IDR Working Group
Internet-Draft
Intended status: Standards Track
Expires: 13 November 2026

K. Vairavakkalai
M. Jeyananth
HPE
M. Nanduri
Microsoft
A. Reddy
AT&T

12 May 2026

BGP MultiNexthop Attribute
draft-ietf-idr-multinexthop-attribute-05

Abstract

Today, a BGP speaker can advertise one nexthop for a set of NLRIs in an Update message. This nexthop can be encoded in either the top-level BGP-Nexthop attribute (code 3), or inside the MP_REACH_NLRI attribute (code 14). Forwarding information related to the nexthop is scattered across various attributes, extended communities or the NLRI field.

This document defines a new optional non-transitive BGP attribute called "MultiNexthop (MNH)" with IANA code TBD. The MNH provides two things: it allows carrying the Nexthop and related forwarding information in one BGP attribute. The MNH also enables carrying an ordered set of multiple Nexthops in the same attribute, with forwarding information scoped on a per Nexthop basis.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

Today, a BGP speaker can advertise one nexthop for a set of NLRIs in an Update message. This nexthop can be encoded in either the top-level BGP-Nexthop attribute (code 3), or inside the MP_REACH_NLRI attribute (code 14). Forwarding information related to the nexthop is scattered across various attributes, extended communities or the NLRI field.

This document defines a new optional non-transitive BGP attribute called "MultiNexthop (MNH)" with IANA code TBD. The MNH provides two things: it allows carrying the Nexthop and related forwarding information in one BGP attribute. The MNH also enables carrying an ordered set of multiple Nexthops in the same attribute, with forwarding information scoped on a per Nexthop basis.

2. Terminology

iSN: Ingress Service Node

eSN: Egress Service Node

NLRI: Network Layer Reachability Information

AFI: Address Family Identifier

SAFI: Subsequent Address Family Identifier

PE : Provider Edge

RT : Route-Target extended community

RD : Route-Distinguisher

MPLS: Multi Protocol Label Switching

ECMP: Equal Cost Multi Path

WECMP: Weighted Equal Cost Multi Path

FRR: Fast Re Route

PNH : Protocol Next hop address carried in a BGP Update message

MNH: BGP MultiNextHop attribute

NFI: Nexthop Forwarding Information

FI: Forwarding Instruction

FA: Forwarding Argument

2.1. Definitions

MULTI_NEXT_HOP (aka MNH): BGP MultiNexthop attribute. The new attribute defined by this document.

MNH TLV: Top level TLV contained in a MULTI_NEXT_HOP.

NFI TLV: Nexthop Forwarding Information TLV, contained in a MNH TLV.

FI TLV: Forwarding Instruction TLV, contained in a NFI TLV.

FA TLV: Forwarding Argument TLV, contained as an argument to a FI in the FI TLV.

3. Motivation

Today, in a BGP Update, forwarding information related to the BGP nexthop is scattered across various attributes, extended communities or the NLRI field. On some other address families like Flowspec, nexthop address is carried without using the nexthop attribute, in one or more extended communities of specific type.

It may be desirable to carry the forwarding information for a nexthop scoped in a single attribute, and uniformly for all address families.

For cases where multiple nexthops need to be advertised, BGP Addpath [RFC7911] is used with some address families. Though Addpath allows basic ability to advertise multiple routes, it does not allow the sender to express the desired relationship between the multiple nexthops being advertised e.g., relative ordering, type of load balancing, fast reroute. These are local decisions based on configuration and path selection at the receiving node. Also, Addpath does not consider things like Link-bandwidth community when selecting add-path routes. Some scenarios (explained in Appendix A) may benefit from having a mechanism, where egress node can signal multiple nexthops along with their relationship to ingress nodes.

It would be desirable to have a common way to carry more than one nexthop on a BGP route of any family, and express relationship between them.

This document defines a new optional non-transitive BGP attribute "MultiNexthop (MNH)" that can be used for these purposes. The MNH attribute can be used in any BGP family that wants to carry one or more nexthops, with all forwarding information being carried in one attribute, scoped on a per nexthop basis.

E.g. The MNH can be used to advertise MPLS label along with nexthop for labeled and unlabeled families (e.g. Inet Unicast, Inet6 Unicast, Flowspec) alike. Such that, mechanisms at the transport layer can work uniformly on labeled and unlabled BGP families to realize various use-cases.

4. Base Encoding And Protocol Procedures

"MultiNexthop (MNH)" is a new BGP optional non-transitive attribute (code TBD), that can be used to carry an ordered set of one or more Nexthops in the same route, with all forwarding information being carried in one attribute, scoped on a per nexthop basis. This attribute describes forwarding instructions using TLVs as shown below.

This section describes the organization and encoding of the MNH attribute.

```
MNH Attribute: {
      PrimaryPath {
          [Forwarding Instruction 1],
           ..
          [Forwarding Instruction n]
      }
      RepairPath {
          [Forwarding Instruction 1],
           ..
          [Forwarding Instruction n]
      }
}

Forwarding Instruction: {
    {FwdAction, Forwarding Arguments}
}
```

Figure 1: Overview of MNH Attribute Layout - Eye candy summary

A MNH attribute consists of a Header and one or more "MNH TLVs".

A MNH TLV contains a Type and one unit of "Nexthop Forwarding Information" (NFI TLV).

A NFI TLV contains one or more "Forwarding Instructions" (FI TLVs).

A FI TLV contains a "Forwarding Action" code and one more "Forwarding Arguments" (FA TLVs).

The FA TLVs describe the parameters required to complete a "Forwarding Action".

4.1. MultiNexthop Attribute

```
 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. Flags  |Attr. Type Code|          Length               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Ver| MNH-Flags |       Advertising Router ID  (Four ..         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   .. bytes)   |       MNH TLV                                 ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                       MNH TLV                                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

MNH Header:

- Attr. Flags (1 octet)
  BGP Path-attribute flags. indicating an Optional Non-Transitive attribute. i.e. Optional bit set, Transitive bit reset.

- Attr. Type Code (1 octet)
  Type code allotted by IANA. TBD.

- Length (1 or 2 octets)
  One or Two bytes field stating length of attribute value in bytes.

- Version (2 bits)
  MNH Version - indicates layout of the MNH header.
  Set to 0x0 indicating "MNH v0", which is defined in this document.

  If there is any significant changes to the skeletal layout of MNH attribute in future, this Version field will be useful.

- MNH Flags (6 bits)

