Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: May 29, 2020

X. Xu
Alibaba Inc
K. Talaulikar
Cisco Systems
K. Bi
Huawei
J. Tantsura
Apstra
N. Triantafillis
Amazon Web Services

November 26, 2019

BGP Neighbor Discovery
draft-xu-idr-neighbor-autodiscovery-12

Abstract

BGP is being used as the underlay routing protocol in some large-scaled data centers (DCs). Most popular design followed is to do hop-by-hop external BGP (EBGP) session configurations between neighboring routers on a per link basis. The provisioning of BGP neighbors in routers across such a DC brings its own operational complexity.

This document introduces a BGP neighbor discovery mechanism that greatly simplifies BGP operations in such DC and other networks by automatic setup of BGP sessions between neighbor routers using this mechanism.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

BGP is being used as the underlay routing protocol instead of link-state routing protocols like IS-IS and OSPF in some large-scale data centers (DCs). [RFC7938] describes the design, configuration and operational aspects of using BGP in such networks. The most popular design scheme involves the setup of external BGP (EBGP) sessions over individual links between directly connected routers using their interface addresses. Such BGP neighbor provisioning requires configuration of the neighbor IP address and Autonomous System (AS) Number (ASN) for BGP neighbor on each and every link of every BGP router. As a DC fabric comprising of topology described in [RFC7938] grows with addition of new leafs, spines, and links between them, the BGP provisioning needs to be carefully updated. Unlike with the link-state protocols, in the case of BGP, there is no automatic discovery of neighbors and route exchange between them by simply adding links and nodes of the fabric into the routing protocol operation.

In some DC designs with BGP, multiple links are added between a leaf and spine to add additional bandwidth. Use of link-aggregation at Layer 2 level may not be always desirable in such cases due to the risk of flow polarization on account of a mix of ECMP at Layer 2 and Layer 3 levels. In such cases, one option is for EBGP sessions to be setup between two BGP neighbors over each of the links between them. In such a case, the BGP session scale and the resultant increase in update processing may pose scalability challenges. A second option is for a single EBGP session to be setup between the loopback IP addresses between the neighbor and then configure some static routes for loopback reachability over the underlying links. This option introduces an additional provisioning task for the static routes.

Furthermore, there is also a need for BGP to be able to describe its links and its neighbors on its directly connected links and export this information via BGP-LS [RFC7752] to provide a detail link-level topology view of a data center running BGP. The ability of BGP in discovering its neighbors over its links, monitoring their liveliness and learning the link attributes (such as addresses) is required for the conveying the link-state topology in such a BGP network. This information can be leveraged by the BGP-SPF proposal [I-D.ietf-lsvr-bgp-spf] which introduces link-state routing capabilities in BGP. This information can also be leveraged to convey the link-state topology in a network running traditional BGP routing using BGP-LS as described in [I-D.ketant-idr-bgp-ls-bgp-only-fabric] and to enabled end to end traffic engineering use-cases spanning across DCs and the core/access networks.

1.1. 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. Terminology

This document makes use of the terms defined in [RFC4271] and [RFC7938] .

3. Applicability

The applicability of the BGP Neighbor Discovery mechanism described in this document is limited to deployments where BGP is used as routing protocol between directly connected routers and when there is a requirement for automatic setup of BGP peering between them.

- In DC networks where BGP is used as a hop-by-hop routing protocol [RFC7938].

- In metro networks where access aggregation topologies are architected as a CLOS topology (or similar other networks) and BGP is used as a hop-by-hop routing protocol.

While this document uses EBGP examples, the mechanism is equally applicable in designs that use IBGP similarly for hop-by-hop routing.

The applicability of the BGP Neighbor Discovery mechanism to any other BGP protocol deployment is outside the scope of this document.

4. Requirements

This section describe the requirements for the BGP hop-by-hop routing deployments that were considered for the definition of the BGP Neighbor Discovery extensions proposed in this document..

Following are the key requirements related for the BGP neighbor discovery process:

1. It should perform discovery of directly connected BGP routers. Mechanism should support either IPv4 or IPv6 or a dual stack design and it should be generic for any link-layer.

2. It should include exchange of BGP peering addresses (IPv4 or IPv6 or both) that routers can use to automatically setup BGP TCP peering between themselves. The mechanism should leverage the existing capability negotiation process performed as part of the BGP TCP session establishment.

3. When BGP peering is desired to be performed over loopback addresses of the routers, then the mechanism should automatically setup reachability to the loopback over one or more underlying directly connected links between them. In this scenario, the mechanism should also provide resolution for the BGP next-hop address (i.e. the loopback address) for the BGP routes exchanged over these sessions between the loopback addresses.

4. Mechanism should enable exchange of link-level information such as IP addresses and link attributes between the directly connected BGP routers. It should be extensible to include other information in the future.

5. Mechanism should be limited to link scope for security and use link-local addressing only. Cryptographic mechanisms should be also provided for additional security.

6. Mechanism should support capabilities for performing optional validation of parameters to detect misconfiguration (e.g. link address subnet mismatch, peering between incorrect AS, etc.) in an extensible manner before going on to use the link and the setup of the BGP TCP peering session over it.

7. The mechanism should not affect or change the BGP TCP session establishment procedures and the BGP routing exchange over the TCP session other than the interactions for triggering the setup/ removal of peer session that is based on discovery mechanism.

8. The mechanism should leverage existing fast-detection techniques for failures that are used currently for EBGP sessions over directly connected links like fast-external-failover and BFD.

9. The mechanism should focus on the discovery process and exchange of status as a control plane procedure and be sufficiently loosely coupled with the base BGP operations to enable

implementations to ensure scalability of BGP operations when using the discovery procedures.

5. Overview

At a high level, this specification introduces the use of UDP based BGP Hello messages to be exchanged between directly connected BGP routers for neighbor discovery.

1. Information is exchanged between BGP routers on a per link basis leading to discovery of each others peering address and other information.

