PCE Working Group
Internet-Draft
Updates: 8664 9603 (if approved)
Intended status: Standards Track
Expires: 2 April 2026

S. Sidor
Z. Rose
Cisco Systems, Inc.
S. Peng
ZTE Corporation
S. Peng
Huawei Technologies
A. Stone
Nokia

29 September 2025

Carrying SR-Algorithm in Path Computation Element Communication Protocol (PCEP)
draft-ietf-pce-sid-algo-25

Abstract

This document specifies extensions to the Path Computation Element Communication Protocol (PCEP) to enhance support for Segment Routing (SR) with a focus on the use of Segment Identifiers (SIDs) and SR-Algorithms in Traffic Engineering (TE). The SR-Algorithm associated with a SID defines the path computation algorithm used by Interior Gateway Protocols (IGPs). This document introduces extensions in three main areas.

Mechanisms for informing PCEP peers about the SR-Algorithm associated with SIDs by encoding this information in Explicit Route Object (ERO) and Record Route Object (RRO) subobjects. This document updates RFC 8664 and RFC 9603 to allow such extension.

The document specifies SR-Algorithm constraint, enabling refined path computations that can leverage IGP algorithm logic, including Flexible Algorithms, and their associated constraints and optimization metrics.

It defines new metric types for the METRIC object required to support SR-Algorithm based path computation, but also applicable to Label Switched Paths (LSPs) setup using different Path Setup Types.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

This Internet-Draft will expire on 2 April 2026.

Copyright Notice

Table of Contents

1. Introduction

[RFC5440] describes the Path Computation Element Communication Protocol (PCEP) for communication between a Path Computation Client (PCC) and a Path Computation Element (PCE) or between a pair of PCEs. [RFC8664] and [RFC9603] specify PCEP extensions to support Segment Routing (SR) over MPLS and IPv6 respectively.

This document specifies extensions to PCEP to enhance support for SR Traffic Engineering (TE). Specifically, it focuses on the use of Segment Identifiers (SIDs) and SR-Algorithms. An SR-Algorithm associated with a SID defines the path computation algorithm used by Interior Gateway Protocols (IGPs).

The PCEP extensions specified in this document are as follows:

Signaling SR-Algorithm in ERO and RRO: Mechanisms are introduced for PCEP peers to exchange information about the SR-Algorithm associated with each SID. This includes extending SR-ERO, SR-RRO and SRv6-ERO, SRv6-RRO subobjects to carry an Algorithm field. This document updates [RFC8664] and [RFC9603] to enable such encoding.

SR-Algorithm Constraint for Path Computation: Mechanisms are defined for signaling a specific SR-Algorithm as a constraint to the PCE for path computation. This includes a new SR-Algorithm TLV carried in the Label Switched Path Attributes (LSPA) Object.

Extensions to METRIC Object: Several new metric types are introduced for the METRIC Object to support optimization metrics derived from FADs during Flexible Algorithm path computation, their application is not restricted to Flexible Algorithms and they may be used with LSPs setup using different Path Setup Types.

1.1. Requirements Language

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

2. Terminology

This document uses the following terms defined in [RFC5440]: ERO, LSPA, PCC, PCE, PCEP, PCEP Peer, PCEP speaker, TED.

This document uses the following term defined in [RFC3031]: LSP.

This document uses the following term defined in [RFC9479] and [RFC9492]: ASLA.

This document uses the following terms defined in [RFC8664]: NAI and SR-DB.

Note that 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. 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, in the context of this document, refers to a Candidate Path within an SR Policy, which may be an SRv6 path (still represented using the LSP Object as specified in [RFC8231]).

The term extension block is used in this document to identify the additional bytes appended to a PCEP Object, which may exist depending on the inclusion of a flag in that object

The following terminologies are used in this document:

P2MP: Point-to-Multipoint

FAD: Flexible Algorithm Definition [RFC9350]

Winning FAD: The FAD selected according to the rules described in Section 5.3 of [RFC9350]

Subobject Extension Block: Optional, variable-length extension block for SR-ERO and SR-RRO subobjects defined in Section 4.2.1 of this document.

Subobject Extension Block Flag (SEBF): Any flag in Flags field of SR-ERO or SR-RRO subobjects that is used to signal that the corresponding field is encoded in the Subobject Extension Block.

3. Motivation

Existing PCEP specifications lack the mechanisms to explicitly signal and negotiate SR-Algorithm capabilities and constraints. This limits the ability of PCEs to make informed path computation decisions based on the specific SR-Algorithms supported and desired within the network. The absence of an explicit SR-Algorithm specification in PCEP messages implied no specific constraint on the SR-Algorithm to be used for path computation, effectively allowing the use of SIDs with any SR-algorithm.

A primary motivation for these extensions is to enable the PCE to leverage the path computation logic and topological information derived from Interior Gateway Protocols (IGPs), including Flexible Algorithms. Aligning PCE path computation with these IGP algorithms enables network operators to obtain paths that are congruent with the underlying routing behavior, which can result in segment lists with a reduced number of SIDs. The support for SR-Algorithm constraints in PCE path computation simplifies the deployment and management of Flexible Algorithm paths in multi-domain network scenarios.

The PCE and the headend router may independently compute SR-TE paths with different SR-Algorithms. This information needs to be exchanged between PCEP peers for purposes such as network monitoring and troubleshooting. In scenarios involving multiple (redundant) PCEs, when a headend receives a path from the primary PCE, it needs to be able to report the complete path information, including the SR-

Algorithm, to a backup PCE. This is essential for high availability (HA) scenarios, ensuring that the backup PCE can correctly verify Prefix SIDs.

The introduction of an SR-Algorithm TLV within the LSPA object allows operators to specify SR-Algorithm constraints directly, thereby refining path computations to meet specific needs, such as low-latency paths.

The ability to specify an SR-Algorithm per SID in ERO and RRO is crucial for multiple reasons, for example:

* SID types without algorithm specified - Certain SID types, such as Binding SIDs (BSIDs) [RFC8402], may not have an SR-algorithm specified. It may be inaccurate to state that an entire end-to-end path adheres to a specific algorithm if it includes a BSID from another policy. Note: In SRv6, the BSID can be allocated from an algo-specific SRv6 Locator which will result in the path to that BSID headend node following that algo-specific path. However, the implicit algorithm of BSID is independent from SR algorithm used for the SR Policy associated with that BSID.

* Topologies with two Interior Gateway Protocol (IGP) domains, each using the same FAD but with differing algorithm numbers.

