
Stream: Internet Engineering Task Force (IETF)
RFC: [10066](#)
Category: Standards Track
Published: October 2026
ISSN: 2070-1721
Authors: M. Koldychev, Ed. S. Sidor, Ed.
Ciena Corporation Cisco Systems.

RFC 10066

Path Computation Element Communication Protocol (PCEP) Extensions for Signaling Multipath Information

Abstract

This document defines Path Computation Element Communication Protocol (PCEP) extensions to signal multipath information, associating multiple forwarding paths with a single PCEP Label Switched Path (LSP) to allow load balancing and redundancy across diverse paths. The extensions are generic and applicable to both stateless and stateful PCEP and are designed to be reusable for future path types. Their primary application is Segment Routing Policy (SR Policy), where a Candidate Path can contain multiple segment lists: current PCEP extensions for SR Policy only allow signaling of a single segment list per Candidate Path. These extensions enable multipath capabilities such as weighted or equal-cost load balancing across paths.

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at <https://www.rfc-editor.org/info/rfc10066>.

Copyright Notice

Copyright (c) 2026 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction	3
1.1. Requirements Language	4
1.2. Terminology	4
2. Motivation	5
2.1. Signaling Multiple Segment Lists of an SR Candidate Path	5
2.2. Splitting of Requested Bandwidth	5
2.3. Reverse Path Information	5
3. Protocol Extensions	6
3.1. PATH-ATTRIB Object	6
3.2. METRIC Object	7
3.3. MULTIPATH-WEIGHT TLV	7
3.4. MULTIPATH-OPPDIR-PATH TLV	7
3.5. Composite Candidate Path	9
3.5.1. Per-Flow Candidate Path	9
4. Operation	10
4.1. Capability Negotiation	10
4.1.1. Multipath Capability TLV	10
4.2. Path ID	12
4.3. Signaling Multiple Paths for Load Balancing	13
4.4. Signaling Opposite-Direction Path Information	13
5. PCEP Message Extensions	14
6. IANA Considerations	16
6.1. PCEP Object	16
6.2. PCEP TLV	17

6.3. PCEP-Error Object	17
6.4. Flags in the MULTIPATH-CAP TLV	18
6.5. Flags in the PATH-ATTRIB Object	18
6.6. Flags in the MULTIPATH-OPPDIR-PATH TLV	19
6.7. Flags in the MULTIPATH-FORWARD-CLASS TLV	19
7. Security Considerations	19
8. Operational Considerations	20
8.1. Control of Function and Policy	20
8.2. Information and Data Models	20
8.3. Liveness Detection and Monitoring	20
8.4. Verify Correct Operations	20
8.5. Requirements on Other Protocols	20
8.6. Impact on Network Operations	21
9. References	21
9.1. Normative References	21
9.2. Informative References	22
Appendix A. Examples	23
A.1. SR Policy Candidate Path with Multiple Segment Lists	23
A.2. Composite Candidate Path	25
A.3. Opposite-Direction Tunnels	25
Acknowledgements	28
Contributors	28
Authors' Addresses	29

1. Introduction

Segment Routing Policy (SR Policy) for Traffic Engineering [[RFC9256](#)] details the concepts of Segment Routing 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

specifications only allow for signaling of one segment list per Candidate Path. The PCEP extension to support Segment Routing Policy Candidate Paths [\[RFC9862\]](#) specifically kept the signaling of multiple segment lists outside its scope.

This document defines the required extensions that allow the signaling of multipath information via PCEP. Although these extensions are motivated by the SR Policy use case, they are also applicable to other technologies. For SR Policy, support for [\[RFC9862\]](#) is a prerequisite for using the multipath extensions defined in this document with SR Policy Candidate Paths.

1.1. Requirements Language

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

1.2. Terminology

The following terms are used in this document:

ECMP:

Equal-Cost Multipath, equally distributing traffic among multiple paths/links, where each path/link gets the same share of traffic as others.

W-ECMP:

Weighted Equal-Cost Multipath, unequally distributing traffic among multiple paths/links, where some paths/links get more traffic than others.

PLSP:

PCE Label Switched Path, a path or set of paths computed or controlled by the PCE. In the context of SR Policy, a PLSP corresponds to a Candidate Path.

Segment List:

An ordered list of segments that defines a forwarding path in Segment Routing, as defined in [\[RFC9256\]](#). In PCEP for SR Policy, each segment list is encoded as an ERO or RRO.

ERO:

Explicit Route Object, defined in [\[RFC5440\]](#), encodes an explicit path. In the context of SR Policy, an ERO encodes a segment list.

RRO:

Record Route Object, defined in [\[RFC5440\]](#), encodes the actual signaled path. In the context of SR Policy, an RRO reports a segment list.

Path:

In the context of this document, a path refers to a single forwarding path encoded in an ERO or RRO. For SR Policy, a path corresponds to a segment list. The mechanisms defined in this document use the generic term "path" to allow applicability beyond SR Policy.

LSP:

Label Switched Path. The base PCEP specification [RFC4655] originally defined the use of the PCE architecture for MPLS and GMPLS networks with LSPs instantiated using the RSVP-TE signaling protocol [RFC3209]. Over time, support for additional path setup types such as SRv6 has been introduced [RFC9603]. The term "LSP" is used extensively in PCEP specifications and, while the multipath extensions defined in this document are applicable beyond SR Policy, in the context of PCEP for SR Policy [RFC9862], an LSP object represents an SR Policy Candidate Path, which may be an SRv6 path (still represented using the LSP object as specified in [RFC8231]). A single LSP may contain multiple paths (segment lists).

SR Policy:

A Segment Routing Policy as defined in [RFC9256], identified by the tuple (headend, color, endpoint).

2. Motivation

This extension is motivated by the use cases described below.

2.1. Signaling Multiple Segment Lists of an SR Candidate Path

The Candidate Path of an SR Policy is the unit of signaling in PCEP [RFC9862]. A single Candidate Path can consist of multiple segment lists. Each segment list is represented by an ERO. In existing PCEP specifications, a PCEP LSP object is associated with exactly one ERO. This restriction prevents the encoding of multiple segment lists (i.e., multiple EROs) within the single LSP.

2.2. Splitting of Requested Bandwidth

For example, a Path Computation Client (PCC) may request a path whose bandwidth demand cannot be accommodated on any single link along an available path in the network. The PCE can return multiple paths that together can carry the requested bandwidth. The PCC can then equally or unequally split the incoming traffic among these paths.

2.3. Reverse Path Information

PCEP extensions for Associated Bidirectional LSPs [RFC9059] defines a mechanism in PCEP to associate two opposite-direction SR Policy Candidate Paths. However, within each Candidate Path there can be multiple segment lists, and [RFC9059] does not define a mechanism to specify mapping between segment lists of the forward and reverse Candidate Paths.