2. The TCP session establishment for the BGP protocol operation and the BGP routing exchange over these sessions can then follow without any change/modification from the existing BGP protocol operations as specified in [RFC4271].

3. As part of the neighbor information exchange the route to a neighbor’s peering address is also automatically setup pointing over the links over which the neighbor is discovered.

4. This route is used for both the BGP TCP session establishment as well as for resolution of the BGP next-hop (NH) for the routes learnt via the neighbor instead of an underlying IGP or static route.

This document prefers the use of an extension to BGP protocol since the deployments and use-cases targeted (i.e. large-scale DCs) are already running BGP as their routing protocol. Extending BGP with neighbor discovery capabilities is operationally and implementation wise a simpler approach than requiring a new or an additional protocol to be first extended to do this functionality (to exchange BGP-specific parameters) and then also integrated its operations with BGP protocol operations.

The BGP Neighbor discovery mechanism is a control plane mechanism intended to discovery and maintain the BGP router’s adjacencies with its neighbors over directly connected links. Maintaining an adjacency also involves detecting any changes in parameters using periodic messages and triggering corresponding actions based on the change. Such actions also include removal of the BGP TCP peering for an auto discovered peering session based on the neighbor discovery. However, the mechanism is not intended for a fast liveness detection of neighbor and existing mechanisms for this purpose such as BFD [RFC5880] may be leveraged.

The BGP Neighbor discovery mechanism is scoped to a link and works using link-local addressing. In a BGP DC network that is using IPv6 in the fabric underlay, it is possible that no IPv6 global addresses are assigned to the interfaces between the nodes and the IPv6 Global address(es) are assigned only to the loopback interfaces of these nodes. The Neighbor discovery mechanism enables the setup of BGP peering using the IPv6 Global addresses on the loopback interfaces and hop by hop routing with just IPv6 link-local addresses on the interfaces. Such a design eases introduction of nodes in the fabric and links between them from a provisioning aspect. In a deployment with IPv4 addressing, IP unnumbered could be similarly used for all the links between the nodes using the IPv4 address assigned to the loopback interfaces on those nodes.

The BGP neighbor discovery mechanism defined in this document borrows ideas from the Label Distribution Protocol (LDP) [RFC5036]. However, most importantly, only the concept of link-local signaling based neighbor discovery is borrowed while the discovery aspect for targeted LDP sessions does not apply to this BGP neighbor discovery mechanism.

The further sections in this document first describe the newly introduced message formats and TLVs and then go on to describe the procedures of BGP neighbor discovery and its integration with the base BGP protocol mechanism as specified in [RFC4271].

The operational and management aspects of the BGP neighbor discovery mechanism are described in Section 12.

6. UDP Message Header

The BGP neighbor discovery mechanism will operate using UDP messages. The UDP port of TBD (179 is the preferred port number to be assigned as specified in Section 13) is used which is same as the TCP port 179 used by BGP. The BGP UDP message common header format is specified 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Version   |      Type     |       Message Length          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          AS number                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        BGP Identifier                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 1: BGP UDP Message Header

Version: This 1-octet unsigned integer indicates the protocol version number of the message. The current BGP version number is 4.

Type: The type of BGP message

Message Length: This 2-octet unsigned integer specifies the length in octets of the entire BGP UDP message including the header.

AS number: AS Number of the UDP message sender.

BGP Identifier: BGP Identifier of the UDP message sender.

BGP UDP messages can be sent using either IPv4 or IPv6 depending on the address used for session establishment and provisioned on the interfaces over which these messages are sent.

7. Hello Message Format

A BGP router uses UDP based Hello messages to discover directly connected BGP neighbors over those interfaces enabled for Neighbor Discovery. The BGP Hello messages for the Neighbor Discovery procedure are used for link-locally signaling and hence MUST be addressed to the "all routers on this subnet" group multicast address (i.e., 224.0.0.2 in the IPv4 case and FF02::2 in the IPv6 case) and the TTL for the IP packets SHOULD be set to 1. The IP source address MUST be set to the address of the interface over which the message is sent out which would be the primary interface address or unnumbered address in the IPv4 case and the IPv6 link-local address on the interface in the IPv6 case.

The Hello message format 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
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |     Version   |     Type      |      Message Length           |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                           AS number                           |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                         BGP Identifier                        |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |     Adjacency Hold Time       |    Flags      |   Reserved    |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                             TLVs                              |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2: BGP Hello Message

Version: This 1-octet unsigned integer indicates the protocol version number of the message. The current BGP version number is 4.

Type: The type of BGP message (Hello - TBD value from BGP Message Types Registry)

Message Length: This 2-octet unsigned integer specifies the length in octets of the TLVs field.

AS number: AS Number of the BGP router sending the Hello message.

BGP Identifier: BGP Identifier of the BGP router sending the Hello message.

Adjacency Hold Time: Hello adjacency hold timer in seconds. Adjacency Hold Time specifies the time, for which the receiving BGP neighbor router SHOULD maintain adjacency state for it, without receipt of another Hello. A value of 0 means that the receiving BGP peer should immediately mark that the adjacency to the sender is going down.

Flags : Current defined bits are as follows. All other bits SHOULD be cleared by sender and MUST be ignored by receiver.

```
 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+
|S|             |
+-+-+-+-+-+-+-+-+
```

where:

S bit - indicates that this is a State Change Hello message when SET and normal periodic Hello message when CLEAR

Reserved: SHOULD be set to 0 by sender and MUST be ignored by receiver.

TLVs: This field contains one or more TLVs as described below.

BGP HELLO messages can be sent using either IPv4 or IPv6 addresses depending on the addressing used for session establishment and provisioned on the interfaces over which these messages are sent. When both IPv4 and IPv6 is enabled on the interface, then IPv6 address SHOULD be used. Implementations MAY provide an option to override the choice of address family to be used. The choice of address family to be used MUST be consistent on all BGP routers on a given link for neighbor discovery.