4. Object Formats

4.1. OPEN Object

4.1.1. SR PCE Capability Sub-TLV

The SR-PCE-CAPABILITY Sub-TLV is defined in Section 4.1.2 of [RFC8664] to be included in the PATH-SETUP-TYPE-CAPABILITY TLV.

This document defines the following flag in the SR-PCE-CAPABILITY Sub-TLV:

* SR-Algorithm Capability (S): If the S-flag is set, a PCEP speaker indicates support for the Algorithm field and the Subobject Extension Block in the SR-ERO subobject described in Section 4.2 and the SR-Algorithm TLV described in Section 4.4 for LSPs setup using Path Setup Type 1 (Segment Routing) [RFC8664]. It does not indicate support for these extensions for other Path Setup Types.

4.1.2. SRv6 PCE Capability sub-TLV

The SRv6-PCE-CAPABILITY sub-TLV is defined in Section 4.1.1 of [RFC9603] to be included in the PATH-SETUP-TYPE-CAPABILITY TLV.

This document defines the following flag in the SRv6-PCE-CAPABILITY sub-TLV:

* SR-Algorithm Capability (S): If the S-flag is set, a PCEP speaker indicates support for the Algorithm field in the SRv6-ERO Subobject described in Section 4.3 and the SR-Algorithm TLV described in Section 4.4 for LSPs setup using Path Setup Type 3 (SRv6) [RFC9603]. It does not indicate support for these extensions for other Path Setup Types.

4.2. SR-ERO Subobject

This document updates the SR-ERO subobject format defined in Section 4.3.1 of [RFC8664] with a new optional, variable-length Subobject Extension Block field. The block is used to convey additional information, such as the Algorithm field, and is designed to allow future extensibility. Further, a new "A" flag in Flags field is introduced as shown in Figure 1.

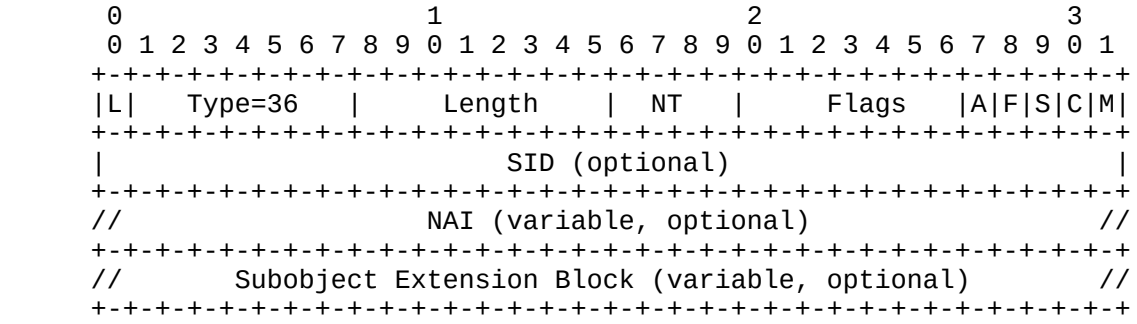

Figure 1: SR-ERO Subobject Format

A new bit in the Flags field:

* A-flag (SR-Algorithm Flag): If set to ’1’ by a PCEP speaker, the Subobject Extension Block MUST be included in the SR-ERO subobject as shown in Figure 1 along with the specified algorithm. The length of this block is variable and determined by subtracting the size of the fixed fields and any optional SID or NAI fields from the total subobject Length. The length of the Subobject Extension Block MUST always be a multiple of 4 bytes. If this flag is set to 0, then either:

  - the Subobject Extension Block is not included and processing described in Section 5.2.1 of [RFC8664] applies, or

- the Subobject Extension Block is included (due to an SEBF in a future document) and the Algorithm field MUST be ignored.

This document updates the SR-ERO subobject validation defined in Section 5.2.1 of [RFC8664] by extending existing validation to include the Subobject Extension Block and A bit as follows.

On receiving an SR-ERO subobject, a PCC MUST validate that the Length field, S bit, F bit, A bit, NT field, and any present SEBFs are consistent, as follows:

* If Subobject Extension Block is included (i.e., if any SEBF, such as A or a future flag, is set), the length of the subobject MUST include the size of the entire Subobject Extension Block as determined by the set of SEBFs.

  - The minimum size of the Subobject Extension Block is 4 bytes when only a single SEBF (such as A) is set, and may be longer (in multiples of 4 bytes) if additional SEBFs are set and require more space.

  - The total subobject Length is the sum of the sizes of the fixed and optional fields (SID, NAI, etc.) and the total size of the Subobject Extension Block required by the set of SEBFs.

  - The exact calculation of Length for each NT, S, F, and set of SEBFs is as follows:

    o If NT=0, the F bit MUST be 1, the S bit MUST be zero, and the Length MUST be 8 + the size of the Subobject Extension Block.

    o If NT=1, the F bit MUST be zero.

      + If the S bit is 1, the Length MUST be 8 + the size of the Subobject Extension Block.

      + If the S bit is 0, the Length MUST be 12 + the size of the Subobject Extension Block.

    o If NT=2, the F bit MUST be zero.

      + If the S bit is 1, the Length MUST be 20 + the size of the Subobject Extension Block.

      + If the S bit is 0, the Length MUST be 24 + the size of the Subobject Extension Block.

- If NT=3, the F bit MUST be zero.
  - If the S bit is 1, the Length MUST be 12 + the size of the Subobject Extension Block.
  - If the S bit is 0, the Length MUST be 16 + the size of the Subobject Extension Block.
- If NT=4, the F bit MUST be zero.
  - If the S bit is 1, the Length MUST be 36 + the size of the Subobject Extension Block.
  - If the S bit is 0, the Length MUST be 40 + the size of the Subobject Extension Block.
- If NT=5, the F bit MUST be zero.
  - If the S bit is 1, the Length MUST be 20 + the size of the Subobject Extension Block.
  - If the S bit is 0, the Length MUST be 24 + the size of the Subobject Extension Block.
- If NT=6, the F bit MUST be zero.
  - If the S bit is 1, the Length MUST be 44 + the size of the Subobject Extension Block.
  - If the S bit is 0, the Length MUST be 48 + the size of the Subobject Extension Block.

* If no SEBF (including A flag defined in this document) is set, the Length value MUST match the requirements as defined in Section 5.2.1 of [RFC8664] applies.

