC6286, June 2011, <https://www.rfc-editor.org/info/rfc6286>.

[RFC7752] Gredler, H., Ed., Medved, J., Previdi, S., Farrel, A., and S. Ray, "North-Bound Distribution of Link-State and Traffic Engineering (TE) Information Using BGP", RFC 7752, DOI 10.17487/RFC7752, March 2016, <https://www.rfc-editor.org/info/rfc7752>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

25.2. Informative References

[Clos0] Clos, C., "A study of non-blocking switching networks [PAYWALLED]", Bell System Technical Journal 32 (2), pp 406-424, March 1953.

[Clos1] "Clos Network", <https://en.wikipedia.org/wiki/Clos_network/>.

[I-D.malhotra-bess-evpn-lsoe] Malhotra, N., Patel, K., and J. Rabadan, "LSoE-based PE-CE Control Plane for EVPN", draft-malhotra-bess-evpn-lsoe-00 (work in progress), March 2019.

[JUPITER] Singh, A., Ong, J., Agarwal, A., Anderson, G., Armistead, A., Bannon, R., Boving, S., Desai, G., Felderman, B., Germano, P., Kanagala, A., Liu, H., Provost, J., Simmons, J., Tanda, E., Wanderer, J., HAP.lzle, U., Stuart, S., and A. Vahdat, "Jupiter rising", Communications of the ACM Vol. 59, pp. 88-97, DOI 10.1145/2975159, August 2016.

[RFC0791] Postel, J., "Internet Protocol", STD 5, RFC 791, DOI 10.17487/RFC0791, September 1981, <https://www.rfc-editor.org/info/rfc791>.

[RFC1122] Braden, R., Ed., "Requirements for Internet Hosts - Communication Layers", STD 3, RFC 1122, DOI 10.17487/RFC1122, October 1989, <https://www.rfc-editor.org/info/rfc1122>.

[RFC1982] Elz, R. and R. Bush, "Serial Number Arithmetic", RFC 1982, DOI 10.17487/RFC1982, August 1996, <https://www.rfc-editor.org/info/rfc1982>.

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

Network Working Group R. Bush
Internet-Draft Arrcus & IIJ
Intended status: Standards Track R. Austein
Expires: November 27, 2020 Arrcus
May 26, 2020

Layer 3 Discovery and Liveness Signing
draft-ietf-lsvr-l3dl-signing-00

Abstract

The Layer 3 Discovery and Liveness protocol OPEN PDU may contain a key and a certificate, which can be used to verify signatures on subsequent PDUs. This document describes two mechanisms based on digital signatures, one that is Trust On First Use (TOFU), and one that uses certificates to provide authentication as well as session integrity.

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

Table of Contents

1. Introduction

This draft is being published without incorporating changes from an excellent security review. This is being done so a couple of other drafts can reference it. While all comments will, of course, be appreciated, readers may want to wait for the -01 version.

The Layer 3 Discovery and Liveness protocol [I-D.ietf-lsvr-l3dl] OPEN PDU contains an algorithm specifier, a key, and a certificate, which can be used to verify signatures on subsequent PDUs. This document describes two methods of key generation and signing for use by L3DL, Trust On First Use (TOFU) and a PKI-based mechanism to provide authentication as well as session integrity.

The Key in the OPEN PDU SHOULD be the public key of an asymmetric key pair. The sender signs with the private key, of course. The device sending the OPEN may use one key for all links, a different key for each link, or some aggregation(s) thereof.

In the TOFU method the OPEN key is generated on the sending device, believed without question by the receiver, and used to verify all subsequent PDUs from the same sender with the same Key Type.

With the PKI-mechanism, an enrollment step is performed. The public key is put into a certificate [RFC5280], which is signed by the the operational environment's trust anchor. In this way, the relying party can be confident that the public key is under control of the identified L3DL protocol entity.

To the receiver verifying signatures on PDUs, the two methods are indistinguishable; the key provided in the OPEN PDU is used to verify the signatures of subsequent PDUs. The difference that PKI-based keys may be verified against the trust anchor when the OPEN PDU is received.

In the PKI method the OPEN key MUST be verified against the trust anchor for the operational domain. It is then used to verify all subsequent PDUs in the session.

2. Trust On First Use Method

There are three parts to using a key: signing PDUs, verifying the OPEN PDU, and verifying subsequent PDUs.

