Workgroup: PCE Working Group
Internet-Draft: draft-ietf-pce-multipath-02
Published: 17 October 2021
Intended Status: Standards Track
Expires: 20 April 2022
Authors: M. Koldychev          S. Sivabalan
         Cisco Systems, Inc.   Ciena Corporation
         T. Saad                  V. Beeram
         Juniper Networks, Inc.   Juniper Networks, Inc.
         H. Bidgoli   B. Yadav   S. Peng
         Nokia        Ciena      Huawei Technologies

# PCEP Extensions for Signaling Multipath Information

## Abstract

Path computation algorithms are not limited to return a single optimal path. Multiple paths may exist that satisfy the given objectives and constraints. This document defines a mechanism to encode multiple paths for a single set of objectives and constraints. This is a generic PCEP mechanism, not specific to any path setup type or dataplane. The mechanism is applicable to both stateless and stateful PCEP.

## Status of This Memo

## Copyright Notice

## Table of Contents

## 1. Introduction

Path Computation Element (PCE) Communication Protocol (PCEP) [RFC5440] enables the communication between a Path Computation Client (PCC) and a Path Control Element (PCE), or between two PCEs based on the PCE architecture [RFC4655].

PCEP Extensions for the Stateful PCE Model [RFC8231] describes a set of extensions to PCEP that enable active control of Multiprotocol Label Switching Traffic Engineering (MPLS-TE) and Generalized MPLS (GMPLS) tunnels. [RFC8281] describes the setup and teardown of PCE-initiated LSPs under the active stateful PCE model, without the need for local configuration on the PCC, thus allowing for dynamic centralized control of a network.

PCEP Extensions for Segment Routing [RFC8664] specifies extensions to the Path Computation Element Protocol (PCEP) that allow a stateful PCE to compute and initiate Traffic Engineering (TE) paths, as well as for a PCC to request a path subject to certain constraint(s) and optimization criteria in SR networks.

Segment Routing Policy for Traffic Engineering [I-D.ietf-spring-segment-routing-policy] details the concepts of SR Policy and approaches to steering traffic into an SR Policy. In particular, it describes the SR candidate-path as a collection of one or more Segment-Lists. The current PCEP standards only allow for signaling of one Segment-List per Candidate-Path. PCEP extension to support Segment Routing Policy Candidate Paths [I-D.ietf-pce-segment-routing-policy-cp] specifically avoids defining how to signal multipath information, and states that this will be defined in another document.

This document defines the required extensions that allow the signaling of multipath information via PCEP.

## 2. Terminology

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

### 2.1. Terms and Abbreviations

The following terms are used in this document:

PCEP Tunnel:

The object identified by the PLSP-ID, see [I-D.koldychev-pce-operational] for more details.

## 3. Motivation

This extension is motivated by the use-cases described below.

### 3.1. Signaling Multiple Segment-Lists of an SR Candidate-Path

The Candidate-Path of an SR Policy is the unit of report/update in PCEP, see [I-D.ietf-pce-segment-routing-policy-cp]. Each Candidate-Path can contain multiple Segment-Lists and each Segment-List is encoded by one ERO. However, each PCEP LSP can contain only a single ERO, which prevents us from encoding multiple Segment- Lists within the same SR Candidate-Path.

With the help of the protocol extensions defined in this document, this limitation is overcome.

### 3.2. Splitting of Requested Bandwidth

A PCC may request a path with 80 Gbps of bandwidth, but all links in the network have only 50 Gbps capacity. The PCE can return two paths, that can together carry 80 Gbps. The PCC can then equally or unequally split the incoming 80 Gbps of traffic among the two paths. Section 4.3 introduces a new TLV that carries the path weight that allows for distribution of incoming traffic on to the multiple paths.

### 3.3. Providing Backup path for Protection

It is desirable for the PCE to compute and signal to the PCC a backup path that is used to protect a primary path within the multipaths in a given LSP.

Note that [RFC8745] specify the Path Protection association among LSPs. The use of [RFC8745] with multipath is out of scope of this document and is for future study.

