Updates to SID Verification for SR-MPLS in RFC 8664
draft-chen-pce-sr-mpls-sid-verification-04

Abstract

This document updates [RFC8664] to clarify usage of "SID verification" bit signalled in Path Computation Element Protocol (PCEP), and this document proposes to define a new flag for indicating the headend is explicitly requested to verify SID(s) by the PCE.

Status of This Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 9 January 2023.

Copyright Notice

Copyright (c) 2022 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.
1. Introduction

[I-D.ietf-spring-segment-routing-policy] describes the "SID verification" bit usage. SID verification is performed when the headend is explicitly requested to verify SID(s) by the controller via the signaling protocol used. Implementations MAY provide a local configuration option to enable verification on a global or per policy or per candidate path basis.

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

This document updates [RFC8664] to clarify usage of "SID verification" bit signalled in Path Computation Element Protocol (PCEP), and this document proposes to define a new flag for indicating the headend is explicitly requested to verify SID(s) by the PCE.

2. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].
cloud transport network: It is usually a national or province backbone network to achieve interconnection between multiple regional clouds/core clouds deployed in the country/province.

3. SID verification flag (V-Flag)

3.1. Extended V-Flag in SR-ERO Subobject

Section 4.3.1 in Path Computation Element Communication Protocol (PCEP) Extensions for Segment Routing [RFC8664] describes a new ERO subobject referred to as the "SR-ERO subobject" to carry a SID and/or NAI information. A new flag is proposed in this document in the SR-ERO Subobject for indicating the pcc is explicitly requested to verify SID(s) by the PCE.