Based on the setting of the S flag, there are two variants of the Hello message:

1. State Change Hello Message : these Hello messages include TLVs which convey the state and parameters of the local interface and adjacency to other routers on the link. They are generated only when there is a change in state of the adjacency or some parameter at the interface level.

2. Periodic Hello Message : these are the normal periodic Hello messages which do not include TLVs and are used to maintain the adjacency on the link during steady state conditions.

These Hello message variants are intended to limit the exchange of information and state via TLVs to only those periods where necessary while using lightweight Hello messages during steady state. This simplifies the Hello message processing and improves scalability of the discovery mechanism.

The neighbor discovery procedure using the Hello message is described in Section 9 and its relation with the BGP Keepalives and Hold Timer for the TCP session is described in Section 10.

8. Hello Message TLVs

The BGP Hello message carries TLVs as described in this section that enable exchange of information on a per interface basis between directly connected BGP neighbors. These messages enable the neighbor discovery process.

8.1. Accepted ASN List TLV

The Accepted ASN List TLV is an optional TLV that is used to signal an unordered list of AS numbers from which the BGP router would accept BGP sessions. When not signaled, it indicates that the router will accept BGP peering from any ASN from its neighbors. Indicating the list of ASNs, helps avoid the neighbor discovery process getting stuck in a 1-way state where one side keeps attempting to setup adjacency while the other does not accept it due to incorrect ASN.

The operational and management aspects of this ASN based policy control for BGP neighbor discovery are described further in Section 12.

This TLV SHOULD NOT be included in a Hello message with the S bit CLEAR. More than a single instance of this TLV MUST NOT be included in a Hello message. If a router receives multiple instances of this

TLV then it should only consider the first instance in the sequence and ignore the rest.

The format of this TLV is shown below

```
 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                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                 Accepted ASN List(variable)                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 3: Accepted ASN List TLV

Type: TBD1

Length: Specifies the length of the Value field in octets (in multiple of 4)

Accepted ASN-List: This variable-length field contains one or more accepted 4-octet ASNs.

8.2. Peering Address TLV

The Peering Address TLV is used to indicate to the neighbor the address to be used for setting up the BGP TCP session. Along with the peering address, the router can specify its supported AFI/SAFI(s). When the AFI/SAFI values are specified as 0/0, then it indicates that the neighbor can attempt for negotiation of any AFI/SAFIs. The indication of AFI/SAFI(s) in the Peering Address TLV is not intended as an alternative for the MP capabilities negotiation mechanism done as part of the BGP TCP session establishment.

Multiple instances of this TLV MAY be included in the Hello message, one for each peering address (e.g. IPv4 and IPv6 or multiple IPv4 addresses for different AFI/SAFI sessions). When multiple peering addresses are provisioned, then the indication helps the router select the appropriate peer address of the neighbor based on its local peering address profile by matching the supported AFI/SAFIs.

This TLV is essential for the setting up of the TCP peering between BGP neighbors using the neighbor discovery mechanism. When a BGP router stops including a Peer Address in its State Change Hello messages, then it is no longer accepting TCP peering sessions to that address and the neighbor SHOULD clean up any peering session that was setup to that address via the discovery mechanism.

Implementations SHOULD support the signaling of an interface IP address in the Peering Address TLV and perform the BGP TCP session establishment using interface addresses (i.e. the neighbor discovery mechanism is not limited to the use of loopback addresses for the peering session establishment). Implementations MAY support the signaling of IPv6 Link Local addresses using the Peering Address TLV and using the same for the BGP TCP session setup.

This TLV SHOULD NOT be included in a Hello message with the S bit CLEAR.

The Peering Address TLV format is shown below.

```
     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                  |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |   Flags       | No. AFI/SAFI  |       Reserved                |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |      Address (4-octet or 16-octet)                            |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |            AFI                |    SAFI       |  ...
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |      sub-TLVs ...
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 4: Peering Address TLV

Type: TBD2

Length: Specifies the length of the Value field in octets.

Flags : Current defined bits are as follows. All other bits SHOULD be cleared by sender and MUST be ignored by receiver.

```
    0 1 2 3 4 5 6 7
   +-+-+-+-+-+-+-+-+
   |A|             |
   +-+-+-+-+-+-+-+-+
```

where:

A bit - address is IPv6 when SET and IPv4 when CLEAR

Number of AFI/SAFI: indicates the number of AFI/SAFI pairs that the router supports on the given peering address.

Reserved: sender SHOULD set to 0 and receiver MUST ignore.

Address: This 4 or 16 octet field indicates the IPv4 or IPv6 address which is used for establishing BGP sessions.

AFI/SAFI : one or more pairs of these values that indicate the supported capabilities on the peering address.

Sub-TLVs : optional and currently none defined

8.3. Local Prefix TLV

BGP neighbor discovery mechanism, in certain scenarios, requires a BGP router to program a route in its local routing table for a prefix belonging to its neighbor router. On such scenario is when the BGP TCP peering is to be setup between the loopback addresses on the neighboring routers. This requires that the routers have reachability to their each other’s loopback addresses before the TCP session can be brought up.

The Local Prefix TLV is an optional TLV which enables a BGP router to explicitly signal its local prefix to its neighbor for setting up of such a local routing entry pointing over the underlying link over which it is being signaled. This enables the BGP router to have control over the specific links over which its neighbor that may reach it for the specific local prefix. The details of the procedure for programming of the route corresponding to the prefix signaled using the Local Prefix TLV is described in Section 9.3..

Multiple instances of the Local Prefix TLV MAY be included in the Hello message with each carrying a specific prefix in it. This TLV SHOULD NOT be included in a Hello message with the S bit CLEAR.

The Local Prefix TLV format is as shown below.

```
     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                   |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |   Flags       | Prefix Length |      Reserved                 |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |   Prefix Address (4-octet or 16-octet)                        |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |  sub-TLVs ...
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 5: Local Prefix TLV

Type: TBD3

Length: Specifies the length of the Value field in octets

Flags : Current defined bits are as follows. All other bits SHOULD be cleared by sender and MUST be ignored by receiver.