When multipath is used, a backup path may protect one or more primary paths. For this reason, primary and backup path identifiers are needed to indicate which backup path(s) protect which primary path(s). Section 4.4 introduces a new TLV that carries the required information.

## 4. Protocol Extensions

### 4.1. Multipath Capability TLV

We define the MULTIPATH-CAP TLV that MAY be present in the OPEN object and/or the LSP object. The purpose of this TLV is two-fold:

1. From PCC: it tells how many multipaths per PCEP Tunnel, the PCC can install in forwarding.

2. From PCE: it tells that the PCE supports this standard and how many multipaths per PCEP Tunnel, the PCE can compute.

Only the first instance of this TLV can be processed, subsequent instances SHOULD be ignored.

Section 5 specify the usage of this TLV with Open message (within the OPEN object) and other PCEP messages (within the LSP object).

```
 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             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Number of Multipaths      |           Flags         |O|B|W|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 1: MULTIPATH-CAP TLV format

Type: TBD1 for "MULTIPATH-CAP" TLV.

Length: 4.

Number of Multipaths: the maximum number of multipaths per PCEP Tunnel. The value 0 indicates unlimited number.

W-flag: whether MULTIPATH-WEIGHT-TLV is supported.

B-flag: whether MULTIPATH-BACKUP-TLV is supported.

O-flag: whether MULTIPATH-OPPDIR-PATH-TLV is supported.

### 4.2. Path Attributes Object

We define the PATH-ATTRIB object that is used to carry per-path information and to act as a separator between several ERO/RRO objects in the <intended-path>/<actual-path> RBNF element. The PATH-ATTRIB object always precedes the ERO/RRO that it applies to. If multiple ERO/RRO objects are present, then each ERO/RRO object MUST be preceded by an PATH-ATTRIB object that describes it.

The PATH-ATTRIB Object-Class value is TBD2.

The PATH-ATTRIB Object-Type value is 1.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Flags                           |  O  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Path ID                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                          Optional TLVs                        ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2: PATH-ATTRIB object format

O (Operational - 3 bits): operational state of the path, same values as the identically named field in the LSP object [RFC8231].

Path ID: 4-octet identifier that identifies a path (encoded in the ERO/RRO) within the set of multiple paths under the PCEP LSP. Value 0x0 is reserved to indicate the absense of a Path ID. The value of 0x0 MAY be used when this Path is not being referenced by any other path and the allocation of a Path ID is not necessary.

TLVs that may be included in the PATH-ATTRIB object are described in the following sections. Other optional TLVs could be defined by future documents to be included within the PATH-ATTRIB object body.

## 4.3. Multipath Weight TLV

We define the MULTIPATH-WEIGHT TLV that MAY be present in the PATH-ATTRIB object.

```
 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            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             Weight                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 3: MULTIPATH-WEIGHT TLV format

Type: TBD3 for "MULTIPATH-WEIGHT" TLV.

Length: 4.

Weight: weight of this path within the multipath, if W-ECMP is desired. The fraction of flows a specific ERO/RRO carries is derived from the ratio of its weight to the sum of all other multipath ERO/RRO weights.

When the MULTIPATH-WEIGHT TLV is absent from the PATH-ATTRIB object, or the PATH-ATTRIB object is absent from the <intended-path>/<actual-path>, then the Weight of the corresponding path is taken to be "1".

### 4.4. Multipath Backup TLV

This document introduces a new MULTIPATH-BACKUP TLV that is optional and can be present in the PATH-ATTRIB object.

This TLV is used to indicate the presence of a backup path that is used for protection in case of failure of the primary path. The format of the MULTIPATH-BACKUP TLV is:

```
 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            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       Backup Path Count       |             Flags           |B|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Backup Path ID 1                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Backup Path ID 2                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                              ...                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Backup Path ID n                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 4: MULTIPATH-BACKUP TLV format

Type: TBD4 for "MULTIPATH-BACKUP" TLV

Length: 4 + (N * 4) (where N is the Backup Path Count)

Backup Path Count: Number of backup path(s).

B: If set, indicates a pure backup path. This is a path that only carries rerouted traffic after the protected path fails. If this flag is not set, or if the MULTIPATH-BACKUP TLV is absent, then the path is assumed to be primary that carries normal traffic.

Backup Path ID(s): a series of 4-octet identifier(s) that identify the backup path(s) in the set that protect this primary path.

### 4.5. Multipath Opposite Direction Path TLV

This document introduces a new MULTIPATH-OPPDIR-PATH TLV that is optional and can be present in the PATH-ATTRIB object.

This TLV is used to indicate whether the given path is a forward path or a reverse path in its PCEP Tunnel, as well as give information about the opposite-direction path(s) of the given path.