Certain applications, such as Circuit Style SR Policy [CS-SR-POLICY], require the knowledge of reverse paths per segment list, not just per Candidate Path. For example, when the headend knows the reverse segment list for each forward segment list, then Performance Measurement (PM) / Bidirectional Forwarding Detection (BFD) can run a separate session on every segment list, by imposing a double stack (forward stack followed by a reverse stack) onto the packet. If the reverse segment list is co-routed with the forward segment list, then the PM/BFD session would traverse the same links in the forward and reverse directions, thus allowing detection of link/node failures in both directions.

3. Protocol Extensions

3.1. PATH-ATTRIB Object

This document defines the PATH-ATTRIB object that is used to carry per-path information and to act as a separator between EROs/RROs in the <intended-path>/<actual-path> Routing Backus-Naur Form (RBNF) [RFC5511] element. The PATH-ATTRIB object always precedes the ERO or RRO that it applies to. If multiple EROs or RROs are present, then each ERO or RRO **MUST** be preceded by a PATH-ATTRIB object that describes it.

The PATH-ATTRIB Object-Class value is 45.

The PATH-ATTRIB Object-Type value is 1.

The format of the PATH-ATTRIB object is shown in Figure 1.

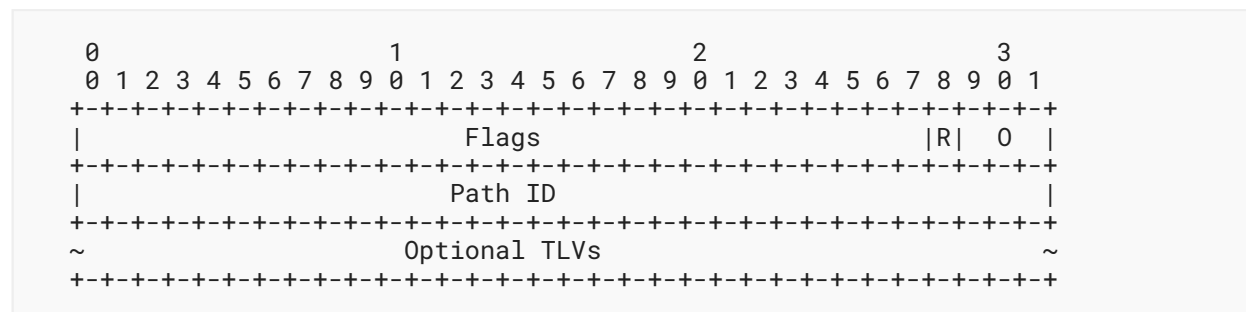


Figure 1: PATH-ATTRIB Object Format

Flags (32 bits):

O (Operational - 3 bits): Operational state of the path, same values as the identically named field in the LSP object [RFC8231]. The relationship between the per-path Operational state and the LSP-level Operational state in the LSP object is outside the scope of this document; for SR Policy, the Candidate Path validity criterion is defined in Section 2.8 of [RFC9256].

R (Reverse - 1 bit): Indicates the path flows in the reverse direction, i.e., it originates on the LSP destination and terminates on the LSP source (usually the PCC headend itself). Paths with this flag set are not installed in the forwarding plane for load-balancing purposes, but **MAY** be used by the PCC for operations such as Performance Measurement (PM) or Bidirectional Forwarding Detection (BFD).

Unassigned bits **MUST** be set to 0 on transmission and **MUST** be ignored on receipt.

Path ID (32 bits): 4-octet identifier that identifies a path (encoded in the ERO/RRO) within the set of multiple paths under the PCEP LSP. See Section 4.2 for details.

Optional TLVs: Variable length field that can contain zero or more Type-Length-Value (TLV) elements that carry additional per-path information. The specific TLVs that can be included are defined in subsequent sections of this document.

3.2. METRIC Object

The PCEP METRIC object can continue to be used at the LSP level to describe properties of the overall LSP. Mechanisms for encoding per-path metrics (e.g., a separate METRIC for each path) are outside the scope of this document and would require further extensions.

3.3. MULTIPATH-WEIGHT TLV

A new MULTIPATH-WEIGHT TLV is optional in the PATH-ATTRIB object.

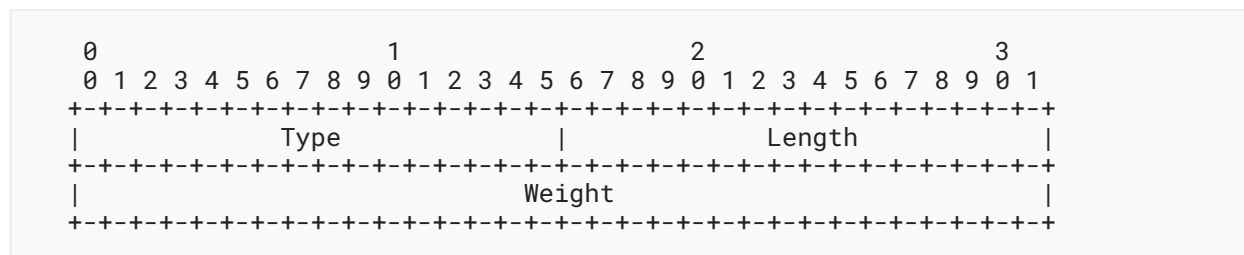


Figure 2: MULTIPATH-WEIGHT TLV Format

Type (16 bits): 61 for "MULTIPATH-WEIGHT" TLV.

Length (16 bits): **MUST** be set to 4.

Weight (32 bits): unsigned integer weight of the path described by the enclosing PATH-ATTRIB object, if W-ECMP is desired. The fraction of flows that a specific ERO/RRO carries is derived from the ratio of its weight to the sum of the weights of all paths in the multipath (including that path); see [Section 4.3](#) for details. For SR Policy, if the Weight value is 0, the corresponding segment list is declared invalid per [Section 5.1](#) of [\[RFC9256\]](#) and carries no traffic. If all paths within an LSP have Weight 0, the sum of weights is zero, making the candidate path invalid per [Section 2.11](#) of [\[RFC9256\]](#). Signaling a zero-weight path alongside paths with non-zero weights can be used to drain traffic from a path while retaining its forwarding instructions.

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 assumed to be 1.

3.4. MULTIPATH-OPPDIR-PATH TLV

A new MULTIPATH-OPPDIR-PATH TLV is optional in the PATH-ATTRIB object. Multiple instances of the TLV are allowed in the same PATH-ATTRIB object. Each TLV instance identifies one opposite-direction path for the path described by this PATH-ATTRIB object. This provides per-path-level opposite-direction mapping within an LSP. In the context of SR Policy, this corresponds

to per-Segment-List mapping within a Candidate Path, complementing the Candidate Path level bidirectional association defined in [RFC10067], which also describes the usage of this TLV in the context of associated bidirectional SR Paths. See [Section 4.4](#) for operational details.