```
 2 3 4 5 6 7
+-+-+-+-+-+-+
|R R R R R M|
+-+-+-+-+-+-+
```

  6 bits following the Version bits are MNH Flags.

  M: "Mandatory".
  Value 1 indicates that this MNH attribute is mandatory.
  If this MNH attr is invalid, the route is Unusable Hidden.

  R: Reserved. MUST be set to zero, SHOULD be ignored by receiver.

- Advertising Router ID (4 octets)
  Router ID of the BGP speaker that is the source of truth for the

contents of this MNH attribute.

MNH TLVs: One or more MNH TLVs are carried in a MNH attr. MNH TLV is described in subsequent sections.

Figure 2: MultiNexthop - BGP Attribute

4.1.1. Processing the MNH Header

A BGP speaker MUST fill MNH Version field to 0.

If a MNH is received with a Version other than 0, the MNH attribute MUST be considered invalid, and be treated as Unrecognized Non-transitive attribute.

The "Advertising Router ID" field identifies the router that added the MNH attribute.

The "Advertising Router ID" field is intended to help in debugging and troubleshooting only. It is not used for validating the MNH. The non-transitive nature of MNH avoids attribute escape.

4.1.2. Scope of Use, Origination and Propagation

The MNH attribute is intended to be used in a BGP free core, between egress and ingress BGP speakers that understand this attribute. These BGP speakers may have an intra-AS or inter-AS BGP session between them.

To avoid un-intentionally leaking the MNH to another AS, via a BGP speaker that does not understand MNH attribute, it is defined as "optional non-transitive". But this also means that a RR needs to be upgraded to support this attribute before any PEs in the network can make use of it.

Use of MNH on a BGP session is disabled by default. An implementation MUST provide configuration control on a per BGP neighbor address family basis, to enable MNH support.

A BGP speaker MUST NOT advertise MNH on a BGP route if MNH support is not enabled for the corresponding address family on the advertising BGP session.

If the MNH attribute is received on a BGP session where MNH support is not enabled, the attribute MUST be treated as Unrecognized non-transitive. This rule provides additional protection against unintended propagation of this attribute, when both BGP speakers understand MNH but receiver has not enabled the support. A RFC3392 Capability is not used for this purpose, because it would cause session reset whenever 'MNH support' config is changed.

Remaining text in this section apply when both receiving and advertising BGP sessions are enabled with MNH support.

When a BGP speaker receives the MNH attribute on a BGP route, and re-advertises it with the nexthop unchanged, it MUST propagate the attribute unchanged. E.g. a Route Reflector. The "Advertising Router ID" field in the MNH will thus contain the original BGP speaker that added it.

When a BGP speaker receives the MNH attribute on a BGP route, and re-advertises it with the nexthop altered, it processes the attribute but MUST NOT propagate it as-is. The BGP speaker MAY however attach a new instance of MNH attribute on the re-advertised route, and MAY derive its value from the received MNH. The "Advertising Router ID" field is set to the Router-ID of the speaker attaching the MNH.

A BGP speaker re-advertising a BGP route with nexthop unchanged MAY add the MNH attribute on the reflected BGP route, on behalf of the originating BGP speaker. The "Advertising Router ID" field in the MNH is set to the Router-ID of the speaker adding the MNH.

4.1.3. Interaction with Forwarding Info in Rest of Update

A type of forwarding information may be carried in both the MNH as-well as remaining portions of the Update message. E.g., AIGP, Link Bandwidth Extended Community.

The instance of forwarding information carried outside the MNH is associated with the BGP Nexthop (attribute codes 3, 14). The instance of forwarding information carried inside the MNH attribute is more specific, and overrides the one carried outside the MNH. It is associated with the Endpoint (Section 5.3.1) in Forwarding Instruction TLV (Section 4.4) that is part of same MNH attribute.

This rule holds good for any type of forwarding information carried in the MNH, unless specified otherwise by that type of forwarding information.

An exception to this rule is the MPLS Label stack information carried in [RFC8277] NLRI field. The Labels carried in the NLRI are imposed as inner labels with the Encapsulation Info specified in Encap Info TLV (Section 5.3.3) used as outer encapsulation.

4.1.4. MNH attribute Error Handling

As specified in [RFC7606] BGP update message can contain no more than one instance of MP_REACH attribute or NEXT_HOP attribute. Similarly, a BGP update MUST contain no more than one instance of MNH attribute. If the MNH attribute (whether recognized or unrecognized) appears more than once in an UPDATE message, then all the occurrences of the attribute other than the first one SHALL be discarded and the UPDATE message will continue to be processed. The anomaly MAY be logged for diagnosis.

A TLV or sub-TLV of a certain Type in a MNH attribute can occur only once, unless specified otherwise by that type value. If multiple instances of such TLV or sub-TLV is received, the instances other than the first occurrence are ignored.

MNH employs a hierarchical error detection mechanism, where an error in lower layer TLVs is percolated upwards to the MNH attribute, based on the M bit. If the M bit in a TLV is 0, any error in the TLV is ignored and continue processing. If the M bit is 1, the TLV is considered invalid, and the error exception percolates up to the upper layer TLV, which takes the decision again based on it’s M bit, until the MNH attribute top level. If the M bits all the way to the top were set to 1, then the error makes the MNH invalid.

If processing of a received MNH attribute resulted in an error, then the "M bit" is used to decide the action. If the M bit is 0, then the MNH attribute is ignored, [RFC7606] ’Attribute Discard’ approach MUST be used, and continue to process rest of the update. If M bit is 1, then the BGP Route containing the MNH MUST be considered Unusable.

Implementations MAY provide policy configuration to set M bit to 0 on a MNH attribute being added, this helps with testing impact of the MNH on receiving nodes. Once confident, the MNH attribute can be re-advertised with M bit set. This helps in graceful incremental deployment.

The definition of a certain type of TLV or Sub-TLV in the MNH should specify in it’s procedures, the value of M bit to be used. An implementation MAY provide configuration to set or reset the M bit.

4.2. MNH TLV

The type of MNH TLV describes how the forwarding information carried in the MNH TLV is used.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  MNH-TLV Flags| MNH Type Code |          Length               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               Value                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

- MNH-TLV Flags (1 octet)

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

All bits are reserved.

M: "Mandatory".
Value 1 indicates that this MNH TLV is mandatory.
If this MNH TLV is not understood, the MNH attribute containing it is considered invalid.

R: "Reserved".
MUST be set to zero, SHOULD be ignored by receiver.

- MNH Type Code (1 octet)
  Type of MNH TLV. 0 is Reserved.

- Length
  Length of Value portion in octets.

- Value
  Value portion contains the NFI TLV.

Figure 3: MNH TLV

A sending BGP speaker advertises the information for one ore more nexthops in a MNH TLV.

Information received in MNH TLV is used to create the Forwarding state at receiving BGP speaker.

The MNH Type code indicates how the information carried in the TLV is used at the receiving node.

4.2.1. MNH TLV Error Handling

If invalid Type Code 0 is received, the MNH TLV is ignored irrespective of "M bit", and continue to process rest of the update.

If the received Type Code is incompatible for the prefix in BGP NLRI, the MNH TLV is considered invalid.

If an unrecognized Type Code is received, or processing of a recognized MNH TLV type results in an error, the TLV is considered invalid.

Invalid MNH TLV is handled based on the "M bit" on the MNH TLV.

If the M bit is 0, then the MNH TLV is ignored and continue to process rest of the update. If M bit is 1, then the MNH attribute containing this MNH TLV is considered invalid, triggering the procedures in Section 4.1.4.

4.3. Nexthop Forwarding Information TLV

A Nexthop Forwarding Information TLV describes a MNH TLV. It contains one or more Forwarding Instruction TLVs. These Forwarding Instructions are the Forwarding Legs of the MNH.

```
       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
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |  NFI  Flags   |      Num-Nexthops             |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |        Forwarding Instruction TLV (F.I. TLV)                  ~
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      ~        Forwarding Instruction TLV (F.I. YLV)                  |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

 - NFI Flags (1 octet)

          0 1 2 3 4 5 6 7
         +-+-+-+-+-+-+-+-+
         |R R R R R R R M|
         +-+-+-+-+-+-+-+-+

          M: "Mandatory".
             Value 1 indicates that this NFI TLV (Nexthop Leg) is mandatory.
             If this Nexthop Leg is not understood, the MNH TLV
             containing it is considered invalid.

          R: "Reserved".
             MUST be set to zero, SHOULD be ignored by receiver.

 - Num-Nexthops
        Number of F.I. TLVs.

 - Forwarding Instruction TLV
        Each F.I. TLV describes a Nexthop Leg.
        Layout of Forwarding Instruction TLV is described in next section.
```

Figure 4: Nexthop Forwarding Information TLV

M bit on a NFI TLV SHOULD be set to 1.

4.3.1. NFI TLV Error Handling

If Num-Nexthops in a received NFI is 0, it is considered invalid. Irrespective of M bit value, the NFI TLV is ignored and remaining update is processed.

The receiving BGP speaker MAY consider the "Num-Nexthops" value in a Nexthop Forwarding Information TLV not acceptable, based on it’s forwarding capabilities or local policy. In such cases, the NFI TLV is considered Invalid.

An Invalid NFI TLV is handled based on value of M bit on it. If the M bit is 0, the NFI TLV is ignored, and remaining update continue to be processed. If M bit is 1, the MNH TLV carrying this NFI is considered Invalid, triggering the procedures in Section 4.2.1.

4.4. Forwarding Instruction TLV

Each Forwarding Instruction TLV describes a Nexthop Leg. It expresses a "Forwarding Action" (FwdAction) along with arguments required to complete the action. The type of actions defined by this TLV are given below. The arguments are denoted by "Forwarding Argument TLVs". The Forwarding Argument TLVs takes appropriate values based on the FwdAction.

Each FwdAction should note the Arguments needed to complete the action. Any extraneous arguments should be ignored. If the minimum set of arguments required to complete an action is not received, the Forwarding Instruction TLV should be ignored. Appropriate logging and diagnostic info MAY be provided by an implementation to help troubleshoot such scenarios.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  F.I. Flags   |          Relative Pref        |  FwdAction    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Fwd Argument TLV                           ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                    Fwd Argument TLV                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

- F.I. Flags (1 octet)

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

M: "Mandatory".
Value 1 indicates that this Forwarding Instruction is mandatory. If this instruction is not understood, the NFI TLV containing it is considered invalid.

R: "Reserved".
MUST be set to zero, SHOULD be ignored by receiver.

- Relative Pref (2 octets)

Unsigned 2 octet integer specifying relative order or preference, among the many forwarding instructions, to use in FIB. All usable nexthop legs with lowest relative-pref are installed in FIB as primary-path. Thus if multiple legs exist with that lowest relative-pref, ECMP is formed.

- FwdAction (1 octet)

Type Code denoting the Forwarding action to be performed by receiving node. 0 is Reserved.

- Length (2 octets)

Length in octets, of all Forwarding Argument TLVs.

Figure 5: Forwarding Instruction TLV

Definition of a Forwarding Action should specify the set of forwarding arguments required to execute the action, and value of M bit.

4.4.1. FI TLV Error Handling

If an Invalid value of 0 is received as FwdAction, the TLV is ignored irrespective of "M bit", and continue to process rest of the update..

If an unrecognized or unsupported FwdAction is received, the FI TLV is considered Invalid.

If a certain Forwarding Action is unable to be executed because the set of required arguments are not available, the FI TLV is considered Invalid. If a certain Forwarding Action is applied to an incompatible NLRI, the FI TLV is considered Invalid.

An Invalid FI TLV is handled based on value of M bit on it. If the M bit is 0, the FI TLV is ignored, and remaining update continue to be processed. If M bit is 1, the NFI TLV carrying this NFI is considered Invalid, triggering the procedures in Section 4.3.1.

4.5. Forwarding Argument TLV

The Forwarding Argument TLV describes various parameters required to execute a FwdAction.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  F.A. Flags   |     F.A. Type Code            |  Length       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Length     |     Value                                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

- F.A. Flags (1 octet)