```
 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             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        Reserved (MBZ)         |            Flags       |L|N|R|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                 Opposite Direction Path ID                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 5: MULTIPATH-OPPDIR-PATH TLV format

Type: TBD9 for "MULTIPATH-OPPDIR-PATH" TLV

Length: 16.

R (Reverse path): If set, indicates this path is reverse, i.e., it originates on the Tunnel destination and terminates on the Tunnel source (usually the PCC headend itself). Paths with this flag set MUST NOT be installed into forwarding, they serve only informational purposes.

N (Node co-routed): If set, indicates this path is node co-routed with its opposite direction path, specified in this TLV. Two opposite direction paths are node co-routed if they traverse the same nodes, but MAY traverse different links.

L (Link co-routed): If set, indicates this path is link co-routed with its opposite directions path, specified in this TLV. Two opposite direction paths are link co-routed if they traverse the same links (but in the opposite directions).

Opposite Direction Path ID: Identifies a path that goes in the opposite direction to this path. If no such path exists, then this field MUST be set to 0x0, which is reserved to indicate the absense of a Path ID.

Multiple instances of this TLV present in the same PATH-ATTRIB object indicate that there are multiple opposite-direction paths corresponding to the given path. This allows for many-to-many relationship among the paths of two opposite direction Tunnels.

Whenever path A references another path B as being the opposite-direction path, then path B typically also reference path A as its own opposite-direction path.

See Section 7.4 for an example of usage.

### 4.6. Composite Candidate Path

SR Policy Architecture [I-D.ietf-spring-segment-routing-policy] defines the concept of a Composite Candidate Path. Unlike a Non-Composite Candidate Path, which contains Segment Lists, the Composite Candidate Path contains Colors of other policies. The traffic that is steered into a Composite Candidate Path is split among the policies that are identified by the Colors contained in the Composite Candidate Path. The split can be either ECMP or UCMP by adjusting the weight of each color in the Composite Candidate Path, in the same manner as the weight of each Segment List in the Non-Composite Candidate Path is adjusted.

To signal the Composite Candidate Path, we make use of the COLOR TLV, defined in [I-D.peng-pce-te-constraints]. For a Composite Candidate Path, the COLOR TLV is included in the PATH-ATTRIB Object, thus allowing each Composite Candidate Path to do ECMP/UCMP among SR Policies or Tunnels identified by its constituent Colors. Only one COLOR TLV SHOULD be included into the PATH-ATTRIB object. If multiple COLOR TLVs are contained in the PATH-ATTRIB object, only the first one MUST be processed and the others SHOULD be ignored.

An empty ERO object MUST be included as per the existing RBNF, i.e., ERO MUST contain no sub-objects. If the head-end receives a non-empty ERO, then it MUST send PCError message with Error-Type 19 ("Invalid Operation") and Error-Value = TBD8 ("Non-empty path").

See Section 7.3 for an example of the encoding.

## 5. Operation

When the PCC wants to indicate to the PCE that it wants to get multipaths for a PCEP Tunnel, instead of a single path, it can do (1) or both (1) and (2) of the following:

(1) Send the MULTIPATH-CAP TLV in the OPEN object during session establishment. This applies to all PCEP Tunnels on the PCC, unless overridden by PCEP Tunnel specific information.

(2) Additionally send the MULTIPATH-CAP TLV in the LSP object for a particular PCEP Tunnel in the PCRpt or PCReq message. This applies to the specified PCEP Tunnel and overrides the information from the OPEN object.

When PCE computes the path for a PCEP Tunnel, it MUST NOT return more multipaths than the corresponding value of "Number of Multipaths" from the MULTIPATH-CAP TLV. If this TLV is absent (from both OPEN and LSP objects), then the "Number of Multipaths" is assumed to be 1.