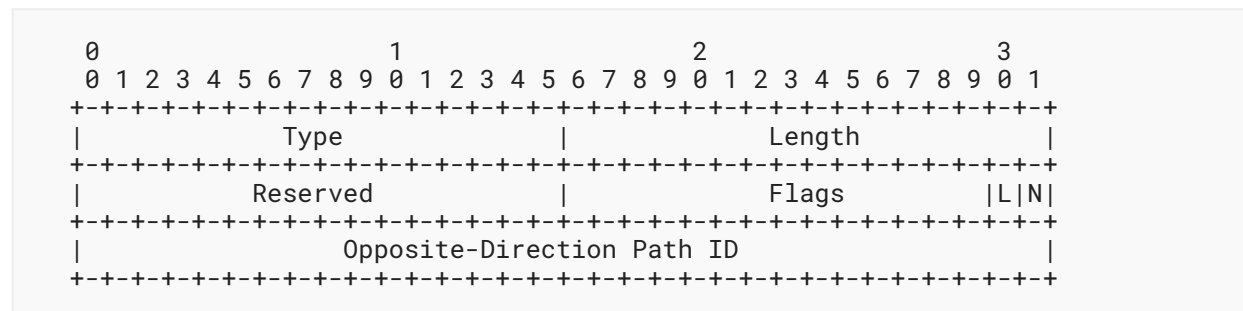


Figure 3: MULTIPATH-OPPDIR-PATH TLV Format

Type (16 bits): 63 for "MULTIPATH-OPPDIR-PATH" TLV

Length (16 bits): **MUST** be set to 8.

Reserved: This field **MUST** be set to zero on transmission and **MUST** be ignored on receipt.

Flags (16 bits):

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. If not set, the paths are not guaranteed to be node co-routed (they may or may not traverse the same set of nodes).

L (Link co-routed): If set, indicates this path is link co-routed with its opposite-direction path, specified in this TLV. Two opposite-direction paths are link co-routed if they traverse the same links (but in opposite directions). Link co-routing implies node co-routing; therefore, it is not necessary to set the N-flag when the L-flag is set.

The terms "link" and "node" used by the N-flag and L-flag refer to the links and nodes as represented in the topology that the PCE uses for path computation. The exact nature of a link is determined by the PCE's topology view and is out of scope of this document. The same notion of "link" and "node" **MUST** be used consistently for both the forward and the opposite-direction path when evaluating co-routing; a path and its opposite-direction path are represented in the same topology view.

Unassigned bits **MUST** be set to 0 on transmission and **MUST** be ignored on receipt.

Opposite-Direction Path ID (32 bits): References the Path ID field (see [Section 4.2](#)) of a PATH-ATTRIB object that identifies a path going in the opposite direction to this path. The value 0 is reserved and **MUST NOT** be used in this field. If no opposite-direction path exists, the MULTIPATH-OPPDIR-PATH TLV **MUST NOT** be included in the PATH-ATTRIB object (see [Section](#)

4.4). If a PCEP speaker receives a MULTIPATH-OPPDIR-PATH TLV with Opposite-Direction Path ID set to 0, it **MUST** send a PCErr message with Error-Type = 19 ("Invalid Operation") and Error-Value = 37 ("Invalid opposite-direction path mapping").

3.5. Composite Candidate Path

SR Policy Architecture [RFC9256] defines the concept of a composite candidate path. A regular SR Policy candidate path outputs traffic to a set of segment lists, while an SR Policy composite candidate path outputs traffic recursively to a set of SR Policies on the same headend. In PCEP, the composite candidate path still consists of PATH-ATTRIB objects, but ERO is replaced by color of the recursively used SR Policy.

To signal the composite candidate path, this document makes use of the COLOR TLV, defined in [RFC9863]. For a composite candidate path, the COLOR TLV is included in the PATH-ATTRIB object, thus allowing each composite candidate path to do ECMP/W-ECMP among SR Policies identified by its constituent colors. To achieve W-ECMP, the MULTIPATH-WEIGHT TLV (Section 3.3) is included alongside the COLOR TLV in each PATH-ATTRIB object. Each PATH-ATTRIB object identifies one constituent color of the composite candidate path. If multiple COLOR TLVs are contained in the PATH-ATTRIB object, the first one is processed and the others **MUST** be ignored.

An ERO **MUST** be included as per the existing RBNF; this ERO **MUST** contain no subobjects. This empty ERO serves as a placeholder to maintain compatibility with existing implementations based on the RBNF defined in [RFC8231]. If the headend receives a non-empty ERO for a composite candidate path, it **MUST** send a PCErr message with Error-Type = 19 ("Invalid Operation") and Error-Value = 21 ("Non-empty path").

See Appendix A.2 for an example of the encoding.

3.5.1. Per-Flow Candidate Path

Per-Flow Candidate Path builds on the concept of the composite candidate path. Each Path in a Per-Flow Candidate Path is assigned a 3-bit forwarding class value, which allows Quality-of-Service (QoS) classified traffic to be steered depending on the forwarding class.

A new MULTIPATH-FORWARD-CLASS TLV is optional in the PATH-ATTRIB object.

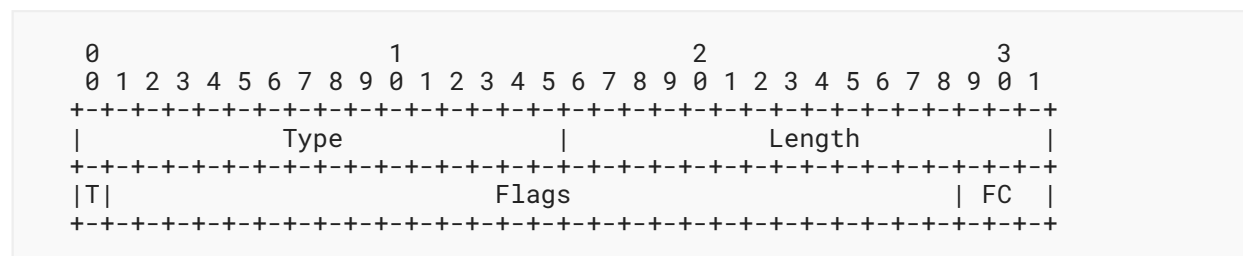


Figure 4: MULTIPATH-FORWARD-CLASS TLV Format

Type (16 bits): 76 for "MULTIPATH-FORWARD-CLASS" TLV.

Length (16 bits): **MUST** be set to 4.

Flags (32 bits):

T (1 bit): MPLS Traffic Class (TC) type. This is the most significant bit of the Flags field. When set, indicates that the FC value is derived from the MPLS TC bits as described in [Section 8.6](#) of [RFC9256]. When not set, the interpretation of the FC value is reserved for future use; a future specification **MAY** define a wider, right-aligned classifier value occupying additional low-order bits of the Flags field (e.g., for a non-MPLS data plane).

FC (3 bits): Forwarding class value. When the T-flag is set, this carries the MPLS TC-based forwarding class value as defined in [Section 8.6](#) of [RFC9256]. This value is given by the QoS classifier to traffic entering the given Candidate Path. Different classes of traffic that enter the given Candidate Path can be differentially steered into different colors. The FC field allows up to 8 different forwarding classes (values 0-7). The semantics of specific FC values are significant at the headend node (PCC) that implements the SR Policy and are determined by that node's local QoS policy or configuration. Coordination of FC value meanings between PCEP peers (e.g., through out-of-band configuration management or operational procedures) is outside the scope of this document.