```
    0 1 2 3 4 5 6 7
   +-+-+-+-+-+-+-+-+
   |A|             |
   +-+-+-+-+-+-+-+-+
```

where:

A bit - address is IPv6 when SET and IPv4 when CLEAR

Prefix Length: specifies the Prefix length

Reserved: sender SHOULD set to 0 and receiver MUST ignore.

Prefix Address: This 4 or 16 octet field indicates the IPv4 or IPv6 prefix address.

Sub-TLVs : optional and currently none defined

8.4. Link Attributes TLV

The Link Attributes TLV is a mandatory TLV in a State Change Hello message that signals to the neighbor the link attributes of the interface on the local router. One and only one instance of this TLV MUST be included in the State Change Hello message. A State Change Hello message without this TLV included MUST be discarded and an error logged for the same.

This TLV enables a BGP router to learn all its neighbors IP addresses on the specific link as well as it’s link identifier. When the interface is IPv4 enabled, all the IPv4 addresses configured on it are included in this TLV. IPv4 unnumbered address is not included in this TLV and no IPv4 address would be included for the interface in such cases. When the interface is IPv6 enabled, all the IPv6 global addresses configured on the interface are included in this TLV. IPv6 link-local addresses are not included in this TLV. In case of an interface running dual stack, both IPv4 and IPv6 addresses are included in this TLV irrespective of the address family that is used for UDP message exchange.

Additional sub-TLVs may be defined in the future to exchange other link attributes between BGP neighbors. This TLV SHOULD NOT be included in a Hello message with the S bit CLEAR.

The Link Attributes TLV format is as shown below.

```
 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                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      Local Interface ID       |      Flags    |   Reserved    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    No. of IPv4 Addresses      |      No. of IPv6 Addresses    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   IPv4 Interface Address                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Prefix Mask  | ...
+-+-+-+-+-+-+-+-+

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                IPv6 Global Interface Address                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Prefix Mask  | ...
+-+-+-+-+-+-+-+-+

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  sub-TLVs ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 6: Link Attributes TLV

Type: TBD4

Length: Specifies the length of the Value field in octets

Local Interface ID : the local interface ID of the interface (refer unnumbered link section of [RFC2104] e.g. the MIB-2 ifIndex). This helps uniquely identify the link even when there are multiple links between two neighbors using IPv4 unnumbered address or only having IPv6 link-local addresses.

Flags : Currently defined bits are as follows. Other bits SHOULD be cleared by sender and MUST be ignored by receiver.

```
 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+
|I|V|B|         |
+-+-+-+-+-+-+-+-+
```

where:

I bit - indicates link is enabled for IPv4

V bit - indicates link is enabled for IPv6

B bit - indicates support for BFD monitoring [RFC5880] over the link

Reserved: SHOULD be set to 0 by sender and MUST be ignored by receiver.

No. of IPv4 Addresses : specifies the number of IPv4 addresses on the interface. When value is 0, then it indicates no IPv4 Prefixes are present or the interface is IPv4 unnumbered if it is enabled for IPv4

No. of IPv6 Addresses : specifies the number of IPv6 global addresses on the interface. When value is 0, then it indicates no IPv6 Global Prefixes are present and the interface is only configured with IPv6 link-local addresses if it is enabled for IPv6.

IPv4 Address & Mask: Zero or more pairs of IPv4 address and their mask.

IPv6 Address & Mask: Zero or more pairs of IPv6 address and their mask.

Sub-TLVs : optional and currently none defined

8.5. Neighbor TLV

The Neighbor TLV is used by a BGP router to indicate its Hello adjacency state with its neighboring router(s) on the specific link. The neighbor is identified by its AS Number and BGP Identifier. The router MUST include the Neighbor TLV for each of its discovered neighbors on that link irrespective of its status.

The usage of the Neighbor TLV is described in detail in Section 9. This TLV SHOULD NOT be included in a Hello message with the S bit CLEAR.

The Neighbor TLV format is as shown below.

```
  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                  |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |   Flags       |  State        |       Reserved                |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                      Neighbor AS number                       |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                   Neighbor BGP Identifier                     |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |  sub-TLVs ...
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 7: Neighbor TLV

Type: TBD5

Length: Specifies the length of the Value field in octets

Flags : Current defined bits are as follows. All other bits SHOULD be cleared by sender and MUST be ignored by receiver.

```
 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+
|B|             |
+-+-+-+-+-+-+-+-+
```

where:

B bit - When SET with the adjacency state not in Accepted state indicates that the adjacency is not accepted due to BFD down.

State : Indicates the state code of the adjacency state machine (refer to Section 9.2 for details) for the neighbor over this link. The following codes are currently defined

0 - Down (not to be used as state in this TLV

1 - Initial (not to be used as state in this TLV)

2 - 1-way

3 - 2-way

4 - Adj-Reject

5 - Adj-OK

6 - Accepted

Reserved: SHOULD be set to 0 by sender and MUST be ignored by receiver.

Neighbor AS number: AS Number of the neighbor BGP router as signaled in its Hello message.

Neighbor BGP Identifier: BGP Identifier of the neighbor BGP router as signaled in its Hello message.

Sub-TLVs : currently none defined

8.6. Cryptographic Authentication TLV

The Cryptographic Authentication TLV is an optional TLV that is used as part of an authentication mechanism for BGP Hello message by securing against spoofing attacks. It also introduces a cryptographic sequence number carried in the Hello messages that can be used to protect against replay attacks. Using this Cryptographic Authentication TLV, one or more secret keys (with corresponding Security Association (SA) IDs) are configured on each BGP router. For each BGP Hello message, the key is used to generate and verify an HMAC Hash that is stored in the Cryptographic Authentication TLV. For the cryptographic hash function, this document proposes to use SHA-1, SHA-256, SHA-384, and SHA-512 defined in US NIST Secure Hash Standard (SHS) [FIPS-180-4]. The HMAC authentication mode defined in [RFC2104] is used. Of the above, implementations MUST include support for at least HMAC-SHA-256, SHOULD include support for HMAC-SHA-1, and MAY include support for HMAC-SHA-384 and HMAC-SHA-512.

Further details for ensuring the security of the BGP Hello UDP messages are described in Section 11.

The Cryptographic Authentication TLV format is as shown below.