If the PCE supports this standard, then it MUST include the MULTIPATH-CAP TLV in the OPEN object. This tells the PCC that it can report multiple ERO/RRO objects per PCEP Tunnel to this PCE. If the PCE does not include the MULTIPATH-CAP TLV in the OPEN object, then the PCC MUST assume that the PCE does not support this standard and fall back to reporting only a single ERO/RRO. The PCE MUST NOT include MULTIPATH-CAP TLV in the LSP object in any other PCEP message towards the PCC and the PCC MUST ignore it if received.

The Path ID of each ERO/RRO MUST be unique within that LSP. If a PCEP speaker detects that there are two paths with the same Path ID, then the PCEP speaker SHOULD send PCError message with Error-Type = 1 ("Reception of an invalid object") and Error-Value = TBD5 ("Conflicting Path ID").

### 5.1. Signaling Multiple Paths for Loadbalancing

The PATH-ATTRIB object can be used to signal multiple path(s) and indicate (un)equal loadbalancing amongst the set of multipaths. In this case, the PATH-ATTRIB is populated for each ERO as follows:

1. The PCE assigns a unique Path ID to each ERO path and populates it inside the PATH-ATTRIB object. The Path ID is unique within the context of a PLSP or PCEP Tunnel.

2. The MULTIPATH-WEIGHT TLV MAY be carried inside the PATH-ATTRIB object. A weight is populated to reflect the relative loadshare that is to be carried by the path. If the MULTIPATH-WEIGHT is not carried inside a PATH-ATTRIB object, the default weight 1 MUST be assumed when computing the loadshare.

3. The fraction of flows carried by a specific primary path is derived from the ratio of its weight to the sum of all other multipath weights.

### 5.2. Signaling Multiple Paths for Protection

The PATH-ATTRIB object can be used to describe a set of backup path(s) protecting a primary path within a PCEP Tunnel. In this case, the PATH-ATTRIB is populated for each ERO as follows:

1. The PCE assigns a unique Path ID to each ERO path and populates it inside the PATH-ATTRIB object. The Path ID is unique within the context of a PLSP or PCEP Tunnel.

2. The MULTIPATH-BACKUP TLV MUST be added inside the PATH-ATTRIB object for each ERO that is protected. The backup path ID(s) are populated in the MULTIPATH-BACKUP TLV to reflect the set of backup path(s) protecting the primary path. The Length field and Backup Path Number in the MULTIPATH-BACKUP are updated according to the number of backup path ID(s) included.

3. The MULTIPATH-BACKUP TLV MAY be added inside the PATH-ATTRIB object for each ERO that is unprotected. In this case, MULTIPATH-BACKUP does not carry any backup path IDs in the TLV. If the path acts as a pure backup - i.e. the path only carries rerouted traffic after the protected path(s) fail- then the B flag MUST be set.

Note that if a given path has the B-flag set, then there MUST be some other path within the same LSP that uses the given path as a backup. If this condition is violated, then the PCEP speaker SHOULD send a PCError message with Error-Type = 10 ("Reception of an invalid object") and Error-Value = TBD6 ("No primary path for pure backup").

Note that a given PCC may not support certain backup combinations, such as a backup path that is itself protected by another backup path, etc. If a PCC is not able to implement a requested backup scenario, the PCC SHOULD send a PCError message with Error-Type = 19 ("Invalid Operation") and Error-Value = TBD7 ("Not supported path backup").