Unassigned bits **MUST** be set to 0 on transmission and **MUST** be ignored on receipt.

4. Operation

4.1. Capability Negotiation

4.1.1. Multipath Capability TLV

A new MULTIPATH-CAP TLV is defined. This TLV **MAY** be present in the OPEN object during PCEP session establishment. It **MAY** also be present in the LSP object for each individual LSP from the PCC.

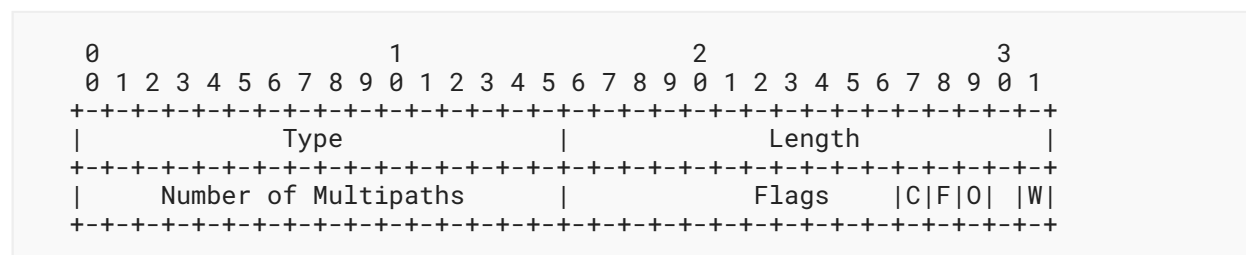


Figure 5: MULTIPATH-CAP TLV Format

Type (16 bits): 60 for "MULTIPATH-CAP" TLV.

Length (16 bits): **MUST** be set to 4.

Number of Multipaths (16 bits): When sent from a PCC, it indicates how many forward primary multipaths the PCC can install in the forwarding plane. From a PCE, it indicates how many forward primary multipaths the PCE can compute. This value is per-LSP when carried in the LSP object; the effective value governing a given LSP is the LSP object value if present; otherwise, it is the OPEN object value (see below). This count does not include reverse paths (R-flag=1, where the R-flag is defined in [Section 3.1](#)), which are not installed in forwarding for load-balancing purposes. Therefore, the total number of PATH-ATTRIB objects in an LSP may exceed this value when reverse paths are also signaled. The value 0 indicates an unlimited number.

Flags (16 bits):

W-flag: Indicates whether MULTIPATH-WEIGHT TLV is supported. This flag covers the use of MULTIPATH-WEIGHT for both regular and composite candidate paths.

O-flag: In the OPEN object, this flag indicates whether the MULTIPATH-OPPDIR-PATH TLV is supported. In the LSP object, this flag indicates that opposite-direction path information is requested or provided for that specific LSP. When set by the PCC (in PCRpt/PCReq), it requests that the PCE provide reverse path information. When set by the PCE (in PCInitiate/PCUpd/PCRep), it indicates the PCE is providing or will provide reverse path information. In both cases, the PCE **SHOULD** provide the reverse path information if it is able to. For PCC-Initiated LSPs, if the PCC has not set the O-flag in the MULTIPATH-CAP TLV of the LSP object, the PCE **SHOULD NOT** include reverse path information in the corresponding response. If the PCC receives unsolicited reverse path information, it **MAY** ignore it.

F-flag: Indicates whether the MULTIPATH-FORWARD-CLASS TLV is supported.

C-flag: Indicates whether the composite candidate path ([Section 3.5](#)) is supported, including the use of the COLOR TLV in the PATH-ATTRIB object.

Unassigned bits **MUST** be set to 0 on transmission and **MUST** be ignored on receipt.

Note that F-flag and C-flag can be set independently for capability negotiation purposes. While Per-Flow Candidate Path ([Section 3.5.1](#)) builds on top of composite candidate path, the F-flag reflects whether the MULTIPATH-FORWARD-CLASS TLV is supported, and the C-flag reflects whether composite candidate path signaling is supported. A peer that supports Per-Flow Candidate Path **MUST** set both the C-flag and F-flag. Note that the F-flag is defined independently of the C-flag to allow for future use cases that may use the MULTIPATH-FORWARD-CLASS TLV for purposes other than Per-Flow Candidate Path; in such cases, the F-flag **MAY** be set without the C-flag.

When a PCE computes an LSP path, it **MUST NOT** return more forward multipaths than the minimum of the effective "Number of Multipaths" values of both the PCE and PCC. The effective value for a given LSP is determined by the per-LSP MULTIPATH-CAP TLV in the LSP object if present; otherwise, it defaults to the value from the MULTIPATH-CAP TLV in the OPEN object. This ensures the PCE does not exceed either its own computation capability or the PCC's

installation capability. If this TLV is absent from both OPEN and LSP objects, the PCEP speaker does not support multipath and the behavior is consistent with existing PCEP specifications, where a single path is associated with each LSP.

If a PCC receives more paths than it advertised support for, it **MUST** send a PCErr message with Error-Type = 19 ("Invalid Operation") and Error-Value = 36 ("Unsupported multipath capability"). When a PCE receives a PCRpt message containing more paths than its own computation capability, it **MUST NOT** treat this as an error condition; PCRpt conveys PCC-reported Operational state and is not subject to the PCE's computation limits.

The PCC **MAY** also include the MULTIPATH-CAP TLV in the LSP object for each individual LSP, to specify per-LSP values. The PCC **MUST NOT** include this TLV in the LSP object if the TLV was not present in the OPEN objects of both PCEP peers. TLV values in the LSP object override the session default values in the OPEN object. If a PCEP speaker receives a PATH-ATTRIB object but the multipath capability was not successfully negotiated during session establishment, it **MUST** treat this as an error. The PCEP speaker **MUST** send a PCErr message with Error-Type = 10 ("Reception of an invalid object") and Error-Value = 45 ("Unexpected PATH-ATTRIB object").

For example, the PCC includes this TLV in the OPEN object at session establishment, setting "Number of Multipaths" to 4 and "O-flag" to 0. The PCC also includes this TLV in the LSP object for a particular LSP, setting "Number of Multipaths" to 16 and "O-flag" to 1. This indicates that the PCC only wants to receive the reverse path information for that particular LSP and that this LSP can have up to 16 multipaths, while other LSPs can only have up to 4 multipaths.

Additionally, if a PCEP speaker receives a TLV within the PATH-ATTRIB object (such as MULTIPATH-WEIGHT, MULTIPATH-OPPDIR-PATH, or MULTIPATH-FORWARD-CLASS) but the corresponding capability flag was not set in the negotiated MULTIPATH-CAP TLV, it **MUST** treat this as an error. The PCEP speaker **MUST** send a PCErr message with Error-Type = 19 ("Invalid Operation") and Error-Value = 36 ("Unsupported multipath capability").