```
 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+
|R R R R R E C M|
+-+-+-+-+-+-+-+-+
```

M: "Mandatory".
Value 1 indicates that this argument is mandatory for the Forwarding Action
.
If this argument is not understood, the FI TLV containing it is considered invalid.

C: "Cumulative".
Request nodes to accumulate value in re-advertised MNH.
By default Forwarding Arguments are not cumulative, so C bit is 0 unless otherwise specified by the forwarding argument type.

E: "Egress Attached".
This bit is maintained when C bit is set to 1.
E bit is set to 1 if a cumulative argument is being added to a route with empty AS-path.

R: "Reserved".
MUST be set to zero, SHOULD be ignored by receiver.

- F.A. Type Code (2 octets)

Type Code of Forwarding Argument. 0 is Reserved.

- Length (2 octets)

Length in bytes of Value field.

Figure 6: Forwarding Argument TLV

The C bit is set to 1 on attributes that need to be accumulated across BGP nexthop-self propagation hops. If a received MNH has a FA with C bit 1, it MUST be set to 1 on the FA inserted in any advertised MNH also. The value of the FA in the advertised MNH MAY be derived from the value of the FA in the received MNH. The specific FA SHOULD define the procedure on how the accumulation of value happens for the specific type of FA.

If a received MNH has a FA with C bit 1, and receiving speaker is unable to perform the accumulation of FA, it MUST NOT include the FA type in any advertised MNH.

A FA that need to be accumulated end-to-end may want to know if the cumulative value denotes the path until the Egress node. The E bit denotes that the FA was originated by the Egress node that originated this BGP route. The E bit is set to 1 by the node adding the FA, if the AS-path on the route is empty. The E bit value received on a MNH MUST be propagated on the MNH added to the re-advertisement. This allows the Ingress node to see the E bit value set by the Egress node.

4.5.1. FA TLV Error Handling

If an Invalid F.A. Type Code value of 0 is received, the TLV is ignored irrespective of "M bit", and continue to process rest of the update..

If an unrecognized F.A. Type Code is received, the FA TLV is considered Invalid.

An Invalid FA TLV is handled based on value of M bit on it. If the M bit is 0, the FA TLV is ignored, and remaining update continue to be processed. If M bit is 1, the FI TLV carrying this FA is considered Invalid, triggering the procedures in Section 4.4.1.

4.6. Interaction with Addpath

[I-D.draft-ietf-idr-add-paths-guidelines-08] Sec 2 suggests the following:

"Diverse path: A BGP path associated with a different BGP next-hop and BGP router than some other set of paths. The BGP router associated with a path is inferred from the ORIGINATOR_ID attribute or, if there is none, the BGP Identifier of the peer that advertised the path."

When selecting "diverse paths" for ADD_PATH as specified above, the MNH attribute should also be compared if it exists, to determine if two routes have "different BGP next-hop".

4.7. Path Selection Considerations

4.7.1. Determining IGP Cost

While tie breaking in the path-selection as described in [RFC4271], 9.1.2.2. step (e) viz. the "IGP cost to nexthop", consider the highest cost among the nexthop-legs present in this attribute.

The IGP cost thus calculated is also used when constructing AIGP TLV ([RFC7311])

5. TLVs Defined In This Document

This section describes the initial set of MNH TLVs, Forwarding Instructions and Arguments that this document defines.

5.1. MNH TLVs

The type of MNH TLV describes how the forwarding information carried in the MNH TLV is used.

This document defines the following MNH TLV types:

MNH Type Code	Meaning
0	Reserved
1	Primary forwarding path
2	Backup forwarding path

- Length
  Length of Value portion in octets.

- Value
  Value portion contains the NFI TLV.

Figure 7: MNH TLV Types

Type codes 1 and 2 are applicable for upstream allocated prefixes, example IP, Upstream MPLS labels, Flowspec routes.

Note that usage of Type code 1 in a BGP route containing IP prefix gives similar result as advertising the route with nexthop contained in BGP path-attributes: Nexthop (code 3) or MP_REACH_NLRI (code 14).

Upstream allocation for MPLS routes is achieved by using mechanisms explained in [I-D.draft-kaliraj-bess-bgp-sig-private-mpls-labels-06].

If an invalid Type Code 0 is received, the TLV is ignored, and continue to process rest of the update.

If the received Type Code is incompatible for the prefix in BGP NLRI, the TLV should be ignored.

5.1.1. Primary Forwarding Path

This is a MNH TLV (Section 4.2) with MNH Type Code = 1, called "Primary Forwarding Path"

This TLV describes forwarding state to be programmed at receiving speaker as Primary Path nexthop leg. This TLV is used with Upstream allocated or global scope prefixes carried in BGP NLRI. Value part of this TLV contains Nexthop Forwarding Information TLV.

A BGP speaker uses the nexthop forwarding information received in this TLV as a primary path nexthop leg when programming the route for the NLRI prefix in its Forwarding table.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  MNH-TLV Flags|  MNH Type = 1 |          Length               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|               Nexthop Forwarding Information TLV              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 8: Primary forwarding path TLV

5.1.2. Repair Forwarding Path

This is a MNH TLV (Section 4.2) with MNH Type Code = 2, called "Repair Forwarding Path"

This TLV describes forwarding state to be programmed during traffic repair at receiving speaker. i.e. This TLV is used to program a backup/repair path. This TLV is used with Upstream allocated prefixes or global scoped prefixes. Value part contains Nexthop Forwarding Information TLV.

Signaling a different nexthop for use as backup path is desirable in some labeled forwarding scenarios, where two multihomed edge devices use each other as backup path to protect traffic when primary path fails.

This is required to avoid label advertisement oscillation between the multihomed PEs when they implement per-nexthop label allocation mode.

The label advertised by a PE1 for primary path advertisement is allocated/forwarded using external paths as primary leg and backup-path label from other multihomed PE2 as backup-path label. Such that primary-path label allocation at PE1 is not a function of the primary-path label advertised by PE2. Thus the primary path label remains stable at a PE and does not change when a new primary path label is received from the other multihomed PE. This prevents the label oscillation problem.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  MNH-TLV Flags|  MNH Type = 2 |          Length               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|               Nexthop Forwarding Information TLV              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 9: Repair forwarding path TLV

The backup path label allocated and advertised by a PE is a function of only the primary path. E.g. path to the CE device. So this label value does not change when a new label is received from the other multihomed PE

5.2. Forwarding Actions in FI TLV

Each Forwarding Instruction TLV describes a Nexthop Leg. It expresses a "Forwarding Action" (FwdAction) along with arguments required to complete the action. The type of actions defined by this TLV are given below. The arguments are denoted by "Forwarding Argument TLVs". The Forwarding Argument TLVs takes appropriate values based on the FwdAction.

Each FwdAction should note the Arguments needed to complete the action. Any extraneous arguments should be ignored. If the minimum set of arguments required to complete an action is not received, the Forwarding Instruction TLV should be ignored. Appropriate logging and diagnostic info MAY be provided by an implementation to help troubleshoot such scenarios.

Following Forwarding Actions are defined by this document.

FwdAction	Meaning
0	Reserved
1	Forward
2	Pop-And-Forward
3	Swap
4	Push
5	Pop-And-Lookup
6	Replicate

Forwarding Instruction TLV with unknown FwdAction should be ignored, skipped and rest of the attribute processed; gracefully handling the error. The event may be appropriately logged for diagnosis.

- Length (2 octets)

  Length in octets, of all Forwarding Argument TLVs.

Figure 10: Forwarding Actions

Meaning of most of the above FwdAction semantics is well understood. FwdAction 1 is applicable for both IP and MPLS routes. FwdActions 2-5 are applicable for encapsulated payloads (like MPLS) only. FwdActions 1, 6 are applicable for Flowspec routes for Redirect and Mirror actions. FwdAction 6 can also be used to indicate multicast replication like functionality.

The "Forward" action means forward the IP/MPLS packet with the destination prefix (IP-dest-addr/MPLS-label) value unchanged. For IP routes, this is the forwarding-action given for next-hop addresses contained in BGP path-attributes: Nexthop (code 3) or MP_REACH_NLRI (code 14). For MPLS routes, usage of this action is equivalent to SWAP with same label-value; one such usage is explained in [I-D.draft-kaliraj-bess-bgp-sig-private-mpls-labels-06] when Upstream-label-allocation is in use.

The "Pop-And-Forward" action means Pop the payload header (e.g. MPLS-label) and forward the payload towards the Nexthop IP-address specified in the Endpoint Id TLV, using appropriate encapsulation to reach the Nexthop.

When applied to MPLS packet, the "Pop-And-Lookup" action may result in a MPLS-lookup or an upper-layer header (like IPv4, IPv6) lookup, depending on whether the label that was popped was the bottom of stack label.

If an incompatible FwdAction is received for a prefix-type, or an unsupported FwdAction is received, it is considered a semantic-error and MUST be dealt with as explained in "Error handling procedures" section.

5.3. Forwarding Argument TLVs

The Forwarding Argument TLV describes various parameters required to execute a FwdAction.

Following types of Forwarding Argument are defined by this document.

F.A. Type Code	Meaning
0	Reserved
1	Endpoint Identifier
2	Path Constraints
3	Payload encapsulation info signaling
4	Endpoint attributes advertisement

- Length (2 octets)

  Length in bytes of Value field.

Figure 11: Forwarding Argument Types

5.3.1. Endpoint Identifier

This is a Forwarding Argument (Section 5.3) with F.A. Type Code = 1. It identifies an Endpoint of certain type.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  F.A. Flags   |     F.A. Type Code =1         |  Length       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Length     | Endpoint Type |  Endpoint Len | Endpoint Value|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  Endpoint Value                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

- F.A. Flags (1 octet)
  As defined in Forwarding Argument TLV.

- Length (2 octets)
  Length in bytes of Value field.

Endpoint Type	Value	Len (octets)
0	Reserved	
1	IPv4 Address	4
2	IPv6 Address	16
3	MPLS Label (Upstream allocated or Global scope)	4
4	Fwd Context RD	8
5	Fwd Context RT	8

- Endpoint Len (1 octet)

  Length in bytes of Endpoint Value field.

Figure 12: Endpoint Identifier

5.3.2. Path Constraints

This is a Forwarding Argument (Section 5.3) with F.A. Type Code = 2. It defines Constraints for Path to the Endpoint..

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  F.A. Flags   |     F.A. Type Code = 2        |  Length       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Length     | ConstrainType | Constrain Len | ConstrainValue|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  ConstrainValue                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

- F.A. Flags (1 octet)
  As defined in Forwarding Argument TLV.

- Length (2 octets)
  Length in bytes of Value field.

ConstrainType	Value	Len (octets)
0	Reserved	
1	Proximity check	2
2	Transport Class ID (Color)	4
3	Load balance factor	2

- Constrain Len (1 octet)

  Length in bytes of Constrain Value field.

- Proximity check Flags (2 octets)
  Flags describing whether the nexthop endpoint is expected to be single hop away, or multihop away. Format of flags is described in next section.

- Transport Class ID (Color):

  This is a 32 bit identifier, associated with the Nexthop address. The Nexthop IP-address specified in "Endpoint Identifier" TLVs are resolved over tunnels of this color. Defined in {{RFC9832}}

- Load balance factor (2 octets)
  Balance Percentage

Figure 13: Path Constraints

5.3.2.1. Proximity Check

Usually EBGP singlehop received routes are expected to be one hop away, directly connected. And IBGP received routes are expected to be multihop away. Implementations today provide configuring exceptions to this rule.

The 'expected proximity' of the Nexthop can be signaled to the receiver using the Proximity check flags. Such that irrespective of whether the route is received from IBGP/EBGP peer, it can be treated as a single-hop away or multihop away nexthop.

The format of the Proximity check Sub-TLV 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  F.A. Flags   |     F.A. Type Code = 2        |  Length       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Length     |ConstrainType=1|  Len = 2      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       Proximity Check Flags   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

- F.A. Flags (1 octet)
  As defined in Forwarding Argument TLV.

- Length (2 octets)
  Length in bytes of Value field.

- Proximity check Flags (2 octets)