## 6. PCEP Message Extensions

The RBNF of PCReq, PCRep, PCRpt, PCUpd and PCInit messages currently use a combination of <intended-path> and/or <actual-path>. As specified in Section 6.1 of [RFC8231], <intended-path> is represented by the ERO object and <actual-path> is represented by the RRO object:

```
<intended-path> ::= <ERO>

<actual-path> ::= <RRO>
```

In this standard, we extend these two elements to allow multiple ERO/RRO objects to be present in the <intended-path>/<actual-path>:

```
   <intended-path> ::= (<ERO>|
                       (<PATH-ATTRIB><ERO>)
                       [<intended-path>])


   <actual-path> ::= (<RRO>|
                      (<PATH-ATTRIB><RRO>)
                      [<actual-path>])
```

## 7. Examples

### 7.1. SR Policy Candidate-Path with Multiple Segment-Lists

Consider the following sample SR Policy, taken from [I-D.ietf-spring-segment-routing-policy].

```
        SR policy POL1 <headend, color, endpoint>
            Candidate-path CP1 <protocol-origin = 20, originator =
    100:1.1.1.1, discriminator = 1>
                Preference 200
                Weight W1, SID-List1 <SID11...SID1i>
                Weight W2, SID-List2 <SID21...SID2j>
            Candidate-path CP2 <protocol-origin = 20, originator =
    100:2.2.2.2, discriminator = 2>
                Preference 100
                Weight W3, SID-List3 <SID31...SID3i>
                Weight W4, SID-List4 <SID41...SID4j>
```

As specified in [I-D.ietf-pce-segment-routing-policy-cp], CP1 and CP2 are signaled as separate state-report elements and each has a unique PLSP-ID, assigned by the PCC. Let us assign PLSP-ID 100 to CP1 and PLSP-ID 200 to CP2.

The state-report for CP1 can be encoded as:

```
<state-report> =
        <LSP PLSP_ID=100>
        <ASSOCIATION>
        <END-POINT>
        <PATH-ATTRIB Path_ID=1 <WEIGHT-TLV Weight=W1>>
        <ERO SID-List1>
        <PATH-ATTRIB Path_ID=2 <WEIGHT-TLV Weight=W2>>
        <ERO SID-List2>
```

The state-report for CP2 can be encoded as:

```
<state-report> =
        <LSP PLSP_ID=200>
        <ASSOCIATION>
        <END-POINT>
        <PATH-ATTRIB Path_ID=1 <WEIGHT-TLV Weight=W3>>
        <ERO SID-List3>
        <PATH-ATTRIB Path_ID=2 <WEIGHT-TLV Weight=W4>>
        <ERO SID-List4>
```

The above sample state-report elements only specify the minimum mandatory objects, of course other objects like SRP, LSPA, METRIC, etc., are allowed to be inserted.

Note that the syntax

```
<PATH-ATTRIB Path_ID=1 <WEIGHT-TLV Weight=W1>>
```

, simply means that this is PATH-ATTRIB object with Path ID field set to "1" and with a MULTIPATH-WEIGHT TLV carrying weight of "W1".

### 7.2. Two Primary Paths Protected by One Backup Path

Suppose there are 3 paths: A, B, C. Where A,B are primary and C is to be used only when A or B fail. Suppose the Path IDs for A, B, C are respectively 1, 2, 3. This would be encoded in a state-report as:

```
<state-report> =
        <LSP>
        <ASSOCIATION>
        <END-POINT>
        <PATH-ATTRIB Path_ID=1 <BACKUP-TLV B=0, Backup_Paths=[3]>>
        <ERO A>
        <PATH-ATTRIB Path_ID=2 <BACKUP-TLV B=0, Backup_Paths=[3]>>
        <ERO B>
        <PATH-ATTRIB Path_ID=3 <BACKUP-TLV B=1, Backup_Paths=[]>>
        <ERO C>
```

Note that the syntax

```
<PATH-ATTRIB Path_ID=1 <BACKUP-TLV B=0, Backup_Paths=[3]>>
```

, simply means that this is PATH-ATTRIB object with Path ID field set to "1" and with a MULTIPATH-BACKUP TLV that has B-flag cleared and contains a single backup path with Backup Path ID of 3.