4.2. Path ID

The Path ID uniquely identifies a Path within the context of an LSP. A single Path ID space is shared among all paths within the LSP, including forward paths and reverse paths (R-flag=1). Path IDs **MUST** be unique across forward and reverse paths within the same LSP. The meaning of "Path" depends on the type of LSP:

- For a regular SR Policy Candidate Path, the Paths within that LSP are the segment lists.
- For a composite candidate path ([Section 3.5](#)), the Paths within that LSP are the constituent SR Policies, each of which is identified by its color (carried in the COLOR TLV within the corresponding PATH-ATTRIB object).

Value 0 indicates an unallocated Path ID. The value of 0 **MAY** be used when this Path is not referenced and the allocation of a Path ID is not necessary.

Path IDs are allocated by the PCEP peer that owns the LSP. If the LSP is delegated to the PCE, then the PCE allocates the Path IDs and sends them in the PCRep/PCUpd/PCInitiate messages. If the LSP is locally computed on the PCC, then the PCC allocates the Path IDs and sends them in the PCReq/PCRpt messages. When LSP delegation changes (e.g., the PCC revokes delegation from the PCE), the new path owner **SHOULD** retain the existing Path IDs to simplify path correlation in the event of redelegation. If the new owner cannot retain them, it **MAY** allocate new Path IDs.

If a PCEP speaker detects that there are two Paths with the same non-zero Path ID, then the PCEP speaker **MUST** send a PCErr message with Error-Type = 10 ("Reception of an invalid object") and Error-Value = 38 ("Conflicting Path ID"). Multiple paths **MAY** have Path ID set to 0, as this value indicates those paths are not referenced and do not require unique identification.

4.3. Signaling Multiple Paths for Load Balancing

The PATH-ATTRIB object can be used to signal multiple paths and indicate equal or unequal load balancing among the set of multipaths. In this case, the PATH-ATTRIB is populated for each ERO as follows:

1. The PCE **MAY** assign a unique Path ID to each ERO path and populate it inside the PATH-ATTRIB object. The Path ID is unique within the context of a PLSP (when non-zero).
2. The PCE **MAY** include the MULTIPATH-WEIGHT TLV inside the PATH-ATTRIB object, populating a weight value to reflect the relative share of traffic to be carried by the path. If the MULTIPATH-WEIGHT is not carried inside a PATH-ATTRIB object, the PCC **MUST** assume the default weight of 1 when computing the traffic share.
3. The PCC derives the fraction of flows carried by a specific primary path from the ratio of its weight to the sum of the weights of all paths in the multipath. For SR Policy, the use of weights for load balancing between segment lists of a Candidate Path is described in [Section 2.11](#) of [RFC9256].

4.4. Signaling Opposite-Direction Path Information

The PATH-ATTRIB object can be used to signal opposite-direction path associations within a PCEP LSP. This capability is used to establish bidirectional path relationships where forward and reverse paths can be explicitly mapped to each other. In this case, the PATH-ATTRIB is populated for each ERO as follows:

1. The PCEP peer (PCC or PCE) allocates a unique Path ID to each path and populates it inside the PATH-ATTRIB object. The Path ID is unique within the context of a PLSP (PCE Label Switched Path).
2. For paths that have opposite-direction counterparts, the MULTIPATH-OPPPDIR-PATH TLV is added to the PATH-ATTRIB object. The Opposite-Direction Path ID field is set to reference the Path ID of the corresponding opposite-direction path.
3. Multiple instances of the MULTIPATH-OPPPDIR-PATH TLV **MAY** be present in the same PATH-ATTRIB object to support many-to-many mappings between forward and reverse paths. This allows a single forward path to map to multiple reverse paths and vice versa. A many-to-

many mapping can occur when a segment list contains Node Segment(s) that traverse parallel links at a midpoint. The reverse of this segment list may require multiple reverse segment lists to cover all the parallel links at the midpoint.

4. The N-flag and L-flag in the MULTIPATH-OPPDIR-PATH TLV **MAY** be set to indicate node co-routing or link co-routing respectively. These flags inform the receiver about the relationship between the forward and reverse paths.

The N-flag and L-flag describe the relationship between the signaled paths as known to the PCEP speaker at the time the message is sent. They do not guarantee that transient local protection, local repair, or local policy actions in the network will preserve node or link co-routing for all forwarded traffic.

5. For paths that have no opposite-direction counterpart, the MULTIPATH-OPPDIR-PATH TLV is omitted from the PATH-ATTRIB object.

Forward paths (R-flag=0) and reverse paths (R-flag=1) are included in the same PCEP LSP, allowing bidirectional relationships to be established in a single message exchange. The opposite-direction path associations **MUST** be symmetric within the same LSP: for each (A -> B) reference expressed via a MULTIPATH-OPPDIR-PATH TLV instance in path A's PATH-ATTRIB object, the corresponding (B -> A) reference **MUST** also be present as a MULTIPATH-OPPDIR-PATH TLV instance in path B's PATH-ATTRIB object. Multiple MULTIPATH-OPPDIR-PATH TLV instances per PATH-ATTRIB object are permitted, supporting many-to-many mappings. Additionally, the R-flags of opposite-direction paths **MUST** have opposite values (one set to 0, the other to 1).

If a PCEP speaker receives an opposite-direction path mapping that is asymmetric or where the R-flags are inconsistent, it **MUST** send a PCErr message with Error-Type = 19 ("Invalid Operation") and Error-Value = 37 ("Invalid opposite-direction path mapping").

See [Appendix A.3](#) for an example of usage.

5. PCEP Message Extensions

The RBNF of PCRpt and PCUpd messages, as defined in [\[RFC8231\]](#), uses a combination of <intended-path> and/or <actual-path>. PCReq and PCRep messages, as defined in [\[RFC5440\]](#) and extended by [\[RFC8231\]](#), directly include ERO and RRO within their respective message structures rather than encapsulating them within <intended-path> or <actual-path> constructs.

As specified in [Section 6.1](#) of [\[RFC8231\]](#), within the context of messages that use these constructs, <intended-path> is represented by the ERO and <actual-path> is represented by the RRO:

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

This document extends [\[RFC8231\]](#) by allowing multiple EROs/RROs to be present in the <intended-path>/<actual-path>:

```

<intended-path> ::= <ERO> |
                  <PATH-ATTRIB><ERO>[<intended-path-multipath>]

<intended-path-multipath> ::= <PATH-ATTRIB><ERO>
                              [<intended-path-multipath>]

<actual-path> ::= <RRO> |
                 <PATH-ATTRIB><RRO>[<actual-path-multipath>]

<actual-path-multipath> ::= <PATH-ATTRIB><RRO>
                           [<actual-path-multipath>]

```

Similarly, this document extends [RFC8281](#) by allowing multiple paths in the PCInitiate message by allowing multiple EROs with their associated path attributes. The PCE-initiated LSP instantiation format is updated to:

```

<PCE-initiated-lsp-instantiation> ::= <SRP>
                                     <LSP>
                                     [<END-POINTS>]
                                     <intended-path>
                                     [<attribute-list>]

```

where <intended-path> follows the recursive definition above, allowing multiple paths to be signaled in a single PCInitiate message. When multiple paths are present, each ERO **MUST** be preceded by a PATH-ATTRIB object that describes it. A single path **MAY** be sent as a bare ERO without PATH-ATTRIB for backward compatibility.