4.2.1. Subobject Extension Block

The Subobject Extension Block is an optional, extensible field in the SR-ERO subobject. Its presence is indicated by the setting of any SEBF in the subobject's Flags field (e.g., the A-flag defined in this document, or flags defined by future documents).

Block Length and Presence:

* If the A bit is 1, and no other SEBF is set, the block Length MUST be 4.

* The block length is at least 4 bytes when present.

* The block length MUST always be a multiple of 4 bytes

* The block MUST be included if any SEBF is set in the Flags field.

* Future documents may define additional SEBFs and corresponding fields, allowing the block to be increased in size beyond the initial 4 bytes as needed.

The first 4 bytes of the Subobject Extension Block are described in Figure 2.

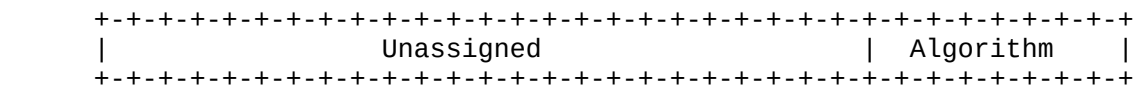

Figure 2: Subobject Extension Block Format

Unassigned (24 bits): This field is reserved for future use and MUST be set to zero when sending and ignored when receiving, unless redefined by a future extension that is indicated by an associated SEBF and capability.

Algorithm (8 bits): SR-Algorithm value from registry "IGP Algorithm Types" of "Interior Gateway Protocol (IGP) Parameters" IANA registry.

Future extensions SHOULD first re-use the Reserved portion of the initial 4 bytes to carry new information. If additional space is needed, the Subobject Extension Block MAY be extended in 4-byte increments. Each such extension MUST be indicated by a dedicated SEBF in the Flags field (similar to the A-flag) and MUST be accompanied by capability signaling in the appropriate capability sub-TLV.

When receiving a Subobject Extension Block longer than 4 bytes, receivers that do not recognize or have not negotiated support for additional flags MUST ignore the unknown additional bytes beyond those defined in this document.

4.2.2. Guidance for Future Extensions

Future documents extending the Subobject Extension Block MUST:

* Define a new SEBF in the Flags field to indicate their extension, and specify corresponding capability signaling.

* Specify which parts of the reserved/extension block are used and how the block length is calculated when their extension is present.

* The reserved bits in the initial 4 bytes are reused when possible, and the block is extended only when additional space is necessary.

* Future documents may define additional SEBFs and corresponding fields, allowing the block to be increased in size beyond the initial 4 bytes as needed.

Example: Future extension introducing a Z-flag and a new Z field (8 bits):

* If the A-flag and/or Z-flag are set, the Subobject Extension Block is included. The Z field may use 8 bits of the reserved portion. A field is only considered valid if its corresponding flag is set. For example, if the Z-flag is set but the A-flag is not, the Z-field is valid, but the Algorithm field is ignored.

* If space beyond the initial 4 bytes is needed, the extension document specifies the new block layout and total length. To simplify parsing, if a flag for such an extension is set, the full extended block is encoded, including the initial 4 bytes, even if the A-flag is not set.

4.3. SRv6-ERO Subobject

This document updates the SRv6-ERO subobject format defined in Section 4.3.1 of [RFC9603] with Algorithm field carved out of the Reserved field. Further, a new "A" flag in defined in the existing Flags field as shown in Figure 3.

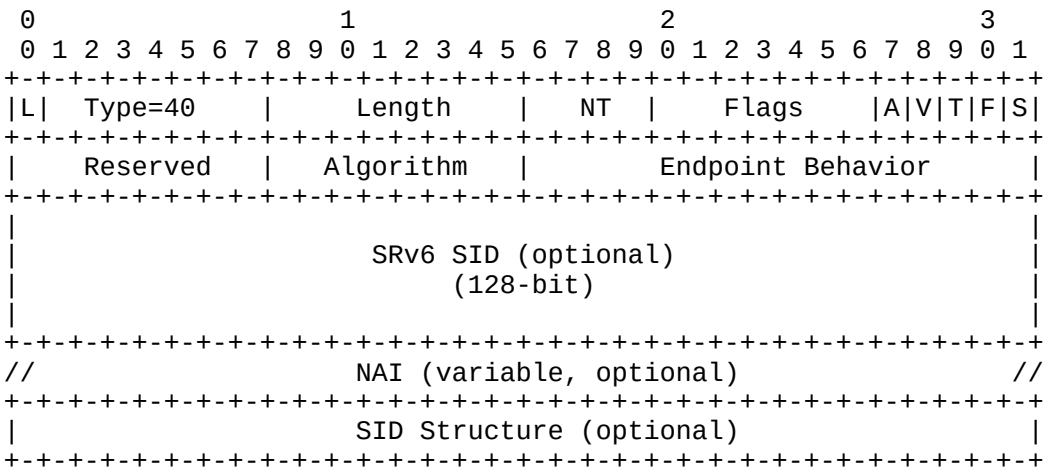

Figure 3: SRv6-ERO Subobject Format

A new bit in the Flags field:

A-flag (SR-Algorithm Flag): If set to ’1’ by a PCEP speaker, the Algorithm field is included in SRv6-ERO subobject as specified in Figure 3. If this flag is set to 0, then the Algorithm field is absent and processing described in Section 5.2.1 of [RFC9603] applies.

Reserved (8 bits): Reduced from 16 to 8 bits. It MUST be set to zero while sending and ignored on receipt.

Algorithm (8 bits): SR-Algorithm value from registry "IGP Algorithm Types" of "Interior Gateway Protocol (IGP) Parameters" IANA registry.

Note: Subobject Extension Block is applicable to SRv6-ERO Subobject, but is not required by this specific document as existing reserved space is re-used. When additional space is needed in the SRv6-ERO subobject, the future extensions SHOULD specify the usage of Subobject Extension Block for the SRv6-ERO Subobject.

4.4. SR-Algorithm TLV

A new TLV for the LSPA Object is introduced to carry the SR-Algorithm constraint (Section 5.2). This TLV SHOULD only be used when PST (Path Setup type) = 1 or 3 for SR-MPLS and SRv6, respectively. Only the first instance of this TLV MUST be processed, subsequent instances MUST be ignored.