```
 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                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   Security Association ID                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       Cryptographic Sequence Number (High-Order 32 Bits)      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       Cryptographic Sequence Number (Low-Order 32 Bits)       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                Authentication Data (Variable)                //
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 8: Cryptographic Authentication TLV

Type: TBD6

Length: Specifies the length of the Value field in octets

Security Association ID: The 32-bit field that maps to the authentication algorithm and the secret key used to create the message digest carried in Hello message payload.

Cryptographic Sequence Number: The 64-bit, strictly increasing sequence number that is used to guard against replay attacks. The 64-bit sequence number MUST be incremented for every BGP Hello message sent by the BGP router. Upon reception, the sequence number MUST be greater than the sequence number in the last BGP Hello message accepted from the sending BGP neighbor. Otherwise, the BGP hello message is considered a replayed packet and is dropped. The Cryptographic Sequence Number is a single space per BGP router.

Authentication Data: This field carries the digest computed by the Cryptographic Authentication algorithm in use. The length of the Authentication Data varies based on the cryptographic algorithm in use, which is shown below:

HMAC-SHA1 20 bytes

HMAC-SHA-256 32 bytes

HMAC-SHA-384 48 bytes

HMAC-SHA-512 64 bytes

9. Neighbor Discovery Procedure

The neighbor discovery mechanism in BGP is implemented with the introduction of an Interface state in BGP and an Adjacency Finite State Machine (FSM). This section describes the states, FSM and procedures involved.

9.1. Interface Procedures

In order to perform neighbor discovery, BGP needs to maintain state for the subset of its connected interfaces over which neighbor discovery is enabled. For these interfaces, BGP sends its Hello messages, including the TLVs described in Section 8, as long as its link is UP. The Neighbor TLV described in Section 8.5 is, included once a neighbor is discovered as described in Section 9.2 .

The Hello messages MUST be originated periodically at an interval which is less than or equal to one third of the Adjacency Hold Time indicated by the router in its Hello message. The RECOMMENDED default value for the Adjacency Hold Time is 45 seconds which makes the hello message interval to be 15 seconds. Period Hello messages ensure robustness of the neighbor discovery mechanism against transient loss of hello messages that are sent over unreliable UDP messaging channel and also enable detection of neighbor down events over specific links. Periodic Hello messages that do not convey any change in state SHOULD exclude TLVs that signal the local interface or adjacency state and have the S bit CLEAR as specified in Section 7.

A State Change Hello message MUST be triggered, without waiting for the periodic timer expiry, whenever there is a change in the router's Hello TLVs' content that needs to be signaled to its neighbor over the specific link. A State Change Hello message MUST also be triggered when a new neighbor's Hello message is first received or change is detected in the neighbor's Hello TLV's that results in change in it's adjacency state. Once a State Change Hello message is triggered on a specific interface, the router MUST continue to generate State Change Hello messages on it with the necessary TLVs included at periodic hello message intervals for a period of time that is at least equal to the Adjacency Hold Time. This ensures that messages carrying the updated information and local state changes are not lost. The router can switch back to Periodic Hello messages after it has transmitted State Change Hello messages with the latest TLV contents for the Adjacency Hold Time period.

When a router receives a Hello message from its neighbor, it MUST restart the Adjacency Hold timer that it is maintaining for the neighbor adjacency using the value indicated in the Hello message. When the message is of type State Change (i.e. with S bit SET), it additionally needs to process all the TLVs included and verify the signaled state against what was conveyed in the previous State Change Hello message from the same neighbor. Any changed identified would trigger the adjacency FSM change as described in Section 9.2.

When a router does not receive a Hello message from its neighbor for a period equal to Adjacency Hold Time, then it MUST treat this as an adjacency down event and clean up its adjacency state to this neighbor as described in Section 9.2.

Before the interface is shut or the neighbor discovery mechanism is disabled on it, the router SHOULD attempt to send out immediate Hello messages, with the S bit CLEAR (i.e. not including state related TLVs) and with Adjacency Hold Time set to 0, to trigger the adjacency down event on its neighbors. It MUST then clean up its own adjacency states on that specific link.

When either the BGP Identifier or the AS number are modified, then the router MUST send out a triggered Hello message, with the S bit CLEAR and with Adjacency Hold Time set to 0 using the old BGP Identifier and AS number values, over all the links enabled for BGP neighbor discovery.

A router receiving a Hello message with Adjacency Hold Time set to 0 MUST treat this event as if the adjacency hold timer has expired for the specific neighbor and proceed to bring down the adjacency.

An interface going down (e.g. due to link failure or loss of signal) MUST immediately trigger the adjacency down event for all adjacencies over it as if the adjacency hold timer expired for all neighbors on that link.

9.2. Adjacency State Machine

On a per interface basis, BGP needs to maintain an adjacency state for each neighbor that it discovers. The adjacency state is maintained as a FSM and it has states as described in the following sections.

9.2.1. Down State

This is the transient terminal state after which an adjacency is deleted.

When transitioning to the Down state from Accepted, the router removes the path corresponding to this adjacency from any Adjacency Route that it had setup to the neighbor's prefixes. If no other adjacency exists in Accepted state to the neighbor, then it also deletes the BGP TCP peering session(s) setup to the neighbor based on the neighbor discovery mechanism.

9.2.2. Initial State

This is the transient initial state from which an adjacency starts, when the router detects a hello message from a new neighbor on the link, and immediately transitions to the 1-way state.