2.1. Signing a PDU

All signed PDUs are generated in the same way:

- o Compose the PDU, with all fields including "Sig Type" and "Signature Length" set, but omitting the trailing "Signature" field itself. The Certificate Length should be zero and the Certificate field should be empty. This is the "message to be signed" for purposes of the signature algorithm.

- o Generate the signature as specified for the chosen signature suite, using the private member of the asymmetric key pair. In general this will involve first hashing the "message to be signed" then signing the hash, but the precise details may vary with the specific algorithm. The result will be a sequence of octets, the length of which MUST be equal to the setting of the "Signature Length" field.

- Construct the complete message by appending the signature octets to the otherwise complete message composed above.

In the case of the OPEN PDU, the message to be signed will include the public member of the asymmetric keypair, but as far as the signature algorithm is concerned that's just payload, no different from any other PDU content.

2.2. Verifying the OPEN PDU

The process for verifying an OPEN PDU is slightly different from the process for verifying other PDU types, because the OPEN PDU also establishes the session key.

- Verify that the PDU is syntactically correct, and extract the Auth Type, Key, Sig Type, and Signature fields.

- Verify that Auth Type and Sig Type refer to the same algorithm suite, and that said algorithm suite is one that the implementation understands.

- Construct the "message to be verified" by truncating the PDU to remove the Signature field (in practice this should not require copying any data, just subtract the signature length from the PDU length).

- Verify the message constructed above against the public key using the rules for the specific signature suite.

- Record Auth Type and Key as this sessions's authentication type and session key, for use in verifying subseuqent PDUs.

If any of the above verification steps fail, generate an error using error code 2 ("Authorization failure in OPEN").

2.3. Verifying Other PDUs

The process for verifying non-OPEN PDUs is slightly simpler, but follows the same basic pattern as for OPEN PDUs.

- Verify that the PDU is syntactically correct, and extract the Sig Type and Signature fields.

- Verify that Sig Type refers to the same algorithm suite as the Auth Type recorded during verification of the OPEN PDU.

- Construct the "message to be verified" by truncating the PDU to remove the Signature field.

- o Verify the message constructed above against the recorded session key using the rules for the specific signature suite.

If any of the above verification steps fail, generate an error using error code 3 ("Signature failure in PDU").

3. Public Key Infrastructure Method

Using a PKI, [RFC5280], is almost the same as using TOFU, but with one additional step: during verification of an OPEN PDU, after extracting the Key field from the PDU but before attempting to use it to verify the PDU’s signature, the receiver MUST verify the received key against the PKI to confirm that it’s an authorized key.

Generating an OPEN PDU using the PKI method requires a certificate, which must be supplied via out of band configuration. The certificate is a signature of the public key to be sent in the Key field of the OPEN PDU, signed by the trust anchor private key.

Verifying an OPEN PDU using the PKI method requires the public key of the trust anchor, which the receiver uses to verify the certificate, thereby demonstrating that the supplied is represents an authorized L3DL speaker in this administrative domain.

We use the term "certificate" here in the generic sense. These are not X.509 certificates: X.509 is much more complicated than we need for L3DL. The certificates used here are just signatures of one key (the session key supplied in the Key field of the OPEN PDU) by another key (the trust anchor).

3.1. Signing OPEN PDU with PKI

Generating and signing the OPEN PDU with the PKI method is almost the same as in Section 2.1. The only difference is that the PKI method MUST supply the appropriate certificate in the Certificate field.

Note that the Auth Type field applies to both the Key and Certificate fields. That is: the certificate uses the same certificate suite as the session keys, L3DL does not support cross-algorithm-suite certification.

3.2. Verifying OPEN PDU with PKI

Verifying the OPEN PDU with PKI is similar to verifying with TOFU as described in Section 2.2, but includes one critical extra step:

After extracting the Key field from the PDU but before verifying the Signature, extract the Certificate field and verfiy that the

Certificate is a valid signature of the Key field, according to the rules for the signature suite specified by Auth Type. If this step fails, handle as in Section 2.2.

4. Local Policy

Whether to use TOFU, PKI, or no signatures at all is a matter of local policy, to be decided by the operator. The useful policy combinations for Key and Certificate are probably:

- o Not signing: sender need not sign, receiver does not check.