The format of the SR-Algorithm TLV is as follows:

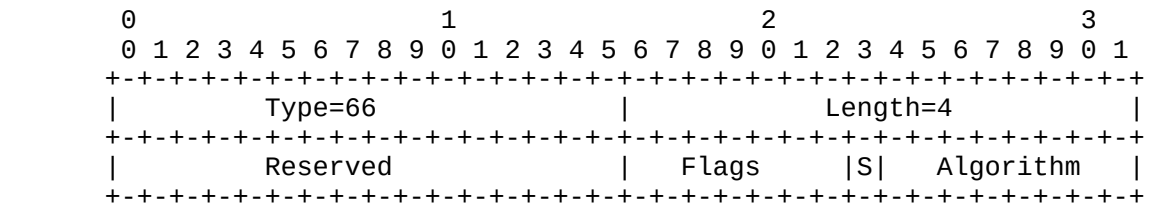

Figure 4: SR-Algorithm TLV Format

Type (16 bits): 66.

Length (16 bits): 4.

The 32-bit value is formatted as follows.

Reserved (16 bits): MUST be set to zero by the sender and MUST be ignored by the receiver.

Flags (8 bits): This document defines the following flag bits. The other bits MUST be set to zero by the sender and MUST be ignored by the receiver.

* S (Strict): If set, the path computation at the PCE MUST fail if the specified SR-Algorithm constraint cannot be satisfied. If unset, the PCE MUST try to compute the path with SR-algorithm constraint specified. If the path computation using the specified SR-Algorithm constraint fails, the PCE MUST try to compute a path that does not satisfy the constraint.

Algorithm (8 bits): SR-Algorithm to be used during path computation (see Section 5.2).

4.5. Extensions to METRIC Object

The METRIC object is defined in Section 7.8 of [RFC5440]. This document specifies new types for the METRIC object to enable the encoding of optimization metric types derived from the FAD during Flexible Algorithm path computation (see Section 5.2.2). While these new metric types are defined to support this specific use case, their use is not restricted to Flexible Algorithm path computation or to any specific Path Setup Type.

* T=22: Path Min Delay metric (Section 4.5.1.1)

* T=23: P2MP Path Min Delay metric (Section 4.5.1.2)

* T=24: Path Bandwidth Metric (Section 4.5.2.1)

* T=25: P2MP Path Bandwidth Metric (Section 4.5.2.2)

* T=128-255: User-defined metric (Section 4.5.3)

The following terminology is used and expanded along the way.

* A network comprises of a set of N links {Li, (i=1...N)}.

* A path P of a point-to-point (P2P) LSP is a list of K links {Lpi,(i=1...K)}.

* A P2MP tree T comprises a set of M destinations {Dest_j,(j=1...M)}.

4.5.1. Path Min Delay Metric

[RFC7471] and [RFC8570] define "Min/Max Unidirectional Link Delay Sub-TLV" to advertise the link minimum and maximum delay in microseconds in a 24-bit field.

[RFC5440] defines the METRIC object with a 32-bit metric value encoded in IEEE floating point format (see [IEEE.754.1985]).

The encoding for the Path Min Delay metric value is quantified in units of microseconds and encoded in IEEE floating point format.

The conversion from 24-bit integer to 32-bit IEEE floating point could introduce some loss of precision.

4.5.1.1. P2P Path Min Delay Metric

The minimum Link Delay metric is defined in [RFC7471] and [RFC8570] as "Min Unidirectional Link Delay". The Path Min Link Delay metric represents measured minimum link delay value over a configurable interval.

The Path Min Delay metric type of the METRIC object in PCEP represents the sum of the Min Link Delay metric of all links along a P2P path.

* A Min Link Delay metric of link L is denoted D(L).

* A Path Min Delay metric for the P2P path P = Sum {D(Lpi), (i=1...K)}.

4.5.1.2. P2MP Path Min Delay Metric

The P2MP Path Min Delay metric type of the METRIC object in PCEP encodes the Path Min Delay metric for the destination that observes the worst (i.e., highest value) delay metric among all destinations of the P2MP tree.

* The P2P Path Min Delay metric of the path to destination Dest_j is denoted by PMDM(Dest_j).

* The P2MP Path Min Delay metric for the P2MP tree T = Maximum{PMDM(Dest_j), (j=1...M)}.

4.5.2. Path Bandwidth Metric

Section 4 of [I-D.ietf-lsr-flex-algo-bw-con] defines a new metric type "Bandwidth Metric", which may be advertised in their link metric advertisements.

When performing Flexible Algorithm path computation as described in Section 5.2.2, procedures described in sections 4.1 and 5 from [I-D.ietf-lsr-flex-algo-bw-con] MUST be followed with automatic metric calculation.

For path computations in contexts other than Flexible Algorithm (including Path Setup Types other than 1 or 3 for SR-MPLS and SRv6), if the Generic Metric sub-TLV with Bandwidth metric type is not advertised for a link, the PCE implementation MAY apply a local policy to derive a metric value (similar to the procedures in Sections 4.1.3 and 4.1.4 of [I-D.ietf-lsr-flex-algo-bw-con]) or the link MAY be treated as if the metric value is unavailable (e.g. by using a default value). If the Bandwidth metric value is advertised for a link, the PCE MUST use the advertised value to compute the path metric in accordance with Section 4.5.2.1 and Section 4.5.2.2.

The Path Bandwidth metric value is encoded in IEEE floating point format.

The conversion from 24-bit integer to 32-bit IEEE floating point could introduce some loss of precision.

4.5.2.1. P2P Path Bandwidth Metric

The Path Bandwidth metric type of the METRIC object in PCEP represents the sum of the Bandwidth Metric of all links along a P2P path. Note: the link Bandwidth Metric utilized in the formula may be the original metric advertised on the link, which may have a value inversely proportional to the link capacity.

* A Bandwidth Metric of link L is denoted B(L).

* A Path Bandwidth metric for the P2P path P = Sum {B(Lpi), (i=1...K)}.

4.5.2.2. P2MP Path Bandwidth Metric

The Bandwidth metric type of the METRIC object in PCEP encodes the Path Bandwidth metric for the destination that observes the worst bandwidth metric among all destinations of the P2MP tree.

* The P2P Bandwidth metric of the path to destination Dest_j is denoted by BM(Dest_j).

* The P2MP Path Bandwidth metric for the P2MP tree T = Maximum{BM(Dest_j), (j=1...M)}.

4.5.3. User Defined Metric

Section 2 of [I-D.ietf-lsr-flex-algo-bw-con] defined a new metric type range for "User defined metric", which may be advertised in their link metric advertisements. These are user defined and can be assigned by an operator for local use.