```
         0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
        +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
        |S M R R R R R R R R R R R R R R|
        +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

S: Restrict to Singlehop path.
M: Expect Multihop path.
R: Reserved. MUST be set to zero, SHOULD be ignored by receiver.

Figure 14: Proximity check constrain

This TLV would be valid with Forwarding Instructions TLV with FwdAction of Forward, Pop-And-Forward, Swap or Push.

When S bit is set, receiver considers the nexthop valid only if it is directly connected to the receiver.

When M bit is set, receiver assumes that the nexthop can be multiple hops away, and resolves the path to the nexthop via another route.

When both S and M bits are set, M bit behavior takes precedence. When both S and M bits are Clear, the current behavior of deriving proximity from peer type (EBGP is singlehop, IBGP is multihop) is followed.

5.3.2.2. Transport Class ID (Color)

The Nexthop can be associated with a Transport Class, so as to resolve a path that satisfies required Transport tunnel characteristics. Transport Class is defined in [RFC9832]

Transport Class is a per-nexthop scoped attribute. Without MNH, the Transport class is applied to the nexthop IP-address encoded in the BGP-Nexthop attribute (code 3), or inside the MP_REACH_NLRI attribute (code 14). With MNH, the Transport Class can be specified per Nexthop-Leg (Forwarding Instruction TLV Section 4.4). It is applied to the IP-address encoded in the Endpoint Identifier TLV of type "IPv4 Address", "IPv6 Address" , "MPLS Label (Upstream allocated or Global scope)".

The format of the Transport Class ID Sub-TLV 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  F.A. Flags   |     F.A. Type Code = 2        |  Length       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Length     |ConstrainType=2|  Len = 4      | Transport..   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  .. Class ID (4 bytes)        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

  - F.A. Flags (1 octet)
     As defined in Forwarding Argument TLV.

  - Length (2 octets)
       Length in bytes of Value field.

  - Transport Class ID (Color):
    This is a 32 bit identifier, associated with the Nexthop address.
    The Nexthop specified in Endpoint Identifier TLVs
    are resolved over tunnels of this color.
  Defined in {{RFC9832}}
```

Figure 15: Transport Class ID (Color)

This TLV would be valid with Forwarding Instructions TLV with FwdAction of Forward, Swap or Push.

5.3.2.3. Load Balance Factor

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  F.A. Flags   |     F.A. Type Code = 2        |  Length       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Length     |ConstrainType=3|  Len = 2      |   Balance..   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|.. Percentage  |
+-+-+-+-+-+-+-+-+
```

- F.A. Flags (1 octet)
  As defined in Forwarding Argument TLV.

- Length (2 octets)
  Length in bytes of Value field.

- Len (1 octet)
  Length of the Constrain Value field.

- Balance Percentage:
  This is the explicit "balance percentage" requested by the sender, for unequal load-balancing over these Nexthop-Descriptor-TLV legs. This balance percentage would override the implicit balance-percentage calculated using "Bandwidth" attribute sub-TLV.

Figure 16: Load Balance Factor

This sub-TLV would be valid with Forwarding Instructions TLV with FwdAction of Forward, Swap or Push.

This is the explicit "balance percentage" requested by the sender, for unequal load-balancing over these Nexthop-Descriptor-TLV legs. This balance percentage would override the implicit balance-percentage calculated using "Bandwidth" attribute sub-TLV

When the sum of "balance percentage" on the nexthop legs does not equal 100, it is scaled up or down to match 100. The individual balance percentages in each nexthop leg are also scaled up or down proportionally to determine the effective balance percentage per nexthop leg.