- o Require TOFU: sender MUST supply key and receiver MUST check, certificate not needed and ignored if sent.

- o Allow TOFU: sender must supply key and receiver MUST check, receiver SHOULD check certificate if supplied by sender.

- o Require PKI: sender must supply key and certificate, receiver must check both.

5. NEWKEY, Key Roll

Modern key management allows for agility in 'rolling' to a new key or even algorithm in case of key expiry, key compromise, or merely prudence. Declaring a new key with an L3DL OPEN PDU would cause serious churn in topology as a new OPEN may cause a withdraw of previously announced encapsulations. Therefore, a gentler rekeying is needed.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Type = 8   |          Payload Length       | New Key Type  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         New Key Length        |           New Key ...         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               |         New Cert Length       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      New Certificate ...                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Old Sig Type |      Old Signature Length     |               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+               +
|                       Old Signature ...                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The New Key Type, New Key Length, New Key, New Cert Length, and New Certificate field declare the replacement algorithm suite, key, and certificate.

The NEWKEY PDU is signed using the current (soon to be old) algorithm suite and key.

The sender and the receiver should be cautious of algorithm suite downgrade attacks.

To avoid possible race conditions, the receiver SHOULD accept signatures using either the new or old key for a configurable time (default 30 seconds). This is intended to accommodate situations such as senders with high peer out-degree and a single per-device asymmetric key.

If the sender does not receive an ACK in the normal window, including retransmission, then the sender MAY choose to allow a session reset by either issuing a new OPEN or by letting the receiver eventually have a signature failure (error code 3) on a PDU.

The rekeying operation changes the session key and algorithm suite described in Section 2.3. The NEWKEY PDU itself is verified using the old algorithm and session key, subsequent PDUs are verified with the new algorithm and session key recorded after the NEWKEY PDU has been accepted.

6. Security Considerations

The TOFU method requires a leap of faith to accept the key in the OPEN PDU, as it can not be verified against any authority. Hence it is jokingly referred to as Married On First Date. The assurance it does provide is that subsequent signed PDUs are from the same peer. And data integrity is a positive side effect of the signature covering the payload.

The PKI-based method offers assurance that the certificate, and hence the keying material, provided in the OPEN PDU are authorized by a central authority, e.g. the network's network security team. The onward assurance of talking to the same peer and data integrity are the same as in the TOFU method.

With the PKI-based method, automated device provisioning could restrict which certificates are allowed from which peers on a per interface basis. This would complicate key rolls. Where one draws the line between rigidity, flexibility, and security varies.

The REKEY PDU is open to abuse to create an algorithm suite downgrade attack.

7. IANA Considerations

This document requests the IANA create a new entry in the L3DL PDU Type registry as follows:

PDU Code	PDU Name
8	NEWKEY

This document requests the IANA add a registry entry for "TOFU - Trust On First Use" to the L3DL-Signature-Type registry as follows:

Number	Name
1	TOFU - Trust On First Use
2	PKI

8. Acknowledgments

The authors than Russ Housley for advice and review.

9. Normative References

[I-D.ietf-lsvr-l3dl]
Bush, R., Austein, R., and K. Patel, "Layer 3 Discovery and Liveness", draft-ietf-lsvr-l3dl-04 (work in progress), May 2020.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, DOI 10.17487/RFC5280, May 2008, <https://www.rfc-editor.org/info/rfc5280>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

Authors' Addresses

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

Email: randy@psg.com

Rob Austein
Arrcus, Inc.

Email: sra@hactrn.net

Network Working Group R. Bush
Internet-Draft Arrcus & IIJ
Intended status: Standards Track K. Patel
Expires: November 27, 2020 Arrcus
May 26, 2020

L3DL Upper Layer Protocol Configuration
draft-ietf-lsvr-l3dl-ulpc-00

Abstract

This document users the Layer 3 Liveness and Discovery protocol to communicate the parameters needed to exchange inter-device Upper Layer Protocol Configuration for upper layer protocols such as the BGP family.

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

Copyright Notice

Table of Contents

1. Introduction

Massive Data Centers (MDCs) which use upper layer protocols such as BGP4, BGP-LS, BGP-SPF, etc. may use the Layer 3 Liveness and Discovery Protocol, L3DP, [I-D.ietf-lsvr-l3dl] to reveal the inter-device links of the topology. It is desirable for devices to facilitate the configuration parameters of those upper layer protocols to enable more hands-free configuration. This document defines a new L3DP PDU to communicate these Upper Layer Protocol Configuration parameters.