User Defined metric values are encoded using the IEEE floating-point format.

The conversion from 24-bit integer to 32-bit IEEE floating point could introduce some loss of precision.

The metric type range was chosen to allow mapping with values assigned in the "IGP Metric-Type Registry". For example, the User Defined metric type 130 of the METRIC object in PCEP can represent the sum of the User Defined Metric 130 of all links along a P2P.

User Defined Metrics are equally applicable to P2P and P2MP paths.

5. Operation

The PCEP extensions defined in Section 5.1, Section 5.1.2 and Section 5.2 of this document MUST NOT be used unless both PCEP speakers have indicated support by setting the S flag in the Path Setup Type Sub-TLV corresponding to the PST of the LSP. If this condition is not met, the receiving PCEP speaker MUST respond with a PCErr message with Error-Type 19 (Invalid Operation) and Error-Value TBD3 (Attempted use of SR-Algorithm without advertised capability).

The SR-Algorithm used in this document refers to a complete range of SR-Algorithm values (0-255) if a specific section does not specify otherwise. Valid SR-Algorithm values are defined in the registry "IGP Algorithm Types" of "Interior Gateway Protocol (IGP) Parameters" IANA registry. Refer to Section 3.1.1 of [RFC8402] and [RFC9256] for the definition of SR-Algorithm in Segment Routing. [RFC8665] and [RFC8667] are describing the use of the SR-Algorithm in IGP. Note that some RFCs are referring to SR-Algorithm with different names, for example "Prefix-SID Algorithm" and "SR Algorithm".

5.1. ERO and RRO Subobjects

If a PCC receives the Algorithm field in the ERO subobject within PCInitiate, PCUpd, or PCRep messages and the path received from those messages is being included in the ERO of PCRpt message, then the PCC MUST include the Algorithm field in the encoded subobjects with the received SR-Algorithm value.

As per [RFC8664], the format of the SR-RRO subobject is the same as that of the SR-ERO subobject, but without the L-Flag, therefore SR-RRO subobject may also carry the A flag and Algorithm field in the Subobject Extension Block. Similarly, as per [RFC9603], the format of the SRv6-RRO subobject is the same as that of the SRv6-ERO subobject but without the L flag, therefore SRv6-RRO subobject may also carry the A flag and Algorithm field.

5.1.1. SR-ERO

A PCEP speaker MAY set the A flag and include the Algorithm field as part of Subobject Extension Block in an SR-ERO subobject if the S flag has been advertised in SR-PCE-CAPABILITY Sub-TLV by both PCEP speakers.

If the PCEP peer receives an SR-ERO subobject with the A flag set, but the S flag was not advertised in SR-PCE-CAPABILITY Sub-TLV, then it MUST consider the entire ERO as invalid as described in Section 5.2.1 of [RFC8664].

The Subobject Extension Block field in the SR-ERO subobject MUST be included after the optional SID, NAI, or SID structure and the length of the SR-ERO subobject MUST be increased by the size of the Subobject Extension Block, as determined by the set of SEBFs.

If the length and the A flag are not consistent as specified in Section 4.2, PCEP peer MUST consider the entire ERO invalid and MUST send a PCErr message with Error-Type = 10 ("Reception of an invalid object") and Error-value = 11 ("Malformed object").

If the SID value is absent (S bit is set to 1), the NAI value is present (F bit is set to 0) and the Algorithm field is set (A bit is set to 1), the PCC is responsible for choosing the SRv6-SID value based on values specified in NAI and Algorithm fields. If the PCC cannot find a SID index in the SR-DB, it MUST send a PCErr message with Error-Type = 10 ("Reception of an invalid object") and Error-value = 14 ("Unknown SID").

5.1.2. SRv6-ERO

A PCEP speaker MAY set the A flag and include the Algorithm field in an SRv6-ERO subobject if the S flag has been advertised in SRv6-PCE-CAPABILITY sub-TLV by both PCEP speakers.

If the PCEP peer receives SRv6-ERO subobject with the A flag set or with the SR-Algorithm included, but the S flag was not advertised in SRv6-PCE-CAPABILITY Sub-TLV, then it MUST consider the entire ERO as invalid as described in Section 5.2.1 of [RFC8664].

The Algorithm field in the SRv6-ERO subobject MUST be included in the position specified in Section 4.3, the length of the SRv6-ERO subobject is not impacted by the inclusion of the Algorithm field.

If the SRv6-SID value is absent (S bit is set to 1), the NAI value is present (F bit is set to 0) and the Algorithm field is set (A bit is set to 1), the PCC is responsible for choosing the SRv6-SID value based on values specified in NAI and Algorithm fields. If the PCC cannot find a SID index in the SR-DB, it MUST send a PCErr message with Error-Type = 10 ("Reception of an invalid object") and Error-value = 14 ("Unknown SID").

5.2. SR-Algorithm Constraint

To signal a specific SR-Algorithm constraint to the PCE, the headend MUST encode the SR-Algorithm TLV inside the LSPA object.

If a PCC receives an LSPA object with SR-Algorithm TLV as part of PCInitiate, PCUpd messages, then it MUST include LSPA object with SR-Algorithm TLV in PCRpt message as part of intended-attribute-list.

If a PCE receives an LSPA object with SR-Algorithm TLV in PCRpt or PCReq, then it MUST include the LSPA object with SR-Algorithm TLV in PCUpd message, or PCRep message in case of an unsuccessful path computation based on rules described in Section 7.11 of [RFC5440].

A PCEP peer that did not advertise the S flag in the Path Setup Type Sub-TLV corresponding to the LSP’s PST, it MUST ignore the SR-Algorithm TLV on receipt.

The PCE MUST NOT use Prefix SIDs associated with an SR-Algorithm other than the one specified in the SR-Algorithm constraint. If a protected Adjacency SID is used without an associated SR-Algorithm, there is a risk that the backup path may fail to forward traffic over parts of the topology that are not included in the specified SR-Algorithm. Consequently, it is NOT RECOMMENDED to use protected Adjacency SIDs without an explicitly specified SR-Algorithm. If an Adjacency SID has an associated SR-Algorithm, the PCE MUST ensure that the SR-Algorithm matches the one specified in the SR-Algorithm constraint.

Other SID types, such as Binding SIDs, are allowed. Furthermore, the inclusion of a path Binding SID (BSID) from another policy is permitted only if the path associated with that policy fully satisfies all the constraints of the current path computation.