This document also extends the PCRep and PCReq messages, defined in [RFC5440](#), to carry multipath information. Because these messages include the ERO and RRO directly rather than within the <intended-path>/<actual-path> constructs, the extension allows a PATH-ATTRIB object to precede each ERO or RRO that it describes.

In the PCRep message, the <path> element is updated to allow a PATH-ATTRIB object to precede the ERO, with multiple <path> elements in the existing <path-list> used to signal multiple paths:

```

<path> ::= <ERO><attribute-list> |
          <PATH-ATTRIB><ERO><attribute-list>

```

When multiple paths are returned for load balancing, each <path> in the <path-list> **MUST** include a PATH-ATTRIB object preceding its ERO. A single <path> **MAY** be sent as a bare ERO without PATH-ATTRIB for backward compatibility.

In the PCReq message, the existing path reported for reoptimization is updated to allow multiple RROs, each preceded by a PATH-ATTRIB object:

```

<request> ::= <RP>
             <END-POINTS>
             [ <LSP> ]
             [ <LSPA> ]
             [ <BANDWIDTH> ]
             [ <metric-list> ]
             [ <reported-path> ]
             [ <IRO> ]
             [ <LOAD-BALANCING> ]

<reported-path> ::= <RRO>[ <BANDWIDTH> ] |
                   <PATH-ATTRIB><RRO>[ <BANDWIDTH> ]
                   [ <reported-path-multipath> ]

<reported-path-multipath> ::= <PATH-ATTRIB><RRO>[ <BANDWIDTH> ]
                              [ <reported-path-multipath> ]

```

When multiple existing paths are reported, each RRO **MUST** be preceded by a PATH-ATTRIB object. A single reported path **MAY** be sent as a bare RRO without PATH-ATTRIB for backward compatibility.

The LOAD-BALANCING object [RFC5440] and the multipath extensions defined in this document are two different mechanisms for requesting multiple paths and **MUST NOT** be combined. The LOAD-BALANCING object requests a set of independent TE LSPs, each with its own bandwidth, whereas these extensions return multiple paths within a single LSP. Accordingly, a PCReq message **MUST NOT** contain both a LOAD-BALANCING object and a MULTIPATH-CAP TLV (in the LSP object) for the same request. A PCEP speaker that receives such a request **MUST** send a PCErr message with Error-Type = 19 ("Invalid Operation") and Error-Value = 38 ("LOAD-BALANCING and MULTIPATH-CAP TLV in the same request").

The capability negotiation and error handling defined in Section 4 apply equally to the PCReq and PCRep messages. In particular, a PCEP speaker that receives a PATH-ATTRIB object, or a TLV within it, in a PCReq or PCRep message without having negotiated the corresponding capability in the MULTIPATH-CAP TLV **MUST** treat it as an error as specified in Section 4.

The RBNF extensions defined in this section are additive and apply only to sessions where the capabilities from the MULTIPATH-CAP TLV have been successfully negotiated (Section 4). Legacy PCEP peers that do not advertise the MULTIPATH-CAP TLV will not receive PATH-ATTRIB objects introduced by the updated RBNF.

6. IANA Considerations

All IANA actions in this section pertain to the "Path Computation Element Protocol (PCEP) Numbers" registry group, available at <https://www.iana.org/assignments/pcep/>.

6.1. PCEP Object

IANA has allocated the following in the "PCEP Objects" registry (<https://www.iana.org/assignments/pcep/>):

Object-Class Value	Name	Object-Type Value	Reference
45	PATH-ATTRIB	0: Reserved	RFC 10066
		1: PATH-ATTRIB	RFC 10066
		2-15: Unassigned	

Table 1

Object-Type values are managed via the IETF Review policy per [RFC8126].

6.2. PCEP TLV

IANA has allocated the following in the "PCEP TLV Type Indicators" registry (<<https://www.iana.org/assignments/pcep>>):

TLV Type Value	TLV Name	Reference
60	MULTIPATH-CAP	RFC 10066
61	MULTIPATH-WEIGHT	RFC 10066
63	MULTIPATH-OPPDIR-PATH	RFC 10066

Table 2

IANA has allocated the following in the "PCEP TLV Type Indicators" registry:

TLV Type Value	TLV Name	Reference
76	MULTIPATH-FORWARD-CLASS	RFC 10066

Table 3

6.3. PCEP-Error Object

IANA has allocated the following in the "PCEP-ERROR Object Error Types and Values" registry (<<https://www.iana.org/assignments/pcep>>):

Error-Type	Error-Value	Reference
10	38 - Conflicting Path ID	RFC 10066
19	21 - Non-empty path	RFC 10066

Table 4

IANA has allocated the following in the "PCEP-ERROR Object Error Types and Values" registry:

Error-Type	Error-Value	Reference
10	45 - Unexpected PATH-ATTRIB object	RFC 10066
19	36 - Unsupported multipath capability	RFC 10066
19	37 - Invalid opposite-direction path mapping	RFC 10066
19	38 - LOAD-BALANCING and MULTIPATH-CAP TLV in the same request	RFC 10066

Table 5

6.4. Flags in the MULTIPATH-CAP TLV

IANA created the "Flags in MULTIPATH-CAP TLV" registry to manage the Flag field of the MULTIPATH-CAP TLV. New values are to be assigned by IETF Review [[RFC8126](#)].

Bit	Description	Reference
0-10	Unassigned	
11	C-flag: Composite Candidate Path support	RFC 10066
12	F-flag: MULTIPATH-FORWARD-CLASS TLV support	RFC 10066
13	O-flag: MULTIPATH-OPPPDIR-PATH TLV support	RFC 10066
14	Unassigned	
15	W-flag: MULTIPATH-WEIGHT TLV support	RFC 10066

Table 6

6.5. Flags in the PATH-ATTRIB Object

IANA has created the "Flags in PATH-ATTRIB Object" registry to manage the Flag field of the PATH-ATTRIB object. New values are to be assigned by IETF Review [[RFC8126](#)].

Bit	Description	Reference
0-27	Unassigned	
28	R-flag: Reverse path	RFC 10066
29-31	O-flag: Operational state	RFC 10066

Table 7

6.6. Flags in the MULTIPATH-OPPDIR-PATH TLV

IANA has created the "Flags in MULTIPATH-OPPDIR-PATH TLV" registry to manage the Flag field of the MULTIPATH-OPPDIR-PATH TLV. New values are to be assigned by IETF Review [RFC8126].

Bit	Description	Reference
0-13	Unassigned	
14	L-flag: Link co-routed	RFC 10066
15	N-flag: Node co-routed	RFC 10066

Table 8

6.7. Flags in the MULTIPATH-FORWARD-CLASS TLV

IANA has created the "Flags in MULTIPATH-FORWARD-CLASS TLV" registry to manage the Flag field of the MULTIPATH-FORWARD-CLASS TLV. New values are to be assigned by IETF Review [RFC8126].