5.3.3. Payload Encapsulation Info

This is a Forwarding Argument (Section 5.3) with F.A. Type Code = 3. It defines Payload Encapsulation Information.

```
     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
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |  F.A. Flags   |     F.A. Type Code =3         |  Length       |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |    Length     | Encap Type  |         Encap Len               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                    Encap Value                                |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

- F.A. Flags (1 octet)
  As defined in Forwarding Argument TLV.

- Length (2 octets)
  Length in bytes of Value field.

Endcap Type	Value
0	Reserved
1	MPLS Label Info
2	SR MPLS label Index Info
3	SRv6 SID info
4	DSCP code point

- Encap Len (2 octets)

  Length in octets of Encap Value field.

Figure 17: Payload Encapsulation Info

5.3.3.1. MPLS Label Info

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  F.A. Flags   |     F.A. Type Code =3         |  Length       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Length     | Encap Type=1 |          Encap Len             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      L.V. Flags (2 bytes)    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| MPLS Label (20 bits) |Rsrv |S~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~ MPLS Label (20 bits) |Rsrv |S|
-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+


  - F.A. Flags (1 octet)
     As defined in Forwarding Argument TLV.

  - Length (2 octets)
       Length in bytes of Value field.

  - Encap Type
     = 1, to signify MPLS Label Info.

  - Encap Len (2 octets)
       Length in bytes of following Encap Value field.

  - L.V. Flags (2 octets):

       0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |E R R R R R R R R R R R R R R R|
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

       E: ELC bit. Indicates if this egress NH is Entropy Label Capable.
             1 means the Entropy Label capable.
             0 means not capable to handle Entropy Label.

       R: Reserved. MUST be set to zero, SHOULD be ignored by receiver.

  - MPLS Label, Rsrv, S bit.
      20 bit MPLS Label stack encoded as in RFC 8277.
      S bit set on last label in label stack.
```

Figure 18: MPLS Label Info

5.3.3.2. SR MPLS Label Index Info

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  F.A. Flags   |     F.A. Type Code =3         |  Length       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Length     | Encap Type=2 |           Encap Len            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   RESERVED    |       LI Flags                |    Label ..   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                ..Index                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

- F.A. Flags (1 octet)
  As defined in Forwarding Argument TLV.

- Length (2 octets)
  Length in bytes of Value field.

- Encap Type
  = 2, to signify SR MPLS SID Info.

- Encap Len (2 octets)
  Length in bytes of following Encap Value field.

Rest of the value portion is encoded as specified in RFC-8669 sec 3.1.

- RESERVED: 8-bit field. MUST be set to zero, SHOULD be ignored by receiver.

- LI Flags: 16 bits of flags. None defined. MUST be set to zero, SHOULD be ignored by receiver.

- Label Index:
  32-bit value representing the index value in the SRGB space.

Figure 19: SR MPLS Label Index Info

5.3.3.3. SRv6 SID Info

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  F.A. Flags   |     F.A. Type Code =3         |  Length       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Length     | Encap Type=3 |           Encap Len            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         .. SRv6 SID Info (variable)                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

- F.A. Flags (1 octet)
  As defined in Forwarding Argument TLV.

- Length (2 octets)
  Length in bytes of Value field.

- Encap Type
  = 3, to signify SRv6 SID Info.

- Encap Len (2 octets)
  Length in bytes of following Encap Value field.

- SRv6 SID Info:
  SRv6 SID Information, as specified in RFC-9252 sec 3.1.

Figure 20: SRv6 SID Info

5.3.3.4. DSCP

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  F.A. Flags   |     F.A. Type Code = 3        |  Length       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Length     | Encap Type=4 |           Encap Len            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|DSCP code point|
+-+-+-+-+-+-+-+-+
```

- F.A. Flags (1 octet)
  As defined in Forwarding Argument TLV.

- Length (2 octets)
  Length in bytes of Value field.

- Encap Type
  = 4, to signify DSCP code point.

- Encap Len (2 octets)
  = 1, Length in bytes of following Encap Value field.

- DSCP code point:
  DS Field, as specified in RFC-2474 sec 3.

Figure 21: DSCP

5.3.4. Endpoint Attributes

This is a Forwarding Argument (Section 5.3) with F.A. Type Code = 4. It defines Attributes of an Endpoint.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  F.A. Flags   |     F.A. Type Code = 4        |  Length       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Length     | Attrib Type  |    Attr Len    |  Attr  Value  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Attr Value                                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

EP Attrib Type	Attrib Value	Attrib Len (octets)
0	None	
1	Endpoint Bandwidth	8
2	Accumulated Metric	Variable

Figure 22: Endpoint attributes

5.3.4.1. Endpoint Bandwidth

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  F.A. Flags   |     F.A. Type Code = 4        |  Length       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Length     | Attrib Type 1|    Attr Len=8  |  Attr  Value  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Bandwidth (8 octets)                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Bandwidth (contd.)                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

- F.A. Flags (1 octet)
  As defined in Forwarding Argument TLV.

- Len (2 octets)
  Length in bytes of remaining portion of SubTLV.

- Bandwidth
  The bandwidth to the endpoint expressed as 8 octets, units being bits per second.

Figure 23: Endpoint Bandwidth

This sub-TLV would be valid with Forwarding Instruction TLV with FwdAction of Forward, Swap or Push.

5.3.4.2. Accumulated Metric to Endpoint

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  F.A. Flags   |     F.A. Type Code = 4        |  Length       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Length     | Attrib Type 2|    Metric Type |  Metric Len   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   Metric Value (variable)                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+


 - F.A. Flags (1 octet)
     As defined in Forwarding Argument TLV.

        C: Cumulative bit is set to 1 by originator of this argument.


- Len (2 octets)
    Length in bytes of remaining portion of SubTLV.

- Metric Type: Type from "IGP Metric-Type" IANA registry under IGP Parameters
   Following types are defined by this document to be accumulated:
      0 IGP Metric
      1 Min Unidirectional Link Delay as defined in [RFC8570, Section 4.2]

- Metric Len: Length in octets of Metric Value field.
      IGP Metric: 4
      Min Unidirectional Link Delay: 4

- Metric Value:
      IGP Metric: 4 octet Accumulated IGP cost
      Min Unidirectional Link Delay: 4 octet Accumulated min delay in microseconds.
```

Figure 24: Accumulated Metric to Endpoint

This sub-TLV would be valid with Forwarding Instruction TLV with FwdAction of Forward, Swap or Push.

6. Scaling Considerations

The MNH attribute allows receiving multiple nexthops on the same BGP session. This flexibility also opens up the possibility that a peer can send large number of multipath (ECMP/UCMP/FRR) nexthops that may overwhelm the local system’s forwarding plane. Prefix-limit based checks will not avoid this situation.

To keep the scaling limits under check, a BGP speaker MAY keep account of number of unique multipath nexthops that are received from a BGP peer, and impose a configurable max-limit on that. This is especially useful for EBGP peers.

A good scaling property of conveying multipath nexthops using the MNH attribute with N nexthop legs on one BGP session, as against BGP routes on N BGP sessions is that, it limits the amount of transitionary multipath combinatorial state in the latter model. Because the final multipath state is conveyed by one route update in deterministic manner, there is no transitionary multipath combinatorial explosion created during establishment of N sessions.

7. IANA Considerations

This document makes request to IANA to allocate the following codes in BGP attributes registry.

7.1. BGP Path Attributes

A new BGP attribute code TBD for "BGP MultiNexthop Attribute (MULTI_NEXT_HOP)", in "BGP Path Attributes" registry.

7.2. BGP MultiNextHop Attribute

This document requests IANA to create a new registry group for MultiNextHop attribute, and the following registries in it.

7.2.1. MultiNextHop (MNH) TLV Types

This is a Registry for Type codes in (Section 4.2) "MNH TLV"

Under "Border Gateway Protocol (BGP) Parameters",

Registry Group: BGP MultiNextHop Attribute

Registry Name: MultiNexthop (MNH) TLV Types

MNH Type Code	Meaning
0	Reserved
1	Primary forwarding path
2	Backup forwarding path
3-254	Unassigned
255	Reserved

Reference: This document.

Registration Procedure(s)
Future assignments are to be made using either the Standards Action process defined in [RFC2434], or the Early IANA Allocation process defined in [RFC4020].

Figure 25: Registry - MNH TLV Types

7.2.2. Forwarding Action Types

This is a Registry for Type codes in Section 4.4 "Forwarding Instruction TLV"

Under "Border Gateway Protocol (BGP) Parameters",

Registry Group: BGP MultiNextHop Attribute

Registry Name: Forwarding Action Types

FwdAction	Meaning
0	Reserved
1	Forward
2	Pop-And-Forward
3	Swap
4	Push
5	Pop-And-Lookup
6	Replicate
7-254	Unassigned
255	Reserved

Reference: This document.

Registration Procedure(s)
Future assignments are to be made using either the Standards Action process defined in [RFC2434], or the Early IANA Allocation process defined in [RFC4020].

Figure 26: Registry - Forwarding Action Types

7.2.3. Forwarding Argument Types

This is a Registry for Type codes in (Section 5.3) "Forwarding Arguments TLV"

Under "Border Gateway Protocol (BGP) Parameters",

Registry Group: BGP MultiNextHop Attribute

Registry Name: Forwarding Argument Types

F.A. Type Code	Meaning
0	Reserved
1	Endpoint Identifier
2	Path Constraints
3	Payload encapsulation info signaling
4	Endpoint attributes advertisement
5-65534	Unassigned
65535	Reserved

Reference: This document.