The specified SR-Algorithm constraint is applied to the end-to-end SR policy path. Using different SR-Algorithm constraint or using winning FAD with different optimization metric or constraints for same SR-Algorithm in each domain or part of the topology in single path computation is out of the scope of this document.

If the PCE is unable to find a path with the given SR-Algorithm constraint, it does not support a combination of specified constraints or if the FAD contains constraints, optimization metric or other attributes, which the PCE does not support or recognize, it MUST use empty ERO in PCInitiate for LSP instantiation or PCUpd message if an update is required or NO-PATH object in PCRep to indicate that it was not able to find the valid path.

If the Algorithm field value is in the range 128-255, the PCE MUST perform path computation according to the Flexible Algorithm procedures outlined in Section 5.2.2. Otherwise, the PCE MUST adhere to the path computation procedures with SID filtering defined in Section 5.2.1.

If the NO-PATH object is included in PCRep, then the PCE MAY include SR-Algorithm TLV to indicate constraint, which cannot be satisfied as described in Section 7.5 of [RFC5440].

SR-Algorithm does not replace the Objective Function defined in [RFC5541].

5.2.1. Path Computation for SR-Algorithms 0-127

The SR-Algorithm constraint acts as a filter, restricting which SIDs may be used as a result of the path computation function. Path computation is done based on optimization metric type and constraints specified in the PCEP message received from the PCC.

The mechanism described in this section is applicable only to SR-Algorithm values in the range 0-127. It is not applicable to Flexible Algorithms (range 128-255), which are handled as described in Section 5.2.2. Within the 0-127 range, currently defined algorithms are 0 (Shortest Path First (SPF)) and 1 (Strict SPF) as introduced in Section 3.1.1 of [RFC8402]. Future algorithms defined within this range that do not require explicit PCEP extensions beyond the SR-Algorithm TLV may also utilize this SID filtering approach. If a PCE implementation receives a request with an SR-Algorithm value in the 0-127 range that it does not support for path computation, it MUST reject the PCEP message and send a PCErr message with Error-Type 19 (Invalid Operation) and Error-Value TBD4 (Unsupported SR-Algorithm).

5.2.2. Path Computation for Flexible Algorithms

This section is applicable only to the Flexible Algorithms range of SR-Algorithm values. The PCE performs Flexible Algorithm path computation based on topology information stored in its TED [RFC5440]. The TED is expected to be populated with necessary information, including Flexible Algorithm Definitions (FADs), node participation, and ASLA-specific link attributes, through standard mechanisms such as Interior Gateway Protocols (IGPs) with Traffic Engineering extensions or BGP-LS [RFC9552].

The PCE must follow the IGP Flexible Algorithm path computation logic as described in [RFC9350]. This includes performing the FAD selection as described in Section 5.3 of [RFC9350] and other sections, determining the topology associated with specific Flexible Algorithm based on the FAD, the node participation Section 11 of [RFC9350], using ASLA-specific link attributes Section 12 of [RFC9350], and applying other rules for Flexible Algorithm path calculation Section 13 of [RFC9350]. While [RFC9350] defines the base procedures for IGP Flexible Algorithms, these procedures are further extended by other documents such as [I-D.ietf-lsr-flex-algo-bw-con], a PCE implementation may need support these IGP extensions to allow use of specific constraints in FAD. [I-D.ietf-lsr-igp-flex-algo-reverse-affinity] introduced IANA registry called "IGP Flex-Algorithm Path Computation Rules Registry" within the "Interior Gateway Protocol (IGP) Parameters" registry group with the ordered set of rules that MUST be used to prune links from the topology during the Flex-Algorithm path computation.

[Note to RFC Editor: The URL of the "IGP Flex-Algorithm Path Computation Rules Registry" IANA registry to be inserted once it will get created after approval of [I-D.ietf-lsr-igp-flex-algo-reverse-affinity].]

The PCE must optimize the computed path based on the metric type specified in the FAD. The optimization metric type included in PCEP messages from the PCC MUST be ignored. The PCE MUST use the metric type from the FAD in messages sent to the PCC unless that metric type is not defined in PCEP or not supported by the PCEP peer. It is allowed to use SID types other than Prefix SID (e.g., Adjacency or BSID), but only from nodes participating in the specified SR-Algorithm.

There are corresponding metric types in PCEP for IGP and TE metric from FAD introduced in [RFC9350], but there were no corresponding metric types defined for "Min Unidirectional Link Delay" from [RFC9350] and "Bandwidth Metric", "User Defined Metric" from [I-D.ietf-lsr-flex-algo-bw-con]. Section 4.5 of this document is introducing them. Note that the defined "Path Bandwidth Metric" is accumulative and is different from the Bandwidth Object defined in [RFC5440].

The PCE MUST use the constraints specified in the FAD and also constraints (except optimization metric type) directly included in PCEP messages from the PCC. The PCE implementation MAY decide to ignore specific constraints received from the PCC based on existing processing rules for PCEP Objects and TLVs, e.g. P flag described in Section 7.2 of [RFC5440] and processing rules described in [RFC9753]. If the PCE does not support a specified combination of constraints,

it MUST fail path computation and respond with a PCEP message with PCInitiate or PCUpd message with empty ERO or PCRep with NO-PATH object. PCC MUST NOT include constraints from FAD in PCEP message sent to PCE as it can result in undesired behavior in various cases. PCE SHOULD NOT include constraints from FAD in PCEP messages sent to PCC.

The combinations of the constraints specified in the FAD and constraints directly included in PCEP messages from the PCC may decrease the chance that Flex-algo specific Prefix SIDs represent an optimal path while satisfying all specified constraints, as a result a longer SID list may be required for the computed path. Adding more constraints on top of FAD requires complex path computation and may reduce the benefit of this scheme.

5.3. New Metric types

All the rules of processing the METRIC object as explained in [RFC5440] and [RFC8233] are applicable to new metric types defined in this document.

6. Manageability Considerations

All manageability requirements and considerations listed in [RFC5440], [RFC8231], [RFC8281], [RFC8664] and [RFC9603] apply to PCEP extensions defined in this document. In addition, the requirements and considerations listed in this section apply.

6.1. Control of Function and Policy

A PCE or PCC implementation MAY allow the capability of supporting PCEP extensions introduced in this document to be enabled or disabled as part of the global configuration. By default, this capability SHOULD be enabled.