Bit	Description	Reference
0	T-flag: MPLS TC type	RFC 10066
1-28	Unassigned	
29-31	FC: Forwarding class	RFC 10066

Table 9

7. Security Considerations

The security considerations described in [RFC5440], [RFC8231], [RFC8281], [RFC8664], [RFC9256], [RFC9862], and [RFC9863] are applicable to this specification.

As per [RFC8231], it is **RECOMMENDED** that these PCEP extensions only be activated on authenticated and encrypted sessions across PCEs and PCCs belonging to the same administrative authority, using Transport Layer Security (TLS) [RFC8253] [RFC9916] as per the recommendations and best current practices in [RFC9325].

The multipath extensions defined in this document allow a PCE to signal multiple paths per LSP, which increases the per-LSP state maintained by the PCC. A misbehaving or compromised PCE could exploit this to amplify state at the PCC by maximizing multipath fan out. The "Number of Multipaths" field in the MULTIPATH-CAP TLV provides an upper bound on the number of paths a

PCC will accept per LSP, and operators **SHOULD** configure a non-zero value to limit exposure. The existing PCEP authentication and encryption recommendations (TLS per [RFC8253]) mitigate the risk of unauthorized PCE access.

8. Operational Considerations

All manageability requirements and considerations listed in [RFC5440], [RFC8231], [RFC8664], and [RFC9256] apply to the PCEP protocol extensions defined in this document. In addition, the requirements and considerations listed in this section apply.

8.1. Control of Function and Policy

A PCEP speaker (PCC or PCE) implementation should allow an operator to enable or disable the multipath capabilities advertised in the MULTIPATH-CAP TLV (see [Section 4](#)).

8.2. Information and Data Models

It is expected that a future version of the PCEP YANG module [RFC9826] will be extended to include the PCEP extensions defined in this document.

8.3. Liveness Detection and Monitoring

The mechanisms defined in this document do not introduce any new liveness detection or monitoring requirements in addition to those already defined in [RFC5440] and [RFC8231].

8.4. Verify Correct Operations

In addition to the verification requirements in [RFC5440] and [RFC8231], the following considerations apply:

- An implementation should allow an operator to view the capabilities advertised in the MULTIPATH-CAP TLV by each PCEP peer for a session and for individual LSPs.
- An implementation should allow an operator to view the PATH-ATTRIB object and all its associated TLVs for each path within an LSP. This includes the Path ID, weight, and opposite-direction path associations.
- An implementation should provide a mechanism to log and display the new PCEP errors defined in this document.

8.5. Requirements on Other Protocols

The PCEP extensions defined in this document do not impose any new requirements on other protocols.

8.6. Impact on Network Operations

The mechanisms in this document allow for more complex LSP structures with multiple paths. Network operators should be aware of the potential increase in PCEP message sizes and the additional state that must be maintained by PCEP speakers. The "Number of Multipaths" field in the MULTIPATH-CAP TLV can be used to control the scale of multipath computations and state.

Paths within a single LSP may have significantly different properties, such as MTU, latency, and available bandwidth. When flow-based load balancing is used across such paths, individual flows are hashed to a single path, so per-path MTU or latency divergence is a per-flow concern rather than a per-packet one. However, operators should be aware that advertising weighted load balancing across paths with very different latencies can degrade application performance. Similarly, if paths have different MTUs, the effective MTU for flows on each path will differ; operators should ensure that PMTUD or consistent MTU configuration is in place. For bidirectional applications relying on Performance Measurement (PM) or BFD, note that forward and reverse paths may have asymmetric properties, and implementations should account for this.

PCE implementations should minimize unnecessary churn in multipath state. With multiple paths per LSP, frequent recomputation or weight changes cause flow redistribution across segment lists, which may perturb congestion control state and trigger PMTUD rediscovery on a broader scale than with single-path LSPs.

For composite candidate paths, recursive policy resolution is therefore bounded, as [Section 2.2 of \[RFC9256\]](#) prohibits constituent SR Policies of a composite candidate path from themselves using composite candidate paths.

9. References

9.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [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>>.
- [RFC5511] Farrel, A., "Routing Backus-Naur Form (RBNF): A Syntax Used to Form Encoding Rules in Various Routing Protocol Specifications", RFC 5511, DOI 10.17487/RFC5511, April 2009, <<https://www.rfc-editor.org/info/rfc5511>>.
- [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>>.
- [RFC8253] Lopez, D., Gonzalez de Dios, O., Wu, Q., and D. Dhody, "PCEPS: Usage of TLS to Provide a Secure Transport for the Path Computation Element Communication Protocol (PCEP)", RFC 8253, DOI 10.17487/RFC8253, October 2017, <<https://www.rfc-editor.org/info/rfc8253>>.
- [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>>.
- [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/info/rfc9256>>.
- [RFC9325] Sheffer, Y., Saint-Andre, P., and T. Fossati, "Recommendations for Secure Use of Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)", BCP 195, RFC 9325, DOI 10.17487/RFC9325, November 2022, <<https://www.rfc-editor.org/info/rfc9325>>.
- [RFC9603] Li, C., Ed., Kaladharan, P., Sivabalan, S., Koldychev, M., and Y. Zhu, "Path Computation Element Communication Protocol (PCEP) Extensions for IPv6 Segment Routing", RFC 9603, DOI 10.17487/RFC9603, July 2024, <<https://www.rfc-editor.org/info/rfc9603>>.
- [RFC9862] Koldychev, M., Sivabalan, S., Sidor, S., Barth, C., Peng, S., and H. Bidgoli, "Path Computation Element Communication Protocol (PCEP) Extensions for Segment Routing (SR) Policy Candidate Paths", RFC 9862, DOI 10.17487/RFC9862, October 2025, <<https://www.rfc-editor.org/info/rfc9862>>.
- [RFC9863] Rajagopalan, B., Beeram, V., Peng, S., Koldychev, M., and G. Mishra, "Path Computation Element Protocol (PCEP) Extension for Color", RFC 9863, DOI 10.17487/RFC9863, October 2025, <<https://www.rfc-editor.org/info/rfc9863>>.
- [RFC9916] Dhody, D., Turner, S., and R. Housley, "Updates to the Usage of TLS to Provide a Secure Transport for the Path Computation Element Communication Protocol (PCEP)", RFC 9916, DOI 10.17487/RFC9916, July 2026, <<https://www.rfc-editor.org/info/rfc9916>>.