Registration Procedure(s)
Future assignments are to be made using either the Standards Action process defined in [RFC2434], or the Early IANA Allocation process defined in [RFC4020].

Figure 27: Registry - Forwarding Argument Types

7.2.4. Endpoint Types

This is a Registry for Type codes in Section 5.3.1 "Endpoint Identifier" Forwarding Argument.

Under "Border Gateway Protocol (BGP) Parameters",

Registry Group: BGP MultiNextHop Attribute

Registry Name: Endpoint Types

Endpoint Type	Value
0	Reserved
1	IPv4 Address
2	IPv6 Address
3	MPLS Label
4	Fwd Context RD
5	Fwd Context RT
6-254	Unassigned
255	Reserved

Reference: This document.

Registration Procedure(s)
Future assignments are to be made using either the Standards Action process defined in [RFC2434], or the Early IANA Allocation process defined in [RFC4020].

Figure 28: Registry - Endpoint Types

7.2.5. Path Constrain Types

This is a Registry for Type codes in Section 5.3.2 "Path Constrain" Forwarding Argument.

Under "Border Gateway Protocol (BGP) Parameters",

Registry Group: BGP MultiNextHop Attribute

Registry Name: Path Constrain Types

ConstrainType	Value
0	Reserved
1	Proximity check
2	Transport Class ID (Color)
3	Load balance factor
4-254	Unassigned
255	Reserved

Reference: This document.

Registration Procedure(s)
Future assignments are to be made using either the Standards Action process defined in [RFC2434], or the Early IANA Allocation process defined in [RFC4020].

Figure 29: Registry - Path Constrain Types

7.2.6. Encapsulation Types

This is a Registry for Type codes in Section 5.3.3 "Payload Encapsulation Info" Forwarding Argument.

Under "Border Gateway Protocol (BGP) Parameters",

Registry Group: BGP MultiNextHop Attribute

Registry Name: Encapsulation Types

Encap Type	Value
0	Reserved
1	MPLS Label Info
2	SR MPLS label Index Info
3	SRv6 SID info
4	DSCP code point
5-254	Unassigned
255	Reserved

Reference: This document.

Registration Procedure(s)
Future assignments are to be made using either the Standards Action process defined in [RFC2434], or the Early IANA Allocation process defined in [RFC4020].

Figure 30: Registry - Encapsulation Types

7.2.7. Endpoint Attribute Types

This is a Registry for Type codes in Section 5.3.4 "Endpoint attributes" Forwarding Argument.

Under "Border Gateway Protocol (BGP) Parameters",

Registry Group: BGP MultiNextHop Attribute

Registry Name: Endpoint Attribute Types

EP Attrib Type	Attrib Value
0	Reserved
1	Bandwidth
2	Accumulated Metric to Endpoint
3-254	Unassigned
255	Reserved

Reference: This document.

Registration Procedure(s)
Future assignments are to be made using either the Standards Action process defined in [RFC2434], or the Early IANA Allocation process defined in [RFC4020].

Figure 31: Registry - Endpoint Attribute Types

Note to RFC Editor: this section may be removed on publication as an RFC.

8. Security Considerations

The MNH attribute is defined as optional non-transitive BGP attribute, such that it does not accidentally get propagated or leaked via BGP speakers that don’t support this feature, especially does not unintentionally leak across EBGP boundaries.

MNH may be used to advertise nexthop with MPLS label in various BGP families. In scenarios where MPLS is enabled on link to a device in an untrusted domain, e.g. a PE-CE link or ASBR-ASBR inter-AS link, security can be provided against MPLS label spoofing by using MPLS context tables as described in Appendix A.9.2. Such that only MPLS traffic with labels advertised to the BGP speaker are allowed to forward. However, the PE may not be able to perform any checks based on inner payload in the MPLS packet since it performs label swap forwarding. Such ’inner payload’ based checks may be offloaded to a downstream node that forwards and processes inner payload, e.g., an IP router having full FIB. These security aspects should be considered when using MPLS enabled CE devices.

9. Contributors

10. Acknowledgements

Thanks to Jeff Haas, Robert Raszuk, Ron Bonica for the review, discussions and input to the draft.

Thanks to Blaine Williams and Satya Mohanty for the discussions on some use-cases.

11. References

11.1. Normative References

[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/rfc/rfc7606>.

[RFC8277] Rosen, E., "Using BGP to Bind MPLS Labels to Address Prefixes", RFC 8277, DOI 10.17487/RFC8277, October 2017, <https://www.rfc-editor.org/rfc/rfc8277>.

[RFC7311] Mohapatra, P., Fernando, R., Rosen, E., and J. Uttaro, "The Accumulated IGP Metric Attribute for BGP", RFC 7311, DOI 10.17487/RFC7311, August 2014, <https://www.rfc-editor.org/rfc/rfc7311>.

[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/rfc/rfc4271>.

[RFC7911] Walton, D., Retana, A., Chen, E., and J. Scudder, "Advertisement of Multiple Paths in BGP", RFC 7911, DOI 10.17487/RFC7911, July 2016, <https://www.rfc-editor.org/rfc/rfc7911>.

11.2. Informative References

[RFC2474] Nichols, K., Blake, S., Baker, F., and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", RFC 2474, DOI 10.17487/RFC2474, December 1998, <https://www.rfc-editor.org/rfc/rfc2474>.

[RFC9832] Vairavakkalai, K., Ed. and N. Venkataraman, Ed., "BGP Classful Transport Planes", RFC 9832, DOI 10.17487/RFC9832, September 2025, <https://www.rfc-editor.org/rfc/rfc9832>.

[RFC9256] Filsfils, C., Talaulikar, K., Ed., Voyer, D., Bogdanov, A., and P. Mattes, "Segment Routing Policy Architecture", RFC 9256, DOI 10.17487/RFC9256, July 2022, <https://www.rfc-editor.org/rfc/rfc9256>.

[I-D.draft-kaliraj-bess-bgp-sig-private-mpls-labels-06] Vairavakkalai, K., Jeganathan, J. M., and P. Ramadenu, "BGP Signaled MPLS Namespaces", Work in Progress, Internet-Draft, draft-kaliraj-bess-bgp-sig-private-mpls-labels-06, 10 July 2023, <https://datatracker.ietf.org/doc/html/draft-kaliraj-bess-bgp-sig-private-mpls-labels-06>.

[I-D.draft-ietf-idr-flowspec-redirect-ip] Haas, J., Henderickx, W., and A. Simpson, "BGP Flow-Spec Redirect-to-IP Action", Work in Progress, Internet-Draft, draft-ietf-idr-flowspec-redirect-ip-10, 28 April 2026, <https://datatracker.ietf.org/doc/html/draft-ietf-idr-flowspec-redirect-ip-10>.

[I-D.draft-ietf-idr-add-paths-guidelines-08] Uttaro, J., Francois, P., Patel, K., Haas, J., Simpson, A., and R. Fragassi, "Best Practices for Advertisement of Multiple Paths in IBGP", Work in Progress, Internet-Draft, draft-ietf-idr-add-paths-guidelines-08, 25 April 2016, <https://datatracker.ietf.org/doc/html/draft-ietf-idr-add-paths-guidelines-08>.

Appendix A. Use Cases

This section describes various example use-cases of the MNH attribute.

A.1. Signaling WECMP to Ingress Node

This section describes how MNH can be used to provide weighted equal cost multipath in a network fabric, while not increasing RIB scale.

```
                                      [RR1]
                                        .
                      . +-[P21]         |
                     .  +-[P22]         __
                    .   +-[P23]      _.(  )..
               [R1].    +-[P24] ..  (_      _) .. [R2]
                    .   +-[P25]       (._..)
                     .
                      . +-[P2n]


                       <---- Traffic Direction ----
```

Figure 32: WECMP without increasing RIB scale

{topo-wecmp} shows a network with BGP speaker R1 connected to a number of routers P21 .. P2n in its region. R1 is eSN and R2 is iSN for the IP traffic in consideration. BGP service families IPv4 Unicast (AFI/SAFI: 1/1) and IPv6 Unicast (AFI/SAFI: 2/1) are negotiated on the BGP sessions between RR1 - R1 and RR1 - R2. RR1 reflects the BGP routes between R1 and R2 with next hop unchanged.

When MNH is not in use, R1 advertises "n" BGP Addpath routes for a service prefix Pfx1, each having a distinct next hop, P21 .. P2n, and desired Link Bandwidth Extended Community. These Addpath routes will be received by R2, which can do WECMP based on the Link Bandwidth Extended Communities attached on the routes. This model increases RIB scale by "n" times, so that WECMP can be achieved.