### 7.3. Composite Candidate Path

Consider the following Composite Candidate Path, taken from [I-D.ietf-spring-segment-routing-policy].

```
SR policy POL100 <headend = H1, color = 100, endpoint = E1>
     Candidate-path CP1 <protocol-origin = 20, originator =
100:1.1.1.1, discriminator = 1>
         Preference 200
         Weight W1, SR policy <color = 1>
         Weight W2, SR policy <color = 2>
```

This is signaled in PCEP as:

```
<LSP PLSP_ID=100>
    <ASSOCIATION>
    <END-POINT>
    <PATH-ATTRIB Path_ID=1
        <WEIGHT-TLV Weight=W1>
        <COLOR-TLV Color=1>>
    <ERO (empty)>
    <PATH-ATTRIB Path_ID=2
        <WEIGHT-TLV Weight=W2>
        <COLOR-TLV Color=2>>
    <ERO (empty)>
```

### 7.4. Opposite Direction Tunnels

Consider the two opposite-direction SR Policies between end-points H1 and E1.

```
SR policy POL1 <headend = H1, color, endpoint = E1>
    Candidate-path CP1
        Preference 200
        Bidirectional Association = A1
        SID-List = <H1,M1,M2,E1>
        SID-List = <H1,M3,M4,E1>
    Candidate-path CP2
        Preference 100
        Bidirectional Association = A2
        SID-List = <H1,M5,M6,E1>
        SID-List = <H1,M7,M8,E1>

SR policy POL2 <headend = E1, color, endpoint = H1>
    Candidate-path CP1
        Preference 200
        Bidirectional Association = A1
        SID-List = <E1,M2,M1,H1>
        SID-List = <E1,M4,M3,H1>
    Candidate-path CP2
        Preference 100
        Bidirectional Association = A2
        SID-List = <E1,M6,M5,H1>
```

The state-report for POL1, CP1 can be encoded as:

```
 <state-report> =
     <LSP PLSP_ID=100>
     <BIDIRECTIONAL ASSOCIATION = A1>
     <PATH-ATTRIB PathID=1
         <OPPDIR-PATH-TLV R-flag=0 OppositePathID=3>>
     <ERO <H1,M1,M2,E1>>
     <PATH-ATTRIB PathID=2
         <OPPDIR-PATH-TLV R-flag=0 OppositePathID=4>>
     <ERO <H1,M3,M4,E1>>
     <PATH-ATTRIB PathID=3
         <OPPDIR-PATH-TLV R-flag=1 OppositePathID=1>>
     <ERO <E1,M2,M1,H1>>
     <PATH-ATTRIB PathID=4
         <OPPDIR-PATH-TLV R-flag=1 OppositePathID=2>>
     <ERO <E1,M4,M3,H1>>
```

The state-report for POL1, CP2 can be encoded as:

```
 <state-report> =
     <LSP PLSP_ID=200>
     <BIDIRECTIONAL ASSOCIATION = A2>
     <PATH-ATTRIB PathID=1
         <OPPDIR-PATH-TLV R-flag=0 OppositePathID=3>>
     <ERO <H1,M5,N6,E1>>
     <PATH-ATTRIB PathID=2
         <OPPDIR-PATH-TLV R-flag=0 OppositePathID=0>>
     <ERO <H1,M7,M8,E1>>
     <PATH-ATTRIB PathID=3
         <OPPDIR-PATH-TLV R-flag=1 OppositePathID=1>>
     <ERO <E1,M6,M5,H1>>
```

The state-report for POL2, CP1 can be encoded as:

```
 <state-report> =
     <LSP PLSP_ID=100>
     <BIDIRECTIONAL ASSOCIATION = A1>
     <PATH-ATTRIB PathID=1
         <OPPDIR-PATH-TLV R-flag=0 OppositePathID=3>>
     <ERO <E1,M2,M1,H1>>
     <PATH-ATTRIB PathID=2
         <OPPDIR-PATH-TLV R-flag=0 OppositePathID=4>>
     <ERO <E1,M4,M3,H1>>
     <PATH-ATTRIB PathID=3
         <OPPDIR-PATH-TLV R-flag=1 OppositePathID=1>>
     <ERO <H1,M1,M2,E1>>
     <PATH-ATTRIB PathID=4
         <OPPDIR-PATH-TLV R-flag=1 OppositePathID=2>>
     <ERO <H1,M3,M4,E1>>
```