6.2. Information and Data Models

An implementation SHOULD allow the operator to view the capability defined in this document. Sections 4.1 and 4.1.1 of [I-D.ietf-pce-pcep-yang] should be extended to include the capabilities introduced in Sections 3.1.1 and 3.1.2 for PCEP peer.

6.3. Liveness Detection and Monitoring

This document does not define any new mechanism that impacts the liveness detection and monitoring of PCEP.

6.4. Verify Correct Operations

An implementation SHOULD also allow the operator to view FADs, which may be used in Flexible Algorithm path computation defined in Section 5.2.2.

An implementation SHOULD allow the operator to view nodes participating in the specified SR-Algorithm.

6.5. Requirements on Other Protocols and Functional Components

This document does not put new requirements but relies on the necessary IGP extensions.

6.6. Impact On Network Operations

This document inherits considerations from documents describing IGP Flexible Algorithm - for example [RFC9350] and [I-D.ietf-lsr-flex-algo-bw-con].

7. Operational Considerations

This document inherits operational considerations from documents describing IGP Flexible Algorithm - for example [RFC9350] and [I-D.ietf-lsr-flex-algo-bw-con].

8. Implementation Status

[Note to the RFC Editor - remove this section before publication, as well as remove the reference to RFC 7942.]

This section records the status of known implementations of the protocol defined by this specification at the time of posting of this Internet-Draft, and is based on a proposal described in [RFC7942]. The description of implementations in this section is intended to assist the IETF in its decision processes in progressing drafts to RFCs. Please note that the listing of any individual implementation here does not imply endorsement by the IETF. Furthermore, no effort has been spent to verify the information presented here that was supplied by IETF contributors. This is not intended as, and must not be construed to be, a catalog of available implementations or their features. Readers are advised to note that other implementations may exist.

According to [RFC7942], "this will allow reviewers and working groups to assign due consideration to documents that have the benefit of running code, which may serve as evidence of valuable experimentation and feedback that have made the implemented protocols more mature. It is up to the individual working groups to use this information as they see fit".

8.1. Cisco

* Organization: Cisco Systems
* Implementation: IOS-XR PCC and PCE.
* Description: SR-MPLS part with experimental codepoints.
* Maturity Level: Production.
* Coverage: Partial.
* Contact: ssidor@cisco.com

8.2. Huawei

* Organization: Huawei
* Implementation: NE Series Routers
* Description: SR Policy with SR Algorithm.
* Maturity Level: Production.
* Coverage: Partial.
* Contact: pengshuping@huawei.com

9. Security Considerations

The security considerations described in [RFC5440], [RFC8231], [RFC8253], [RFC8281], [RFC8664], [RFC9603] and [RFC9350] apply to the extensions described in this document as well.

Note that this specification introduces the possibility of computing paths by the PCE based on Flexible Algorithm related topology attributes and based on the metric type and constraints from FAD. This creates additional vulnerabilities, which are already described for the path computation done by IGP like those described in Security Considerations section of [RFC9350], but which are also applicable to path computation done by PCE. Hence, securing the PCEP session using Transport Layer Security (TLS) [RFC8253] is RECOMMENDED as per the recommendations and best current practices described in [RFC9325].

10. IANA Considerations

10.1. SR Capability Flag

IANA maintains a registry, named "SR Capability Flag Field", within the "Path Computation Element Protocol (PCEP) Numbers" registry group to manage the Flags field of the SR-PCE-CAPABILITY TLV. IANA is requested to confirm the following early allocation:

Bit	Description	Reference
5	SR-Algorithm Capability	This document

Table 1

10.2. SRv6 PCE Capability Flag

IANA maintains a registry, named "SRv6 Capability Flag Field", within the "Path Computation Element Protocol (PCEP) Numbers" registry group to manage the Flags field of SRv6-PCE-CAPABILITY sub-TLV. IANA is requested to make the following assignment:

Bit	Description	Reference
TBD1	SR-Algorithm Capability	This document

Table 2

10.3. SR-ERO Flag

IANA maintains a registry, named "SR-ERO Flag Field", within the "Path Computation Element Protocol (PCEP) Numbers" registry group to manage the Flags field of the SR-ERO Subobject. IANA is requested to confirm the following early allocation:

Bit	Description	Reference
7	SR-Algorithm Flag (A)	This document

Table 3

10.4. SRv6-ERO Flag

IANA maintains a registry, named "SRv6-ERO Flag Field", within the "Path Computation Element Protocol (PCEP) Numbers" registry group to manage the Flags field of the SRv6-ERO subobject. IANA is requested to make the following assignment:

Bit	Description	Reference
TBD2	SR-Algorithm Flag (A)	This document

Table 4

10.5. PCEP TLV Types

IANA maintains a registry, named "PCEP TLV Type Indicators", within the "Path Computation Element Protocol (PCEP) Numbers" registry group. IANA is requested to confirm the early allocation of a new TLV type for the new LSPA TLV specified in this document.

Type	Description	Reference
66	SR-Algorithm	This document

Table 5

10.6. Metric Types

IANA maintains a registry for "METRIC Object T Field" within the "Path Computation Element Protocol (PCEP) Numbers" registry group. IANA is requested to confirm the early allocated codepoints as follows:

Type	Description	Reference
22	Path Min Delay Metric	This document
23	P2MP Path Min Delay Metric	This document
24	Path Bandwidth Metric	This document
25	P2MP Path Bandwidth Metric	This document
128-255	User Defined Metric	This document

Table 6

10.7. PCEP-Error Object

IANA is requested to allocate new error types and error values within the "PCEP-ERROR Object Error Types and Values" sub-registry of the PCEP Numbers registry for the following errors.

Error-Type	Meaning	Error-Value
19	Invalid Operation	TBD3:Attempted use of SR-Algorithm without advertised capability
		TBD4:Unsupported combination of constraints

Table 7

11. References

11.1. Normative References

[I-D.ietf-lsr-flex-algo-bw-con]
Hegde, S., Britto, W., Shetty, R., Decraene, B., Psenak, P., and T. Li, "IGP Flexible Algorithms: Bandwidth, Delay, Metrics and Constraints", Work in Progress, Internet-Draft, draft-ietf-lsr-flex-algo-bw-con-22, 13 February 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-lsr-flex-algo-bw-con-22>.

[I-D.ietf-lsr-igp-flex-algo-reverse-affinity]
Psenak, P., Horn, J., and Dhamija, "IGP Flexible Algorithms Reverse Affinity Constraint", Work in Progress, Internet-Draft, draft-ietf-lsr-igp-flex-algo-reverse-affinity-12, 5 August 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-lsr-igp-flex-algo-reverse-affinity-12>.