When MNH is used in this network, R1 advertises a single BGP route for prefix Pfx1, which contains a MNH attribute with "n" next hops, each carrying the desired link bandwidth using Section 5.3.2.3 or Section 5.3.4.1

This allows achieving WECMP in the network without increasing RIB scale.

A.2. Signaling Optimal Forwarding Exit-points to Ingress Node

In a BGP free core, one can dynamically signal to the ingress-node, how traffic should be load-balanced towards a set of exit nodes, in one BGP-route containing this attribute.

Example, for prefix1, perform equal load balancing towards exit nodes A, B; where as for prefix2, perform weighted load balancing (40%, 30%, 30%) towards exit nodes A, B, C.

Example, for prefix1, use PE1 as primary-nexthop and use PE2 as a backup-nexthop.

A.3. Load balancing to multiple CEs in a VRF

This section describes how MNH can be used to provide load balancing and entropy in a provider network for traffic destined to multiple CEs in a VRF, without increasing RIB scale.

```
                             [RR1]
                               .
              [CE1].           |
                   .          __
              [CE2]  .     _.(  )..
                      .[PE1]  ..  (_    _) .. [PE2]
              [CE3]  .         (._..)
                   .
              [CE4].


                    <---- Traffic Direction ----
```

Figure 33: Load balancing to multiple CEs in a VRF

{topo-cevrf} shows a L3VPN network with multiple CE devices connected to the same VRF at PE1. The VRF is configured with a RD: RD1, and uses "per next hop" label allocation mode to advertise the CE routes to L3VPN core. PE1 is eSN and PE2 is iSN for the IP traffic in consideration. CE1..CE4 advertise route for same prefix Pfx1 in BGP service families IPv4 Unicast (AFI/SAFI: 1/1) negotiated on the BGP sessions between the CEs and PE1. BGP L3VPN address family (AFI/SAFI: 1/128) is negotiated between PE1 - RR1, and RR1 - PE2. RR1 reflects the BGP routes between PE1 and PE2 with next hop unchanged.

PE1 would typically advertise to RR1 only the best path for prefix Pfx1 out of routes received from CE1..CE4. Using per CE RD or Addpath for L3VPN family may allow PE1 to advertise all CE routes to the RR, with an increase in RIB scale. This model increases RIB scale by "n" times, where ’n’ is the number of CEs.

When MNH is used in this network, PE1 advertises a single BGP L3VPN route for prefix Pfx1, which contains a MNH attribute with "n" next hops, each carrying the label pointing towards a particular CE, using Payload Encapsulation Info TLV (Section 5.3.3) along with the Endpoint Identifier TLV (Section 5.3.1)

This allows the network to direct traffic to a specific CE, and better load-balance traffic in the provider network, with entropy provided by the per CE VPN labels, without increasing RIB scale.

A.4. Signaling Desired Forwarding Behavior for MPLS Upstream labels at Receiving Node

In Upstream label allocation case, the receiving speaker’s forwarding-state can be controlled by the advertising speaker, thus enabling a standardized API to program desired MPLS forwarding-state at the receiving node. This is described in the [I-D.draft-kaliraj-bess-bgp-sig-private-mpls-labels-06]

A.5. Load Balancing over EBGP Parallel Links

Consider N parallel links between two EBGP speakers. There are different models possible to do load balancing over these links:N single-hop EBGP sessions over the N links. Interface addresses are used as next-hops. N copies of the RIB are exchanged to form N-way ECMP paths. The routes advertised on the N sessions can be attached with Link bandwidth community to perform weighted ECMP. 1 multi-hop EBGP session between loopback addresses, reachable via static route over the N links. Loopback addresses are used as next-hops. 1 copy of the RIB is exchanged with loopback address as nexthop. And a static route can be configured to the loopback address to perform desired N-way ECMP path. M loopbacks are configured in this model, to achieve M different load balancing schemes: ECMP, weighted ECMP, Fast-reroute enabled paths etc. 1 multi-hop EBGP session between loopback addresses, reachable via static route over the N links. Interface addresses are used as next-hops, without using additional loopbacks. 1 copy of the RIB is exchanged with MNH attribute to form N-way ECMP paths, weighted ECMP, Fast-reroute backup paths etc. BFD may be used to these directly connected BGP nexthops to detect liveness.

* N single-hop EBGP sessions over the N links. Interface addresses are used as next-hops. N copies of the RIB are exchanged to form N-way ECMP paths. The routes advertised on the N sessions can be attached with Link bandwidth community to perform weighted ECMP.

* 1 multi-hop EBGP session between loopback addresses, reachable via static route over the N links. Loopback addresses are used as next-hops. 1 copy of the RIB is exchanged with loopback address as nexthop. And a static route can be configured to the loopback address to perform desired N-way ECMP path. M loopbacks are configured in this model, to achieve M different load balancing schemes: ECMP, weighted ECMP, Fast-reroute enabled paths etc.

* 1 multi-hop EBGP session between loopback addresses, reachable via static route over the N links. Interface addresses are used as next-hops, without using additional loopbacks. 1 copy of the RIB is exchanged with MNH attribute to form N-way ECMP paths, weighted ECMP, Fast-reroute backup paths etc. BFD may be used to these directly connected BGP nexthops to detect liveness.

A.6. Flowspec Routes with Multiple "Redirect IP" next hops

There are existing protocol machinery which can benefit from the ability of MNH to clearly specify fallback behavior when multiple nexthops are involved. One example is the scenario described in [I-D.draft-ietf-idr-flowspec-redirect-ip] Sec 3 where multiple Redirect-to-IP nexthop addresses exist for a Flowspec prefix. In such a scenario, the receiving speakers may redirect the traffic to different nexthops, based on variables like IGP-cost. If instead, the MNH was used to specify the redirect-to-IP nexthop, then the order of preference between the different nexthops can be clearly specified using one flowspec route carrying a MNH containing those different nexthop-addresses specifying the desired preference-order. Such that, irrespective of IGP-cost, the receiving speakers will redirect the flow towards the same traffic collector device.

A.7. Color-Only Resolution next hop

Another existing protocol machinery that manufactures nexthop addresses from overloaded extended color community is specified in [RFC9256] Sec 8.8.1. In a way, the color field is overloaded to carry one anycast BGP next-hop with pre-specified fallback options. This approach gives us only two next-hops to play with. The ’BGP nexthop address’ and the ’Color-only nexthop’

Instead, the MNH could be used to achieve the same result with more flexibility. Multiple BGP nexthops can be carried, each resolving over a desired Transport class (Color), and with customizable fallback order. And the solution will work for non-SRTE networks as-well.

A.8. Problems with Multihomed PEs Protecting Each Other

```
                         +-----[PE11]    [RR1]
                         |       |         |
                         |       +-------+ |
                      [CE1]              [P1]---[PE2]-----[CE2]
                         |       +-------+
                         |       |
                         +-----[PE12]

                 203.0.113.11                        203.0.113.22
                           ----  Traffic direction ---->
```

Figure 34: Example Topology with Multihomed PEs Protecting Each Other

In a MPLS network, a router CE1 may be multihomed to two PEs PE11 and PE12. The PEs may re-advertise routes received from CE1 to the IBGP core with self as nexthop and a MPLS Label. The PEs may also protect failure of primary path to router CE1 by using the IBGP path via the other multihomed PE as a backup path. The advertised label has forwarding state installed with both primary and backup paths

Following problems are possible in this scenario:

A.8.1. Label oscillation between Multihomed PEs

If "per nexthop" label allocation mechanism is used at the PEs, label allocation oscillation may occur when PE11 advertises a new label to PE12. Reception of a new label results in change of nexthop at PE12, as the received label is used as backup/repair nexthop leg, and per-nexthop label allocation is in use. Thus a new label is allocated by PE12 and advertised. And when this new label is received by the PE11, it allocates a new label in turn. This process repeats.

This problem can happen for either SAFI 4 or SAFI 128 routes.

This oscillation can be stopped only if the primary path label allocated by a PE does not depend on the primary path label advertised by other PE. A PE needs to be able to advertise multiple labels, one for use as primary path and another to be used as repair path by the receiver.

MNH attribute allows to advertise a Repair forwarding path label using Section 5.1.2 in addition to Primary forwarding path label using Section 5.1.1. This avoids this label oscillation problem.

A.8.2. Forwarding loop between Multihomed PEs

If "per VRF table" label allocation mechanism is used at the PEs, a temporary forwarding loop may between PE11, PE12 in events like the CE1 router going down, which will cause both PE11-CE1 and PE12-CE1 links go down.

PE11 will forward traffic coming from PE2 on the backup path towards PE12. That packet will perform IP lookup in the VRF at PE12, which will result in the packet getting forwarded over the backup/repair path towards PE11. This loop will persist until global convergence completes, with the PEs send BGP withdrawals for the routes received from CE1 to each other.

This problem can happen for SAFI 128 routes.