The format of the SR-ERO subobject as defined in [RFC8664] is:

```
  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|L|   Type=TBD |     Length    |  NT   |     Flags   |V|F|S|C|M|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         SID (optional)                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
//                   NAI (variable, optional)                  //
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 1

V: When the V-Flag is set then PCC MUST consider the "SID verification" as described in Section 5.1 in [I-D.ietf-spring-segment-routing-policy].

The other fields in the SR-ERO subobject is the same as that of the SR-ERO subobject as defined in [RFC8664].

3.2. Extended V-Flag in SR-RRO Subobject

The format of the SR-RRO subobject is the same as that of the SR-ERO subobject, but without the L-Flag, per [RFC8664].

The V flag has no meaning in the SR-RRO and is ignored on receipt at the PCE.

4. Acknowledgements

TBD.
5. IANA Considerations

5.1. SR-ERO Subobject

This document defines a new bit value in the sub-registry "SR-ERO Flag Field" in the "Path Computation Element Protocol (PCEP) Numbers" registry.

<table>
<thead>
<tr>
<th>Bit</th>
<th>Name</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>TBA</td>
<td>SID verification(V)</td>
<td>This document</td>
</tr>
</tbody>
</table>

Figure 2

6. Security Considerations

TBD.

7. Normative References

[I-D.ietf-spring-segment-routing-policy]


Authors’ Addresses

Ran Chen
ZTE Corporation
Nanjing
China
Email: chen.ran@zte.com.cn
This document defines PCEP extensions to distribute In-situ Flow Information Telemetry (IFIT) information. So that IFIT behavior can be enabled automatically when the path is instantiated. In-situ Flow Information Telemetry (IFIT) refers to network OAM data plane on-path telemetry techniques, in particular the most popular are In-situ OAM (IOAM) and Alternate Marking. The IFIT attributes here described can be generalized for all path types but the application to Segment Routing (SR) is considered in this document. This document extends PCEP to carry the IFIT attributes under the stateful PCE model.

Requirements Language

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

Status of This Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."
time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 25, 2023.

Copyright Notice

Copyright (c) 2022 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 Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction ...................................................... 3
2. PCEP Extensions for IFIT Attributes .......................... 4
   2.1. IFIT for SR Policies ..................................... 5
3. IFIT capability advertisement TLV ............................ 5
4. IFIT Attributes TLV ........................................... 7
   4.1. IOAM Sub-TLVs ............................................ 8
       4.1.1. IOAM Pre-allocated Trace Option Sub-TLV ........... 9
       4.1.2. IOAM Incremental Trace Option Sub-TLV ........... 10
       4.1.3. IOAM Directly Export Option Sub-TLV ............ 10
       4.1.4. IOAM Edge-to-Edge Option Sub-TLV ............... 11
4.2. Enhanced Alternate Marking Sub-TLV .......................... 12
5. PCEP Messages .................................................... 13
   5.1. The PCInitiate Message ................................ 13
   5.2. The PCUpd Message ....................................... 14
   5.3. The PCRpt Message ...................................... 14
6. Example of application to SR Policy ............................ 14
7. IANA Considerations ............................................. 15
   7.1. PCEP TLV Type Indicators ................................ 15
   7.2. IFIT-CAPABILITY TLV Flags field ....................... 16
   7.3. IFIT-ATTRIBUTES Sub-TLV ................................ 16
   7.4. Enhanced Alternate Marking Sub-TLV Flags field .......... 17
   7.5. PCEP Error Codes ........................................ 18
8. Security Considerations ......................................... 18
9. Contributors ...................................................... 19
10. Acknowledgements ............................................... 19
11. References ....................................................... 19
1. Introduction

In-situ Flow Information Telemetry (IFIT) refers to network OAM (Operations, Administration, and Maintenance) data plane on-path telemetry techniques, including In-situ OAM (IOAM) [I-D.ietf-ippm-ioam-data] and Alternate Marking [RFC8321]. It can provide flow information on the entire forwarding path on a per-packet basis in real time.

An automatic network requires the Service Level Agreement (SLA) monitoring on the deployed service. So that the system can quickly detect the SLA violation or the performance degradation, hence to change the service deployment.

This document defines extensions to PCEP to distribute paths carrying IFIT information. So that IFIT behavior can be enabled automatically when the path is instantiated.

RFC 5440 [RFC5440] describes the Path Computation Element Protocol (PCEP) as a communication mechanism between a Path Computation Client (PCC) and a Path Computation Element (PCE), or between a PCE and a PCE.

RFC 8231 [RFC8231] specifies extensions to PCEP to enable stateful control and it describes two modes of operation: passive stateful PCE and active stateful PCE. Further, RFC 8281 [RFC8281] describes the setup, maintenance, and teardown of PCE-initiated LSPs for the stateful PCE model.

When a PCE is used to initiate paths using PCEP, it is important that the head end of the path also understands the IFIT behavior that is intended for the path. When PCEP is in use for path initiation it makes sense for that same protocol to be used to also carry the IFIT attributes that describe the IOAM or Alternate Marking procedure that needs to be applied to the data that flow those paths.

The PCEP extension defined in this document allows to signal the IFIT capabilities. In this way IFIT methods are automatically activated and running. The flexibility and dynamicity of the IFIT applications are given by the use of additional functions on the controller and on the network nodes, but this is out of scope here.

IFIT is a solution focusing on network domains according to [RFC8799] that introduces the concept of specific domain solutions. A network
domain consists of a set of network devices or entities within a single administration. As mentioned in [RFC8799], for a number of reasons, such as policies, options supported, style of network management and security requirements, it is suggested to limit applications including the emerging IFIT techniques to a controlled domain. Hence, the IFIT methods MUST be typically deployed in such controlled domains.

The Use Case of Segment Routing (SR) is also discussed considering that IFIT methods are becoming mature for Segment Routing over the MPLS data plane (SR-MPLS) and Segment Routing over IPv6 data plane (SRv6). SR policy [I-D.ietf-spring-segment-routing-policy] is a set of candidate SR paths consisting of one or more segment lists and necessary path attributes. It enables instantiation of an ordered list of segments with a specific intent for traffic steering. The PCEP extension defined in this document also enables SR policy with native IFIT, that can facilitate the closed loop control and enable the automation of SR service.

It is to be noted the companion document [I-D.qin-idr-sr-policy-ifit] that proposes the BGP extension to enable IFIT methods for SR policy.

2. PCEP Extensions for IFIT Attributes

This document is to add IFIT attribute TLVs as PCEP Extensions. The following sections will describe the requirement and usage of different IFIT modes, and define the corresponding TLV encoding in PCEP.

The IFIT attributes here described can be generalized and included as TLVs carried inside the LSPA (LSF Attributes) object in order to be applied for all path types, as long as they support the relevant data plane telemetry method. IFIT Attributes TLVs are optional and can be taken into account by the PCE during path computation and by the PCC during path setup. In general, the LSPA object can be carried within a PCInitiate message, a PCUpd message, or a PCRpt message in the stateful PCE model.

In this document it is considered the case of SR Policy since IOAM and Alternate Marking are more mature especially for Segment Routing (SR) and for IPv6.

It is to be noted that, if it is needed to apply different IFIT methods for each Segment List, the IFIT attributes can be added into the PATH-ATTRIB object, instead of the LSPA object, according to [I-D.koldychev-pce-multipath] that defines PCEP Extensions for Signaling Multipath Information.
2.1. IFIT for SR Policies

RFC 8664 [RFC8664] and [I-D.ietf-pce-segment-routing-ipv6] specify extensions to the Path Computation Element Communication Protocol (PCEP) that allow a stateful PCE to compute and initiate Traffic-Engineering (TE) paths, as well as a Path Computation Client (PCC) to request a path subject to certain constraints and optimization criteria in SR networks both for SR-MPLS and SRv6.

IFIT attributes, here defined as TLVs for the LSPA object, complement both RFC 8664 [RFC8664], [I-D.ietf-pce-segment-routing-ipv6] and [I-D.ietf-pce-segment-routing-policy-cp].

3. IFIT capability advertisement TLV

During the PCEP initialization phase, PCEP speakers (PCE or PCC) SHOULD advertise their support of IFIT methods (e.g. IOAM and Alternate Marking).

A PCEP speaker includes the IFIT-CAPABILITY TLVs in the OPEN object to advertise its support for PCEP IFIT extensions. The presence of the IFIT-CAPABILITY TLV in the OPEN object indicates that the IFIT methods are supported.

RFC 8664 [RFC8664] and [I-D.ietf-pce-segment-routing-ipv6] define a new Path Setup Type (PST) for SR and also define the SR-PCE-CAPABILITY sub-TLV. This document defined a new IFIT-CAPABILITY TLV, that is an optional TLV for use in the OPEN Object for IFIT attributes via PCEP capability advertisement.