9.2.3. 1-Way State

While in the 1-way state (or when entering it), the adjacency transitions from 1-way to 2-way state when the router detects a Neighbor TLV corresponding to itself in the neighbor's Hello message. If the state does not immediately transition on to 2-way after entering 1-way, the the router MUST immediately trigger a State Change Hello message with the inclusion of the neighbor in a Neighbor TLV with the state set to 1-way.

When transitioning to the 1-way state from Accepted, the router removes the path corresponding to this adjacency from any Adjacency Route that it had setup to the neighbor's prefixes. If no other adjacency exists in Accepted state to the neighbor, then it also deletes the BGP TCP peering session(s) setup to the neighbor based on the neighbor discovery mechanism.

Adjacency transitions to Down state for any of the following events:

- Link goes down operationally or is administratively shut
- Adjacency Hold Timer expires
- Router receives a Hello message from its neighbor with Adjacency Hold Time value set to 0
- Neighbor discovery is disabled on the link
- Change in BGP Identifier or AS number on the local router

9.2.4. 2-Way State

Upon transitioning into this state, the router triggers a State Change Hello message with the neighbor's status set to 2-way in the Neighbor TLV. At this stage, both neighbors have received each other's Hello messages and thus discovered each other.

When the router, in this adjacency state, detects that the neighbor's state for itself is 2-way or higher, then it performs the validation checks based on local policy and information exchanged in the Hello TLVs. Following are some of the validation checks that may be performed on the adjacency:

- Verify subnet matching between the local and remote interface addresses.

- Verify AS numbers based on local policy as well as against the Allowed ASN TLV when one is being exchanged.

- Verify that BFD monitoring (when enabled) is indicating UP state.

When the adjacency passes the validation checks, it transitions to the Adj-OK state and transitions to the Adj-Reject state otherwise.

The adjacency transitions to Down state for any of the adjacency down events described in Section 9.2.3 .

The adjacency transitions to 1-way state when the router stops seeing itself in a Neighbor TLV of its Neighbor's State Change Hello messages.

9.2.5. Adj-Reject State

Upon transitioning into this state, the router triggers a State Change Hello message with the neighbor's status set to Adj-Reject in the Neighbor TLV.

The adjacency remains in the Adj-Reject state as long as the parameters being exchanged via the State Change Hello messages do not pass validation checks. The neighbors continue to include each other in their respective State Change Hello messages.

The adjacency transitions to the Adj-OK state once the validation checks pass (e.g. due to update in any parameters or local policy).

The adjacency transitions to Down state for any of the adjacency down events described in Section 9.2.3 .

The adjacency transitions to 1-way state when the router stops seeing itself in a Neighbor TLV of its Neighbor’s State Change Hello messages.

When transitioning to an Adj-Reject state from Accepted state, the router removes the path corresponding to this adjacency from any Adjacency Route that it had setup to the neighbor’s prefixes. If no other adjacency exists in Accepted state to the neighbor, then it also deletes the BGP TCP peering session(s) setup to the neighbor based on the neighbor discovery mechanism.

9.2.6. Adj-OK State

Upon transitioning into this state, the router triggers a State Change Hello message with the neighbor’s status set to Adj-OK in the Neighbor TLV.

The adjacency transition to Adj-OK state indicates that the router has accepted its neighbor. However, it is possible that the neighbor has not accept it and is signaling Adj-Reject state for the adjacency from it’s end.

The adjacency transitions to the Accepted state from Adj-OK once it detects that its neighbor is also signaling the Adj-OK or Accepted state for it.

The adjacency transitions to Down state for any of the adjacency down events described in Section 9.2.3 .

The adjacency transitions to 1-way state when the router stops seeing itself in a Neighbor TLV of its Neighbor’s State Change Hello messages.

The adjacency transitions to Adj-Reject state when any of the validation checks listed in Section 9.2.4 fail.

When transitioning to an Adj-OK state from Accepted state, the router removes the path corresponding to this adjacency from any Adjacency Route that it had setup to the neighbor’s prefixes. If no other adjacency exists in Accepted state to the neighbor, then it also deletes the BGP TCP peering session(s) setup to the neighbor based on the neighbor discovery mechanism.

9.2.7. Accepted State

The adjacency transition to Accepted state indicates that both the neighboring routers have accepted the adjacency to each other.

On this transition, the router triggers a State Change Hello message with the neighbor's status set to Accepted in the Neighbor TLV. It then installs the Adjacency Route(s) for the Prefix(es) signaled by the neighbor via the Local Prefix TLV via this adjacency link using the neighbor's address on that link. If this is the first Accepted adjacency to the neighbor then the Adjacency Route gets added to the local routing table, otherwise an additional path corresponding to this adjacency link and neighbor address on it gets added to the existing Adjacency Route. The details are described in Section 9.3.

When this is the first Accepted adjacency to the neighbor, then the setup of the BGP TCP session to the Peering Address(es) signaled by the neighbor is also triggered.

The adjacency transitions to Down state for any of the adjacency down events described in Section 9.2.3.

The adjacency transitions to 1-way state when the router stops seeing itself in a Neighbor TLV of its Neighbor's State Change Hello messages.

The adjacency transitions to Adj-Reject state when any of the validation checks listed in Section 9.2.4 fail.

9.3. Adjacency Route

The Adjacency Route programming is an optional part of the BGP Neighbor Discovery mechanism for setting up reachability for the neighbor's prefixes signaled via the Local Prefix TLV corresponding to adjacencies in Accepted state.

Adjacency Routes establish reachability between local prefixes on directly connected BGP routers. They enable reachability between the Peering Addresses (generally loopbacks) of the two neighbors so that the BGP TCP session may come up between them. Then, for the BGP routes learnt over the TCP session, where the next-hop is the neighbor, they also provide the BGP NH resolution.