The state-report for POL2, CP2 can be encoded as:

```
 <state-report> =
     <LSP PLSP_ID=200>
     <BIDIRECTIONAL ASSOCIATION = A2>
     <PATH-ATTRIB PathID=1
         <OPPDIR-PATH-TLV R-flag=0 OppositePathID=3>>
     <ERO <E1,M6,M5,H1>>
     <PATH-ATTRIB PathID=2
         <OPPDIR-PATH-TLV R-flag=1 OppositePathID=0>>
     <ERO <H1,M7,M8,E1>>
     <PATH-ATTRIB PathID=3
         <OPPDIR-PATH-TLV R-flag=1 OppositePathID=1>>
     <ERO <H1,M5,N6,E1>>
```

## 8. IANA Considerations

### 8.1. PCEP Object

IANA is requested to make the assignment of a new value for the existing "PCEP Objects" registry as follows:

| Object-Class Value | Name | Object-Type Value | Reference |
|---|---|---|---|
| TBD2 | PATH-ATTRIB | 1 | This document |

### 8.2. PCEP TLV

IANA is requested to make the assignment of a new value for the existing "PCEP TLV Type Indicators" registry as follows:

| TLV Type Value | TLV Name | Reference |
|---|---|---|
| TBD1 | MULTIPATH-CAP | This document |
| TBD3 | MULTIPATH-WEIGHT | This document |
| TBD4 | MULTIPATH-BACKUP | This document |
| TBD9 | MULTIPATH-OPPDIR-PATH | This document |

### 8.3. PCEP-Error Object

IANA is requested to make the assignment of a new value for the existing "PCEP-ERROR Object Error Types and Values" sub-registry of the PCEP Numbers registry for the following errors:

| Error-Type | Error-Value | Reference |
|---|---|---|
| 10 | TBD5 - Conflicting Path ID | This document |
| 10 | TBD6 - No primary path for pure backup | This document |
| 19 | TBD7 - Not supported path backup | This document |
| 19 | TBD8 - Non-empty path | This document |

### 8.4. Flags in the Multipath Capability TLV

IANA is requested to create a new sub-registry to manage the Flag field of the MULTIPATH-CAP TLV, called "Flags in MULTIPATH-CAP TLV".

| Bit | Description | Reference |
|---|---|---|
| 0-12 | Unassigned | This document |
| 13 | 0-flag: support for processing MULTIPATH-OPPDIR-PATH TLV | This document |
| 14 | B-flag: support for processing MULTIPATH-BACKUP TLV | This document |
| 15 | W-flag: support for processing MULTIPATH-WEIGHT TLV | This document |

### 8.5. Flags in the Path Attribute Object

IANA is requested to create a new sub-registry to manage the Flag field of the PATH-ATTRIBUTE object, called "Flags in PATH-ATTRIBUTE Object".

| Bit | Description | Reference |
|---|---|---|
| 0-12 | Unassigned | This document |
| 13-15 | O-flag: Operational state | This document |

### 8.6. Flags in the Multipath Backup TLV

IANA is requested to create a new sub-registry to manage the Flag field of the MULTIPATH-BACKUP TLV, called "Flags in MULTIPATH-BACKUP TLV".

| Bit | Description | Reference |
|---|---|---|
| 0-14 | Unassigned | This document |
| 15 | B-flag: Pure backup | This document |

### 8.7. Flags in the Multipath Opposite Direction Path TLV

IANA is requested to create a new sub-registry to manage the flag fields of the MULTIPATH-OPPDIR-PATH TLV, called "Flags in the MULTIPATH-OPPDIR-PATH TLV".

| Bit | Description | Reference |
|---|---|---|
| 0-12 | Unassigned | This document |
| 13 | L-flag: Link co-routed | This document |
| 14 | N-flag: Node co-routed | This document |
| 15 | R-flag: Reverse path | This document |

## 9. Security Considerations

None at this time.

## 10. Acknowledgement

Thanks to Dhruv Dhody for ideas and discussion.

## 11. Contributors

Andrew Stone
Nokia

Email: andrew.stone@nokia.com

Gyan Mishra
Verizon Inc.