2. Reading and Terminology

The reader is assumed to have read Layer 3 Discovery and Liveness [I-D.ietf-lsvr-l3dl]. The terminology and PDUs there are assumed here.

Familiarity with the BGP4 Protocol [RFC4271] is assumed. Familiarity with BGP-SPF, [I-D.ietf-lsvr-bgp-spf], might be useful.

3. Upper Layer Protocol Configuration PDU

To communicate parameters required to configure peering and operation of Upper Layer Protocols at IP layer 3 and above, e.g., BGP sessions on a link, a neutral sub-TLV based Upper Layer Protocol PDU is defined 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 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Type = 9   |                 Payload Length                ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~               |   ULPC Type   |   AttrCount   |               ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~       Attribute List ...      |    Sig Type   | Signature Len ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~               |                 Signature ...                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The Type and Payload Length are defined in [I-D.ietf-lsvr-l3dl].

ULPC Type: An integer denoting the type of the uper layer protocol

```
0     :  Reserved
1     :  BGP
2-255 :  Reserved
```

The AttrCount is the number of attribute sub-TLVs in the Attribute List.

The Attribute List is a, possibly null, set of sub-TLVs describing the configuration attributes of the specific upper layer protocol.

An Attribute consists of a one octet Attribute Type, a one octet Attribute Length of the number of octets in the Attribute, and a Payload of arbitrary length up to 253 octets.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Attr Type = 1 |    Attr Len   |            Payload            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

3.1. BGP ULPC Attribute sub-TLVs

The parameters needed for BGP peering on a link are exchanged in sub-TLVs within an Upper Layer Protocol PDU. The following describe the various sub-TLVs for BGP.

The goal is to provide the minimal set of configuration parameters needed by BGP OPEN to successfully start a BGP peering. The goal is specifically not to replace or conflict with data exchanged during BGP OPEN. Multiple sources of truth are a recipe for complexity and hence pain.

If there are multiple BGP sessions on a link, e.g., IPv4 and IPv6, then multiple sets of BGP sub-TLVs MAY BE exchanged within the BGP ULPC PDU or multiple BGP ULPC PDUs may be sent, one for each address family.

A peer receiving BGP ULPC PDUs has only one active BGP ULPC PDU for an particular address family at any point in time; receipt of a new BGP ULPC PDU for a particular address family replaces any previous one.

If there are one or more open BGP sessions, receipt of a new BGP ULPC PDU does not affect these sessions and the PDU SHOULD be discarded. If a peer wishes to replace an open BGP session, they must first close the running session and then send a new BGP ULPC PDU.

As a link may have multiple encapsulations and multiple addresses for an IP encapsulation, which address of which encapsulation is to be used for the BGP session MUST be specified.

For each BGP peering on a link here MUST be one agreed encapsulation, and the addresses used MUST be in the corresponding L3DP IPv4/IPv6 Announcement PDUs. If the choice is ambiguous, an Attribute may be used to signal preferences.

If a peering address has been announced as a loopback, i.e. MUST BE flagged as such in the L3DL Encapsulation PDU, a two or three hop BGP session will be established. Otherwise a direct one hop session is used. the BGP session to a loopback will forward to the peer's address which was marked as Primary in the L3DL Encapsulation Flags, iff it is in a subnet which is shared with both BGP speakers. If the primary is not in a common subnet, then the BGP speaker MAY pick a forwarding next hop that is in a subnet they share. If there are multiple choices, the BGP speaker SHOULD have signaled which subnet to choose in an Upper Layer Protocol Configuration PDU Attribute.

3.1.1. BGP ASN

The Autonomous System number MUST be specified. If the AS Number is less than 32 bits, it is padded with high order zeros.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Attr Type = 1 | Attr Len = 6  |             My ASN            ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

3.1.2. BGP IPv4 Address

The BGP IPv4 Address sub-TLV announces the sender's IPv4 BGP peering source address to be used by the receiver. At least one of IPv4 or IPv6 BGP source addresses MUST be announced.

As usual, the BGP OPEN capability negotiation will determine the AFI/SAFIs to be transported over the peering, see [RFC4760] .