```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Type                |            Length=4           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           Flags                     |P|I|D|E|M|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Fig. 1 IFIT-CAPABILITY TLV Format

Where:

Type: to be assigned by IANA.

Length: 4.

Flags: The following flags are defined in this document:
P: IOAM Pre-allocated Trace Option Type-enabled flag  
[I-D.ietf-ippm-ioam-data]. If set to 1 by a PCC, the P flag indicates that the PCC allows instantiation of the IOAM Pre-allocated Trace feature by a PCE. If set to 1 by a PCE, the P flag indicates that the PCE supports the IOAM Pre-allocated Trace feature instantiation. The P flag MUST be set by both PCC and PCE in order to support the IOAM Pre-allocated Trace instantiation.

I: IOAM Incremental Trace Option Type-enabled flag  
[I-D.ietf-ippm-ioam-data]. If set to 1 by a PCC, the I flag indicates that the PCC allows instantiation of the IOAM Incremental Trace feature by a PCE. If set to 1 by a PCE, the I flag indicates that the PCE supports the relative IOAM Incremental Trace feature instantiation. The I flag MUST be set by both PCC and PCE in order to support the IOAM Incremental Trace feature instantiation.

D: IOAM DEX Option Type-enabled flag  
[I-D.ietf-ippm-ioam-direct-export]. If set to 1 by a PCC, the D flag indicates that the PCC allows instantiation of the relative IOAM DEX feature by a PCE. If set to 1 by a PCE, the D flag indicates that the PCE supports the relative IOAM DEX feature instantiation. The D flag MUST be set by both PCC and PCE in order to support the IOAM DEX feature instantiation.

E: IOAM E2E Option Type-enabled flag [I-D.ietf-ippm-ioam-data]. If set to 1 by a PCC, the E flag indicates that the PCC allows instantiation of the relative IOAM E2E feature by a PCE. If set to 1 by a PCE, the E flag indicates that the PCE supports the relative IOAM E2E feature instantiation. The E flag MUST be set by both PCC and PCE in order to support the IOAM E2E feature instantiation.

M: Alternate Marking enabled flag RFC 8321 [RFC8321]. If set to 1 by a PCC, the M flag indicates that the PCC allows instantiation of the relative Alternate Marking feature by a PCE. If set to 1 by a PCE, the M flag indicates that the PCE supports the relative Alternate Marking feature instantiation. The M flag MUST be set by both PCC and PCE in order to support the Alternate Marking feature instantiation.