Unlike other BGP routes, these are not recursive routes as in they point to the neighbor's interface and IP address. These routes that are setup as part of the neighbor discovery procedure are hence different from the regular IBGP and EBGP routes. These routes also MUST have a better administrative distance as compared to the IBGP and EBGP routes to ensure that they do not get displaced from the forwarding by BGP routes learnt over the very session(s) established using these peering routes.

The Adjacency Routes SHOULD NOT be stored in any of BGP RIBs [RFC4271] since they are not computed based on the BGP decision process. It is RECOMMENDED that these routes be managed in a separate routing table within the BGP Neighbor Discovery function to ensure that none of the processing and validation for BGP RIB affects them and in turn they do not influence the BGP decision process and route calculation.

When there are multiple interconnecting links between two BGP neighbors, a single BGP TCP session may be setup between them over which routes are then exchanged. However, in the forwarding, the Adjacency route will have multiple paths - one for each of these interconnecting links. So the BGP routes learnt over the session actually end up getting resolved over this Adjacency route and in turn gets the ECMP load balancing even with a single BGP session.

10. Interactions with Base BGP Protocol

The BGP Finite State Machine (FSM) as specified in [RFC4271] is unchanged and the BGP TCP session establishment, route updates and processing continues to follow the BGP protocol specifications.

BGP peering addresses along with their respective ASNs have traditionally been explicitly provisioned on both BGP neighbors. The difference that neighbor discovery mechanism brings about is in elimination of this configuration as these parameters are learnt via the neighbor discovery procedure. Once BGP router learns its neighbor’s peering address and ASN, then its initializes the BGP Peer FSM for this neighbor in the Idle State - just as if this neighbor was configured. From thereon, the BGP Peer FSM actions follows.

The BGP Keepalives and Hold Timer for the session over TCP apply unchanged and they govern the operations of the BGP TCP session. While the BGP Keepalive works at the TCP session level, the BGP Adjacency Hold Timer monitors one or more underlying interconnecting link adjacencies between the neighbors. The reachability for the BGP TCP session may also be over the some BGP routes learnt via routing updates over the sessions setup via neighbor discovery. It is likely that even after all the underlying interconnecting link adjacencies between two neighbors are down that the neighbor’s peering address is reachable via BGP routing over some other path in the network. In order to avoid this, it is RECOMMENDED that the BGP TCP sessions setup via neighbor discovery mechanism use TTL set to 1 to ensure they are setup only over directly attached links to the neighbors.

Since the BGP TCP session setup via neighbor discovery was meant for hop-by-hop routing, it would be necessary to bring down the session even while its BGP Hold Timer has not expired for faster convergence.

Therefore, when all the underlying link adjacencies between two BGP neighbors move out of the Accepted state (or go down), then the BGP TCP peering session that was setup using BGP Neighbor Discovery mechanism between these two neighbors is also deleted as if it was un-configured.

Since the BGP neighbor discovery mechanism runs over a UDP socket, it is isolated from the core BGP protocol working which is TCP based. Implementations SHOULD ensure that the hello processing does not affect the base BGP operations and scalability. One option may be to run the BGP neighbor discovery mechanism in a separate thread from the rest of BGP processing. These implementation details, however, are outside the scope of this document.

It is not generally expected that BGP sessions are explicitly provisioned along with the neighbor discovery mechanism. However, in such an event, the neighbor discovery mechanism MUST NOT affect or result in any changes to provisioned BGP neighbors and their operations. Specifically, BGP peering to auto-discovered neighbors MUST NOT be instantiated using the procedures described in this document when the same BGP neighbor is already provisioned. The configured BGP neighbor parameters take precedence and the auto-discovered values and parameters are not used for such configured BGP sessions.

11. Security Considerations

BGP routers accept TCP connection attempts to port 179 only from the provisioned BGP neighbors or, in some implementations, those from within a configured address range. With the BGP neighbor auto-discovery mechanism, it is now possible for BGP to automatically learn neighbors and initiate/receive TCP connections from them. This introduces the need for specific considerations to be taken care of to ensure security of the BGP protocol operations.

This document introduces UDP messages in BGP for the neighbor discovery mechanism using the BGP Hello messages. For security purposes, implementations MUST exchange the Hello messages only on interfaces specifically enabled for neighbor discovery. Hello messages MUST NOT be accepted on other than the 224.0.0.2 or FF02::2 addresses. Optionally, implementations MAY set TTL to 255 when originating the Hello messages and receivers check specifically for the TLV to be 254 and discard the packet when this is not the case. This ensures that the Hello packets signaling happens between directly connected BGP routers only.

The BGP neighbor discovery mechanism is expected to be run typically in DCs and between physically connected routers that are trustworthy.

The Cryptographic Authentication TLV (as described in Section 8.6) SHOULD be used in deployments where this assumption of trustworthiness is not valid. This mechanism is similar to one defined for LDP Hello messages that are also UDP based as specified in [RFC7349]. An updated future version of this document will describe similar procedures for BGP hello in more details.

Once the BGP hello messages and the neighbor discovery mechanism is secured, then the security considerations for BGP protocol operations apply for the auto-discovered neighbor sessions.

12. Manageability Considerations

This section is structured as recommended in [RFC5706].

12.1. Operational Considerations

The BGP neighbor discovery mechanism introduced by this document is not applicable to general BGP deployments as discussed in Section 3. The mechanism is specifically meant for networks where BGP is used as a hop-by-hop routing protocol E.g. as described in [RFC7938]. The neighbor discovery mechanism hence SHOULD NOT be enabled by default in BGP.

Implementations SHOULD provide configuration methods that allow enablement of BGP neighbor discovery on specific local interfaces. In a DC network, it is expected that the operator selects the appropriate links on which to enable this e.g. on a Tier 2 node it is enabled on all links towards the Tier 1 and Tier 3 nodes while on a Tier 1 node, it may be only enabled on the links towards the Tier 2 node. The details of this enablement are outside the scope of this document since it varies based on the DC design and may be implementation specific.