This loop can also be avoided if the a PE can advertise a 'Repair path label' that does not include the primary path label advertised by other PE. A PE needs to be able to advertise multiple labels, one for use as primary path and another to be used as repair path by the receiver.

MNH attribute allows to advertise a Repair forwarding path label using Section 5.1.2 in addition to Primary forwarding path label using Section 5.1.1. This avoids this forwarding loop problem also.

A.9. Signaling Intent over PE-CE Attachment Circuit

BGP CT specifies procedures for Intent Driven Service Mapping in a service provider network, and defines 'Transport Class' construct to represent an Intent.

It may be desirable to allow a CE device to indicate in the data packet it sends what treatment it desires (the Intent) when the packet is forwarded within the provider network.

This section describes the mechanisms that enable such signaling. These procedures use existing AFIs 1 or 2, and service families (SAFI 1) on the PE-CE attachment circuit, with a new BGP attribute.

```
                               ---Gold----->
                     [CE1]-----[PE1]---[P]----[PE2]-----[CE2]
                               ---Bronze--->
               203.0.113.11                             203.0.113.22
                          ----  Traffic direction ---->
```

Figure 35: Example Topology with PE-CE Links

A.9.1. Using DSCP in MultiNexthop Attribute

Such an indication can be in form of DSCP code point ([RFC2474]) in the IP header.

In RFC2474, a Forwarding Class Selector maps to a PHB (Per-hop Behavior). The Transport Class construct is a PHB at transport layer.

Let PE1 be configured to map DSCP1 to Gold Transport class, and DSCP2 to Bronze Transport class. Based on the DSCP code point received on the IP traffic from CE1, PE1 forwards the IP packet over a Gold or Bronze tunnel. Thus, the forwarding is not based on just the destination IP address, but also the DSCP code point. This is known as Class Based Forwarding (CBF). Today CBF is configured at the PE1 device roles and CE1 doesn’t receive any indication in BGP signaling regarding what DSCP code points are being offered by the provider network.

With a BGP MultiNexthop Attribute attached to a AFI/SAFI 1/1 service route, it is possible to extend the PE-CE BGP signaling (if used) to communicate such information to the CE1. In the preceding example, the MNH contains two Next hop Legs, described by two Forwarding Instruction TLVs. Each Next hop Leg contains PE1’s peering self address in Endpoint Identifier TLV (Section 5.3.1), the color Gold or Bronze encoded in the Transport class ID TLV (Section 5.3.2.2), and associated DSCP code point indicating Gold or Bronze transport class encoded in the Payload Encapsulation Info TLV (Section 5.3.3). This allows the CE to discover what transport classes exist in the provider network, and which DSCP codepoint to encode so that traffic is forwarded using the desired transport class in the provided network.

A.9.2. MPLS-enabled CE

If the PE-CE link is MPLS enabled, a distinct MPLS label can also be used to express Intent in data packets from CE. Enabling MPLS forwarding on PE-CE links comes with some security implications. This section gives details on these aspects.

Consider the ingress PE1 receiving a VPN prefix RD:Pfx1 received with VPN label VL1, next hop as PE2 and a mapping community containing TC1 as ’Transport class ID’. PE1 can allocate a MPLS Label PVL1 for the tuple "VPN Label, PNH Address, Transport class ID" and advertise to CE1.

Label PVL1 may identifies a service function at any node in the network, e.g. a Firewall device or egress node PE2. And, for the same service prefix, a distinct label may be advertised to different CEs, such that incoming traffic from different CEs to the same service prefix can be diverted to a distinct devices in the network for further processing. This provides Ingress Peer Engineering control to the network.

PE1 installs a MPLS FIB route for PVL1 with next hop as "Swap VL1, Push TL1 towards PE2". TL1 is the BGP CT label received for the tuple 'PE2, TC1'. In forwarding, when MPLS packet with label PVL1 is received from CE1, PVL1 Swaps to label VL1 and pushes the BGP CT label TL1. PE1 advertises the label "PVL1" in the MNH to CE1. PE1 forwards based on MPLS label without performing any IP lookup. This allows for PE1 to be a low IP FIB device and still support CBF by using MPLS Label inferred PHB. The number of MPLS Labels consumed at PE1 for this approach will be proportional to the number of Service functions and Intents that are exposed to CE1.

A BGP MultiNexthop Attribute is attached to a AFI/SAFI 1/1 service route to convey the MPLS Label information to CE1. In the preceding example, the MNH contains two Next hop Legs, described by two Forwarding Instruction TLVs. Each Next hop Leg contains PE1's peering self address in Endpoint Identifier TLV (Section 5.3.1), the color Gold or Bronze encoded in the Transport class ID TLV (Section 5.3.2.2), and associated MPLS Label "PVL1" or "PVL2" encoded in the Payload Encapsulation Info TLV (Section 5.3.3). This allows the CE to discover what transport classes exist in the provider network, and which MPLS Label to encode so that traffic is forwarded using the desired transport class.

A.9.2.1. Secure MPLS Forwarding on Inter-AS Link

The MPLS enabled PE-CE attachment circuit is considered connecting to an untrusted domain. Such interfaces can be secured against MPLS label spoofing by a walled garden approach using "MPLS context tables".

The PE1-CE1 interface can be confined to a specific MPLS context table "A" corresponding to the BGP peer. Such that only the routes for labels advertised to CE1 are installed in MPLS context table "A".

This ensures that if CE1 sends MPLS packet with a label that was not advertised to the CE1, the packet will be dropped.

Furthermore, the routes for labels PVL1, PVL2 installed in MPLS context table "A" can match on 'Bottom of stack' bit being 'one', ensuring a MPLS packet is accepted from CE1 only if it has no more than one label in the label stack.

However, the PE itself may not be able to perform any checks based on inner payload in the MPLS packet since it performs label swap forwarding. Such inner payload based checks may be offloaded to a downstream node that forwards and processes inner payload, e.g. a IP FIB router. These security aspects should be considered when using MPLS enabled CE devices.

A.10. 4PE - Signal MPLS Label for IPv4 Unicast routes

This section describes how MNH can be used to signal MPLS explicit null label in AFI/SAFI: 1/1 routes in a pure IPv6 core environment, to achieve 4PE.

```
                                     [RR1]
                                       .
                                       |
                                       __
                                    _.(  )..
          [CE1] - [PE1]-[P1]    ..  (_  v6  _) .. [PE2] - [CE2]
                                     (._..)


                        <---- Traffic Direction ----
                  P1: PHP node.
                  PE1: Egress PE.
```

Figure 36: 4PE Network with Pure IPv6 Core

Figure 36 shows a 4PE network with pure IPv6 core, PE1 is the egress PE connected to penultimate hop node P1. PE1 to PE2 have some IPv6 core tunneling protocol like LDPv6. When PE1 has advertised Implicit Null label in LDPv6, some implementations of P1 may not be able to forward the inner IPv4 payload to PE1.

To solve this problem, PE1 needs to signal IPv4 Explicit NULL Label (Special Label 0) to PE2. PE2 will push this IPv4 Explicit NULL Label received in the MNH on the AFI/SAFI:1/1 route. Such that P1 does a MPLS Label swap operation and does not need to look into inner payload.

MNH can be used by PE1 on a AFI/SAFI: 1/1 route, to advertise the IPv4 Explicit Null label for the IPv4 Unicast service route. MPLS Label is encoded in the Payload Encapsulation Info TLV (Section 5.3.3).

This allows the network to provide clear separation of service and transport routes, and not overloading AFI/SAFI: 1/4 to carry the IPv4 service routes. Not mixing service and transport routes improves security and manageability aspects of the network.

An egress PE may not need to advertise IPv4 Explicit Null label for the IPv4 service route, if it does UHP label in LDPv6. This model using MNH provides a homogenous service layer (AFI/SAFI: 1/1) that accommodates differences in requirement of different PE and P routers. Only the PEs which are connected to P nodes that cannot handle the PHP situation need to advertise Label using MNH. The service layer is kept consistent in the network, and can seamlessly extend to multiple domains without needing redistribution between AFI/SAFIs.

Not mixing service and transport routes improves security and manageability aspects of the network.

Authors' Addresses

Kaliraj Vairavakkalai
HPE
1133 Innovation Way
Sunnyvale, CA 94089
United States of America
Email: kaliraj.vairavakkalai@hpe.com

Minto Jeyananth
HPE
1133 Innovation Way
Sunnyvale, CA 94089
United States of America
Email: jeyananth-minto.jeganathan@hpe.com

Mohan Nanduri
Microsoft
1 Microsoft Way
Redmond, WA 98052
United States of America
Email: mohannanduri@microsoft.com

Avinash Reddy
AT&T
400 W Plano Pkwy
Plano, TX 75075
United States of America
Email: ar977m@att.com