Email: gyan.s.mishra@verizon.com

## 12. References

### 12.1. Normative References

**[I-D.ietf-pce-segment-routing-policy-cp]**
Koldychev, M., Sivabalan, S., Barth, C., Peng, S., and H. Bidgoli, "PCEP extension to support Segment Routing Policy Candidate Paths", Work in Progress, Internet-Draft, draft-ietf-pce-segment-routing-policy-cp-05, 23 May 2021, <https://www.ietf.org/archive/id/draft-ietf-pce-segment-routing-policy-cp-05.txt>.

**[I-D.ietf-spring-segment-routing-policy]**
Filsfils, C., Talaulikar, K., Voyer, D., Bogdanov, A., and P. Mattes, "Segment Routing Policy Architecture", Work in Progress, Internet-Draft, draft-ietf-spring-segment-routing-policy-13, 28 May 2021, <https://

www.ietf.org/archive/id/draft-ietf-spring-segment-routing-policy-13.txt>.

**[I-D.koldychev-pce-operational]**
Koldychev, M., Sivabalan, S., Peng, S., Achaval, D., and H. Kotni, "PCEP Operational Clarification", Work in Progress, Internet-Draft, draft-koldychev-pce-operational-04, 19 August 2021, <https://www.ietf.org/archive/id/draft-koldychev-pce-operational-04.txt>.

**[I-D.peng-pce-te-constraints]** Peng, S., Xiong, Q., Qin, F., Koldychev, M., and S. Sivabalan, "PCE TE Constraints", Work in Progress, Internet-Draft, draft-peng-pce-te-constraints-06, 11 July 2021, <https://www.ietf.org/archive/id/draft-peng-pce-te-constraints-06.txt>.

**[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>.

**[RFC5440]** Vasseur, JP., Ed. and JL. Le Roux, Ed., "Path Computation Element (PCE) Communication Protocol (PCEP)", RFC 5440, DOI 10.17487/RFC5440, March 2009, <https://www.rfc-editor.org/info/rfc5440>.

**[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>.

**[RFC8231]** Crabbe, E., Minei, I., Medved, J., and R. Varga, "Path Computation Element Communication Protocol (PCEP) Extensions for Stateful PCE", RFC 8231, DOI 10.17487/RFC8231, September 2017, <https://www.rfc-editor.org/info/rfc8231>.

**[RFC8281]** Crabbe, E., Minei, I., Sivabalan, S., and R. Varga, "Path Computation Element Communication Protocol (PCEP) Extensions for PCE-Initiated LSP Setup in a Stateful PCE Model", RFC 8281, DOI 10.17487/RFC8281, December 2017, <https://www.rfc-editor.org/info/rfc8281>.

**[RFC8664]** Sivabalan, S., Filsfils, C., Tantsura, J., Henderickx, W., and J. Hardwick, "Path Computation Element Communication Protocol (PCEP) Extensions for Segment Routing", RFC 8664, DOI 10.17487/RFC8664, December 2019, <https://www.rfc-editor.org/info/rfc8664>.

### 12.2. Informative References

**[RFC4655]**
Farrel, A., Vasseur, J.-P., and J. Ash, "A Path Computation Element (PCE)-Based Architecture", RFC 4655, DOI 10.17487/RFC4655, August 2006, <https://www.rfc-editor.org/info/rfc4655>.

**[RFC8745]** Ananthakrishnan, H., Sivabalan, S., Barth, C., Minei, I., and M. Negi, "Path Computation Element Communication Protocol (PCEP) Extensions for Associating Working and Protection Label Switched Paths (LSPs) with Stateful PCE", RFC 8745, DOI 10.17487/RFC8745, March 2020, <https://www.rfc-editor.org/info/rfc8745>.

## Authors' Addresses

Mike Koldychev
Cisco Systems, Inc.

Email: mkoldych@cisco.com

Siva Sivabalan
Ciena Corporation

Email: ssivabal@ciena.com

Tarek Saad
Juniper Networks, Inc.

Email: tsaad@juniper.net

Vishnu Pavan Beeram
Juniper Networks, Inc.

Email: vbeeram@juniper.net

Hooman Bidgoli
Nokia

Email: hooman.bidgoli@nokia.com

Bhupendra Yadav
Ciena

Email: byadav@ciena.com

Shuping Peng
Huawei Technologies

Email: pengshuping@huawei.com