Implementations SHOULD provide configuration methods that enable the setup of BGP neighbor templates that enables operator to setup BGP neighbor discovery parameters on the BGP router. Some of the aspects to be considered in such a template are:

o Local address to be used for the BGP TCP session peering along with the local ASN and the AFI/SAFI enabled for the auto-discovered sessions

o BGP policies to be enabled for the auto-discovered sessions

o Optionally specify the list of ASNs with which auto-discovered sessions should be brought up. This is to ensure that when links between different Tier nodes are not used by BGP when they get connected wrongly due to accidents (e.g. say a Tier 3 node is connected to a Tier 1 node).

- Authentication methods that are need to be enabled in an environment which is not secure

- Local interfaces over which the specific template needs to be applied for BGP neighbor discovery

- Other parameters like the Adjacency Hold Timer value to be used or other optional features

This mechanism does not impose any restrictions on the way ASNs or addresses are assigned to the nodes. Various automatic provisioning, auto-configuration or zero-touch-provisioning mechanisms may be used.

Implementations SHOULD report the state of the BGP operations over each link enabled for neighbor discovery including the status of all adjacencies learnt over it. Implementations SHOULD also report the operations of the auto-discovered BGP TCP peering sessions similar to the provisioned BGP neighbors.

Implementations SHOULD support logging of events like discovery of an adjacency using neighbor discovery including peering route updates and events like triggering of BGP TCP session establishment for them. Errors and alarms related to loss of adjacencies and tear down of BGP TCP peering sessions SHOULD also be generated so they could be monitored.

12.2. Management Considerations

This document introduces UDP based messaging in BGP protocol and therefore the necessary fault management mechanisms are required to be implemented for the same. Implementations MUST discard unsupported message types or version types other than 4 received over a UDP session. Such messages MUST NOT affect the neighbor discovery mechanism in operation using the Hello messages. Unknown TLVs received via the Hello messages MUST be ignored and the rest of the Hello message MUST be processed. Implementations SHOULD discard Hello messages with malformed TLVs and this should be logged as an error.

13. IANA Considerations

This documents requests IANA for updates to the BGP Parameters registry as described in this section.

13.1. BGP Hello Message

This document requests IANA to allocate a new UDP port (179 is the preferred number) and a BGP message type code for BGP Hello message.

Value	TLV Name	Reference

Service Name: BGP-HELLO
Transport Protocol(s): UDP
Assignee: IESG <iesg@ietf.org>
Contact: IETF Chair <chair@ietf.org>.
Description: BGP Hello Message.
Reference: This document -- draft-xu-idr-neighbor-autodiscovery.
Port Number: 179 (preferred value) -- To be assigned by IANA.

13.2. TLVs of BGP Hello Message

This document requests IANA to create a new registry "TLVs of BGP Hello Message" with the following registration procedure:

Registry Name: TLVs of BGP Hello Message.

Value	TLV Name	Reference
0	Reserved	This document
1	Accepted ASN List	This document
2	Peering Address	This document
3	Local Prefix	This document
4	Link Attributes	This document
5	Neighbor	This document
6	Cryptographic Authentication	This document
7-65500	Unassigned	
65501-65534	Experimental	This document
65535	Reserved	This document

14. Acknowledgements

The authors would like to thank Enke Chen, Krishna Swamy and Ramesh Yakkala for their valuable comments and suggestions on this document.

15. Contributors

Satya Mohanty
Cisco
Email: satyamoh@cisco.com

Shunwan Zhuang
Huawei
Email: zhuangshunwan@huawei.com

Chao Huang
Alibaba Inc
Email: jingtan.hc@alibaba-inc.com

Guixin Bao
Alibaba Inc
Email: guixin.bgx@alibaba-inc.com

Jinghui Liu
Ruijie Networks
Email: liujh@ruijie.com.cn

Zhichun Jiang
Tencent
Email: zcjiang@tencent.com

Shaowen Ma
Mellanox
mashaowen@gmail.com

16. References

16.1. Normative References

[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>.

[RFC5036] Andersson, L., Ed., Minei, I., Ed., and B. Thomas, Ed., "LDP Specification", RFC 5036, DOI 10.17487/RFC5036, October 2007, <https://www.rfc-editor.org/info/rfc5036>.

[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>.

16.2. Informative References

[FIPS-180-4] Technology, N. I. O. S. A., "Secure Hash Standard (SHS), FIPS PUB 180-4", March 2012.

[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-06 (work in progress), September 2019.

[I-D.ketant-idr-bgp-ls-bgp-only-fabric] Talaulikar, K., Filsfils, C., ananthamurthy, k., Zandi, S., Dawra, G., and M. Durrani, "BGP Link-State Extensions for BGP-only Fabric", draft-ketant-idr-bgp-ls-bgp-only-fabric-03 (work in progress), September 2019.

[RFC2104] Krawczyk, H., Bellare, M., and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication", RFC 2104, DOI 10.17487/RFC2104, February 1997, <https://www.rfc-editor.org/info/rfc2104>.

[RFC5706] Harrington, D., "Guidelines for Considering Operations and Management of New Protocols and Protocol Extensions", RFC 5706, DOI 10.17487/RFC5706, November 2009, <https://www.rfc-editor.org/info/rfc5706>.

[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>.

[RFC7349] Zheng, L., Chen, M., and M. Bhatia, "LDP Hello Cryptographic Authentication", RFC 7349, DOI 10.17487/RFC7349, August 2014, <https://www.rfc-editor.org/info/rfc7349>.

[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>.

Authors' Addresses

Xiaohu Xu
Alibaba Inc
China

Email: xiaohu.xxh@alibaba-inc.com

Ketan Talaulikar
Cisco Systems
India

Email: ketant@cisco.com

Kunyang Bi
Huawei
China

Email: bikunyang@huawei.com

Jeff Tantsura
Apstra
USA

Email: jefftant.ietf@gmail.com

Nikos Triantafillis
Amazon Web Services
USA

Email: ntriantafillis@gmail.com