Unassigned bits are considered reserved. They MUST be set to 0 on transmission and MUST be ignored on receipt.

Advertisement of the IFIT-CAPABILITY TLV implies support of IFIT methods (IOAM and/or Alternate Marking) as well as the objects, TLVs, and procedures defined in this document. It is worth mentioning that IOAM and Alternate Marking can be activated one at a time or can
coexist; so it is possible to have only IOAM or only Alternate Marking enabled but they are recognized in general as IFIT capability.

The IFIT Capability Advertisement can imply the following cases:

- The PCEP protocol extensions for IFIT MUST NOT be used if one or both PCEP speakers have not included the IFIT-CAPABILITY TLV in their respective OPEN message.
- A PCEP speaker that does not recognize the extensions defined in this document would simply ignore the TLVs as per RFC 5440 [RFC5440].
- If a PCEP speaker supports the extensions defined in this document but did not advertise this capability, then upon receipt of IFIT-ATTRIBUTES TLV in the LSP Attributes (LSPA) object, it SHOULD generate a PCErr with Error-Type 19 (Invalid Operation) with the relative Error-value "IFIT capability not advertised" and ignore the IFIT-ATTRIBUTES TLV.

4. IFIT Attributes TLV

The IFIT-ATTRIBUTES TLV provides the configurable knobs of the IFIT feature, and it can be included as an optional TLV in the LSPA object (as described in RFC 5440 [RFC5440]).

For a PCE-initiated LSP RFC 8281 [RFC8281], this TLV is included in the LSPA object with the PCInitiate message. For the PCC-initiated delegated LSPs, this TLV is carried in the Path Computation State Report (PCRpt) message in the LSPA object. This TLV is also carried in the LSPA object with the Path Computation Update Request (PCUpd) message to direct the PCC (LSP head-end) to make updates to IFIT attributes.

The TLV is encoded in all PCEP messages for the LSP if IFIT feature is enabled. The absence of the TLV indicates the PCEP speaker wishes to disable the feature. This TLV includes multiple IFIT-ATTRIBUTES sub-TLVs. The IFIT-ATTRIBUTES sub-TLVs are included if there is a change since the last information sent in the PCEP message. The default values for missing sub-TLVs apply for the first PCEP message for the LSP.

The format of the IFIT-ATTRIBUTES TLV is shown in the following figure:
Where:

Type: to be assigned by IANA.

Length: The Length field defines the length of the value portion in bytes as per RFC 5440 [RFC5440].

Value: This comprises one or more sub-TLVs.

The following sub-TLVs are defined in this document:

<table>
<thead>
<tr>
<th>Type</th>
<th>Len</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>8</td>
<td>IOAM Pre-allocated Trace Option</td>
</tr>
<tr>
<td>2</td>
<td>8</td>
<td>IOAM Incremental Trace Option</td>
</tr>
<tr>
<td>3</td>
<td>12</td>
<td>IOAM Directly Export Option</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>IOAM Edge-to-Edge Option</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>Enhanced Alternate Marking</td>
</tr>
</tbody>
</table>

Fig. 3 Sub-TLV Types of the IFIT-ATTRIBUTES TLV

4.1. IOAM Sub-TLVs

In-situ Operations, Administration, and Maintenance (IOAM) [I-D.ietf-ippm-ioam-data] records operational and telemetry information in the packet while the packet traverses a path between two points in the network. In terms of the classification given in RFC 7799 [RFC7799] IOAM could be categorized as Hybrid Type 1. IOAM mechanisms can be leveraged where active OAM do not apply or do not offer the desired results.
For the SR use case, when SR policy enables IOAM, the IOAM header will be inserted into every packet of the traffic that is steered into the SR paths. Since this document aims to define the control plane, it is to be noted that a relevant document for the data plane is [I-D.ietf-ippm-ioam-ipv6-options] for Segment Routing over IPv6 data plane (SRv6).

4.1.1. IOAM Pre-allocated Trace Option Sub-TLV

The IOAM tracing data is expected to be collected at every node that a packet traverses to ensure visibility into the entire path a packet takes within an IOAM domain. The preallocated tracing option will create pre-allocated space for each node to populate its information.