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Attr Type = 2 | Attr Len = 7  |    My IPv4 Peering Address    ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                               |   Prefix Len  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

3.1.3. BGP IPv6 Address

The BGP IPv6 Address sub-TLV announces the sender's IPv6 BGP peering source address to be used by the receiver. At least one of IPv4 or IPv6 BGP source addresses MUST be announced.

As usual, the BGP OPEN capability negotiation will determine the AFI/SAFIs to be transported over the peering, see [RFC4760] .

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Attr Type = 3 | Attr Len = 19 |                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               +
|                                                               |
+                                                               +
|                     My IPv6 Peering Address                   |
+                                                               +
|                                                               |
+                               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               |   Prefix Len  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

3.1.4. BGP Authentication sub-TLV

The BGP Authentication sub-TLV provides any authentication data needed to OPEN the BGP session. Depending on operator configuration of the environment, it might be a simple MD5 key (see [RFC2385]), the name of a key chain a KARP database (see [RFC7210]), or one of multiple Authentication sub-TLVs to support hop[RFC4808].

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Attr Type = 4 |    Attr Len   |                               ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               ~
~                   BGP Authentication Data ...                 ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

3.1.5. BGP Miscellaneous Flags

The BGP session OPEN has extensive, and a bit complex, capability negotiation facilities. In case one or more extra attributes might be needed, the BGP Miscellaneous Flags sub-TLV may be used. No flags are currently defined.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Attr Type = 5 | Attr Len = 4  |          Misc Flags           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Misc Attrs:

Bit 0:   Ghu knows what
Bit 1-15: Must be zero

4. Security Considerations

All the Security considerations of [I-D.ietf-lsvr-l3dl] apply to this PDU.

As the ULPC PDU may contain keying material, see Section 3.1.4, it SHOULD BE signed.

Any keying material in the PDU SHOULD BE salted and hashed.

The BGP Authentication sub-TLV provides for provisioning MD5, which is a quite weak hash, horribly out of fashion, and kills puppies. But, like it or not, it has been sufficient against the kinds of attacks BGP TCP sessions have endured. Soit is what BGP deployments use.

5. IANA Considerations

This document requests the IANA create a new entry in the L3DL PDU Type registry as follows:

PDU Code	PDU Name
9	ULPC

This document requests the IANA create a registry for L3DL ULPC Type, which may range from 0 to 255. The name of the registry should be L3DL-ULPC-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:

Value	Name
0	Reserved
1	BGP
2-255	Reserved

6. References

6.1. Normative References

[I-D.ietf-lsvr-l3dl]
Bush, R., Austein, R., and K. Patel, "Layer 3 Discovery and Liveness", draft-ietf-lsvr-l3dl-04 (work in progress), May 2020.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC4271] Rekhter, Y., Ed., Li, T., Ed., and S. Hares, Ed., "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, DOI 10.17487/RFC4271, January 2006, <https://www.rfc-editor.org/info/rfc4271>.

[RFC4760] Bates, T., Chandra, R., Katz, D., and Y. Rekhter, "Multiprotocol Extensions for BGP-4", RFC 4760, DOI 10.17487/RFC4760, January 2007, <https://www.rfc-editor.org/info/rfc4760>.

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", RFC 5226, DOI 10.17487/RFC5226, May 2008, <https://www.rfc-editor.org/info/rfc5226>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

6.2. Informative References

[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-09 (work in progress), May 2020.

[RFC2385] Heffernan, A., "Protection of BGP Sessions via the TCP MD5 Signature Option", RFC 2385, DOI 10.17487/RFC2385, August 1998, <https://www.rfc-editor.org/info/rfc2385>.

[RFC4808] Bellovin, S., "Key Change Strategies for TCP-MD5", RFC 4808, DOI 10.17487/RFC4808, March 2007, <https://www.rfc-editor.org/info/rfc4808>.

[RFC7210] Housley, R., Polk, T., Hartman, S., and D. Zhang, "Database of Long-Lived Symmetric Cryptographic Keys", RFC 7210, DOI 10.17487/RFC7210, April 2014, <https://www.rfc-editor.org/info/rfc7210>.

Authors' Addresses

Randy Bush
Arrcus & IIJ
5147 Crystal Springs
Bainbridge Island, WA 98110
US

Email: randy@psg.com

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

Email: keyur@arrcus.com