[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>.

[RFC7471] Giacalone, S., Ward, D., Drake, J., Atlas, A., and S. Previdi, "OSPF Traffic Engineering (TE) Metric Extensions", RFC 7471, DOI 10.17487/RFC7471, March 2015, <https://www.rfc-editor.org/info/rfc7471>.

[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>.

[RFC8233] Dhody, D., Wu, Q., Manral, V., Ali, Z., and K. Kumaki, "Extensions to the Path Computation Element Communication Protocol (PCEP) to Compute Service-Aware Label Switched Paths (LSPs)", RFC 8233, DOI 10.17487/RFC8233, September 2017, <https://www.rfc-editor.org/info/rfc8233>.

[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>.

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

[RFC8570] Ginsberg, L., Ed., Previdi, S., Ed., Giacalone, S., Ward, D., Drake, J., and Q. Wu, "IS-IS Traffic Engineering (TE) Metric Extensions", RFC 8570, DOI 10.17487/RFC8570, March 2019, <https://www.rfc-editor.org/info/rfc8570>.

[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>.

[RFC8665] Psenak, P., Ed., Previdi, S., Ed., Filsfils, C., Gredler, H., Shakir, R., Henderickx, W., and J. Tantsura, "OSPF Extensions for Segment Routing", RFC 8665, DOI 10.17487/RFC8665, December 2019, <https://www.rfc-editor.org/info/rfc8665>.

[RFC8667] Previdi, S., Ed., Ginsberg, L., Ed., Filsfils, C., Bashandy, A., Gredler, H., and B. Decraene, "IS-IS Extensions for Segment Routing", RFC 8667, DOI 10.17487/RFC8667, December 2019, <https://www.rfc-editor.org/info/rfc8667>.

[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>.

[RFC9350] Psenak, P., Ed., Hegde, S., Filsfils, C., Talaulikar, K., and A. Gulko, "IGP Flexible Algorithm", RFC 9350, DOI 10.17487/RFC9350, February 2023, <https://www.rfc-editor.org/info/rfc9350>.

[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>.

[RFC9753] Li, C., Zheng, H., and S. Litkowski, "Extension for Stateful PCE to Allow Optional Processing of Path Computation Element Communication Protocol (PCEP) Objects", RFC 9753, DOI 10.17487/RFC9753, April 2025, <https://www.rfc-editor.org/info/rfc9753>.

11.2. Informative References

[I-D.ietf-pce-pcep-yang]
Dhody, D., Beeram, V. P., Hardwick, J., and J. Tantsura, "A YANG Data Model for Path Computation Element Communications Protocol (PCEP)", Work in Progress, Internet-Draft, draft-ietf-pce-pcep-yang-30, 26 January 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-pce-pcep-yang-30>.

[IEEE.754.1985]
IEEE, "Standard for Binary Floating-Point Arithmetic", DOI 10.1109/IEEESTD.1985.82928, IEEE Standard 754, October 1985, <https://doi.org/10.1109/IEEESTD.1985.82928>.

[RFC3031] Rosen, E., Viswanathan, A., and R. Callon, "Multiprotocol Label Switching Architecture", RFC 3031, DOI 10.17487/RFC3031, January 2001, <https://www.rfc-editor.org/info/rfc3031>.

[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>.

[RFC5541] Le Roux, JL., Vasseur, JP., and Y. Lee, "Encoding of Objective Functions in the Path Computation Element Communication Protocol (PCEP)", RFC 5541, DOI 10.17487/RFC5541, June 2009, <https://www.rfc-editor.org/info/rfc5541>.

[RFC7942] Sheffer, Y. and A. Farrel, "Improving Awareness of Running Code: The Implementation Status Section", BCP 205, RFC 7942, DOI 10.17487/RFC7942, July 2016, <https://www.rfc-editor.org/info/rfc7942>.

[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>.

[RFC9479] Ginsberg, L., Psenak, P., Previdi, S., Henderickx, W., and J. Drake, "IS-IS Application-Specific Link Attributes", RFC 9479, DOI 10.17487/RFC9479, October 2023, <https://www.rfc-editor.org/info/rfc9479>.

[RFC9492] Psenak, P., Ed., Ginsberg, L., Henderickx, W., Tantsura, J., and J. Drake, "OSPF Application-Specific Link Attributes", RFC 9492, DOI 10.17487/RFC9492, October 2023, <https://www.rfc-editor.org/info/rfc9492>.

[RFC9552] Talaulikar, K., Ed., "Distribution of Link-State and Traffic Engineering Information Using BGP", RFC 9552, DOI 10.17487/RFC9552, December 2023, <https://www.rfc-editor.org/info/rfc9552>.

Appendix A. Acknowledgement

Thanks to Dhruv Dhody for shepherding the document and for his contributions and suggestions.

Would like to thank Adrian Farrel, Aijun Wang, Boris Khasanov, Jie Dong, Ketan Talaulikar, Marina Fizgeer, Nagendra Nainar, Rakesh Gandhi, Russ White, Shraddha Hegde for review and suggestions.

Appendix B. Contributors

Mike Koldychev
Ciena Corporation
Email: mkoldych@proton.me

Zafar Ali
Cisco Systems, Inc.
Email: zali@cisco.com

Stephane Litkowski
Cisco Systems, Inc.
Email: slitkows.ietf@gmail.com

Siva Sivabalan
Ciena
Email: msiva282@gmail.com

Tarek Saad
Cisco Systems, Inc.
Email: tsaad.net@gmail.com

Mahendra Singh Negi
RtBrick Inc
Email: mahend.ietf@gmail.com

Tom Petch
Email: ietfc@btconnect.com

Authors' Addresses

Samuel Sidor
Cisco Systems, Inc.
Eurovea Central 3.
Pribinova 10
811 09 Bratislava
Slovakia
Email: ssidor@cisco.com

Zoey Rose
Cisco Systems, Inc.
2300 East President George
Richardson, TX 75082
United States of America
Email: atokar@cisco.com

Shaofu Peng
ZTE Corporation
No.50 Software Avenue
Nanjing
Jiangsu, 210012
China
Email: peng.shaofu@zte.com.cn

Shuping Peng
Huawei Technologies
Huawei Campus, No. 156 Beiqing Rd.
Beijing
100095
China
Email: pengshuping@huawei.com

Andrew Stone
Nokia
Email: andrew.stone@nokia.com