9.2. Informative References

- [CS-SR-POLICY]** Schmutzer, C., Ed., Ali, Z., Ed., Maheshwari, P., Rokui, R., and A. Stone, "Circuit Style Segment Routing Policy", Work in Progress, Internet-Draft, draft-ietf-spring-cs-sr-policy-17, 12 March 2026, <<https://datatracker.ietf.org/doc/html/draft-ietf-spring-cs-sr-policy-17>>.
- [RFC3209]** Awduche, D., Berger, L., Gan, D., Li, T., Srinivasan, V., and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels", RFC 3209, DOI 10.17487/RFC3209, December 2001, <<https://www.rfc-editor.org/info/rfc3209>>.
- [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>>.
- [RFC8126]** Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <<https://www.rfc-editor.org/info/rfc8126>>.
- [RFC9059]** Gandhi, R., Ed., Barth, C., and B. Wen, "Path Computation Element Communication Protocol (PCEP) Extensions for Associated Bidirectional Label Switched Paths (LSPs)", RFC 9059, DOI 10.17487/RFC9059, June 2021, <<https://www.rfc-editor.org/info/rfc9059>>.
- [RFC9826]** Dhody, D., Ed., Beeram, V., Hardwick, J., and J. Tantsura, "A YANG Data Model for the Path Computation Element Communication Protocol (PCEP)", RFC 9826, DOI 10.17487/RFC9826, September 2025, <<https://www.rfc-editor.org/info/rfc9826>>.
- [RFC10067]** Li, C., Ed., Chen, M., Cheng, W., Gandhi, R., Ed., and Q. Xiong, "Path Computation Element Communication Protocol (PCEP) Extensions for Associated Bidirectional Segment Routing (SR) LSPs", RFC 10067, DOI 10.17487/RFC10067, October 2026, <<https://www.rfc-editor.org/info/rfc10067>>.

Appendix A. Examples

A.1. SR Policy Candidate Path with Multiple Segment Lists

Consider the following sample SR Policy.

```

SR policy POL1 <headend, color, endpoint>
  Candidate Path CP1 <protocol-origin = 20, originator-asn = 100,
                      originator-address = 192.0.2.1,
                      discriminator = 1>
    Preference 200
    Weight W1, SID-List1 <SID11...SID1i>
    Weight W2, SID-List2 <SID21...SID2j>
  Candidate Path CP2 <protocol-origin = 20, originator-asn = 100,
                      originator-address = 198.51.100.1,
                      discriminator = 2>
    Preference 100
    Weight W3, SID-List3 <SID31...SID3i>
    Weight W4, SID-List4 <SID41...SID4j>

```

As specified in [RFC9862], CP1 and CP2 are signaled as separate state-report elements and each has a unique PLSP-ID, assigned by the PCC. For this example, PLSP-ID 100 is assigned to CP1 and PLSP-ID 200 to CP2.

The state-report (as defined in [RFC8231]) for CP1 can be encoded as:

```

<state-report> =
  <LSP PLSP-ID=100>
  <ASSOCIATION>
  <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>
  <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>>

```

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

A.2. Composite Candidate Path

Consider the following composite candidate path.

```
SR policy POL100 <headend = H1, color = 100, endpoint = E1>
  Candidate Path CP1 <protocol-origin = 20, originator-asn = 100,
    originator-address = 192.0.2.1,
    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>
  <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)>
```

A.3. Opposite-Direction Tunnels

Consider the two opposite-direction SR Policies between endpoints 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 Path ID=1 R-flag=0
    <OPPDIR-PATH-TLV OppositePath ID=3>>
  <ERO <H1,M1,M2,E1>>
  <PATH-ATTRIB Path ID=2 R-flag=0
    <OPPDIR-PATH-TLV OppositePath ID=4>>
  <ERO <H1,M3,M4,E1>>
  <PATH-ATTRIB Path ID=3 R-flag=1
    <OPPDIR-PATH-TLV OppositePath ID=1>>
  <ERO <E1,M2,M1,H1>>
  <PATH-ATTRIB Path ID=4 R-flag=1
    <OPPDIR-PATH-TLV OppositePath ID=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 Path ID=1 R-flag=0
    <OPPDIR-PATH-TLV OppositePath ID=3>>
  <ERO <H1,M5,M6,E1>>
  <PATH-ATTRIB Path ID=2 R-flag=0>
  <ERO <H1,M7,M8,E1>>
  <PATH-ATTRIB Path ID=3 R-flag=1
    <OPPDIR-PATH-TLV OppositePath ID=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 Path ID=1 R-flag=0
    <OPPDIR-PATH-TLV OppositePath ID=3>>
  <ERO <E1,M2,M1,H1>>
  <PATH-ATTRIB Path ID=2 R-flag=0
    <OPPDIR-PATH-TLV OppositePath ID=4>>
  <ERO <E1,M4,M3,H1>>
  <PATH-ATTRIB Path ID=3 R-flag=1
    <OPPDIR-PATH-TLV OppositePath ID=1>>
  <ERO <H1,M1,M2,E1>>
  <PATH-ATTRIB Path ID=4 R-flag=1
    <OPPDIR-PATH-TLV OppositePath ID=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 Path ID=1 R-flag=0
    <OPPDIR-PATH-TLV OppositePath ID=3>>
  <ERO <E1,M6,M5,H1>>
  <PATH-ATTRIB Path ID=2 R-flag=1>
  <ERO <H1,M7,M8,E1>>
  <PATH-ATTRIB Path ID=3 R-flag=1
    <OPPDIR-PATH-TLV OppositePath ID=1>>
  <ERO <H1,M5,M6,E1>>

```

Acknowledgements

Thanks to Adrian Farrel for shepherding this document, Ketan Talaulikar for his thorough AD review, Dhruv Dhody for ideas and discussion, and Diego Achaval, Quan Xiong, Giuseppe Fioccola, Italo Busi, Yuan Yaping, Cheng Li, Éric Vyncke, Mohamed Boucadair, Gunter Van de Velde, Christopher Inacio, Mike Bishop, Mahesh Jethanandani, Vidhi Goel, Gorrry Fairhurst, and David Mandelberg for their reviews.

Contributors

The following individuals initiated and developed the core work of this document. Mike Koldychev is also listed as editor in the Authors' Addresses section. The other individuals appear here rather than in the Authors' Addresses section due to the IETF guidelines on the maximum number of listed authors, but they should be considered co-authors of this document. Samuel Sidor joined the effort at a later stage as an additional editor.

Mike Koldychev Ed.

Ciena Corporation

Email: mkoldych@ciena.com**Siva Sivabalan**

Ciena Corporation

Email: ssivabal@ciena.com**Tarek Saad**

Cisco Systems

Email: tsaad@cisco.com**Vishnu Pavan Beeram**

Juniper Networks, Inc.

Email: vbeeram@juniper.net**Hooman Bidgoli**

Nokia

Email: hooman.bidgoli@nokia.com**Shuping Peng**

Huawei Technologies

Email: pengshuping@huawei.com**Bhupendra Yadav**

Ciena

Email: byadav@ciena.com

Gyan Mishra

Verizon Inc.

Email: hayabusagsm@gmail.com

The following individuals made contributions to this document:

Zafar Ali

Cisco Systems

Email: zali@cisco.com**Andrew Stone**

Nokia

Email: andrew.stone@nokia.com**Chen Ran**

ZTE

Email: chen.ran@zte.com.cn

Authors' Addresses

Mike Koldychev (EDITOR)

Ciena Corporation

Email: mkoldych@ciena.com**Samuel Sidor (EDITOR)**

Cisco Systems.

Email: ssidor@cisco.com