The format of IOAM pre-allocated trace option Sub-TLV is defined as follows:

```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-------------------------------+-------------------------------+
|          Type=1               |           Length=8            |
|-------------------------------+---------------------------------------------------------------+
|       Namespace ID            |            Rsvd1              |
|-------------------------------+-----------------------+-------+
|         IOAM Trace Type                      | Flags  | Rsvd2 |
|----------------------------------------------+--------+-------+
```

Fig. 4 IOAM Pre-allocated Trace Option Sub-TLV

Where:

Type: 1 (to be assigned by IANA).

Length: 8. It is the total length of the value field not including Type and Length fields.

Namespace ID: A 16-bit identifier of an IOAM-Namespace. The definition is the same as described in section 4.4 of [I-D.ietf-ippm-ioam-data].

IOAM Trace Type: A 24-bit identifier which specifies which data types are used in the node data list. The definition is the same as described in section 4.4 of [I-D.ietf-ippm-ioam-data].

Flags: A 4-bit field. The definition is the same as described in [I-D.ietf-ippm-ioam-flags] and section 4.4 of [I-D.ietf-ippm-ioam-data].
Rsvd1: A 16-bit field reserved for further usage. It MUST be zero and ignored on receipt.

Rsvd2: A 4-bit field reserved for further usage. It MUST be zero and ignored on receipt.

4.1.2. IOAM Incremental Trace Option Sub-TLV

The incremental tracing option contains a variable node data fields where each node allocates and pushes its node data immediately following the option header.

The format of IOAM incremental trace option Sub-TLV is defined as follows:

```plaintext
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-------------------------------+-------------------------------+
|          Type=2               |           Length=8            |
+---------------------------------------------------------------+
|       Namespace ID            |            Rsvd1              |
+-------------------------------+-----------------------+-------+
|         IOAM Trace Type                      | Flags  | Rsvd2 |
+----------------------------------------------+--------+-------+
```

Fig. 5 IOAM Incremental Trace Option Sub-TLV

Where:

Type: 2 (to be assigned by IANA).

Length: 8. It is the total length of the value field not including Type and Length fields.

All the other fields definition is the same as the pre-allocated trace option Sub-TLV in the previous section.

4.1.3. IOAM Directly Export Option Sub-TLV

IOAM directly export option is used as a trigger for IOAM data to be directly exported to a collector without being pushed into in-flight data packets.

The format of IOAM directly export option Sub-TLV is defined as follows:
Fig. 6 IOAM Directly Export Option Sub-TLV

Where:

Type: 3 (to be assigned by IANA).

Length: 12. It is the total length of the value field not including Type and Length fields.

Namespace ID: A 16-bit identifier of an IOAM-Namespace. The definition is the same as described in section 4.4 of [I-D.ietf-ippm-ioam-data].

IOAM Trace Type: A 24-bit identifier which specifies which data types are used in the node data list. The definition is the same as described in section 4.4 of [I-D.ietf-ippm-ioam-data].

Flags: A 16-bit field. The definition is the same as described in section 3.2 of [I-D.ietf-ippm-ioam-direct-export].

Flow ID: A 32-bit flow identifier. The definition is the same as described in section 3.2 of [I-D.ietf-ippm-ioam-direct-export].

Rsvd: A 4-bit field reserved for further usage. It MUST be zero and ignored on receipt.

4.1.4. IOAM Edge-to-Edge Option Sub-TLV

The IOAM edge to edge option is to carry data that is added by the IOAM encapsulating node and interpreted by IOAM decapsulating node.

The format of IOAM edge-to-edge option Sub-TLV is defined as follows:
Fig. 7 IOAM Edge-to-Edge Option Sub-TLV

Where:

Type: 4 (to be assigned by IANA).

Length: 4. It is the total length of the value field not including Type and Length fields.

Namespace ID: A 16-bit identifier of an IOAM-Namespace. The definition is the same as described in section 4.6 of [I-D.ietf-ippm-ioam-data].

IOAM E2E Type: A 16-bit identifier which specifies which data types are used in the E2E option data. The definition is the same as described in section 4.6 of [I-D.ietf-ippm-ioam-data].

4.2. Enhanced Alternate Marking Sub-TLV

The Alternate Marking [RFC8321] technique is an hybrid performance measurement method, per RFC 7799 [RFC7799] classification of measurement methods. Because this method is based on marking consecutive batches of packets, it can be used to measure packet loss, latency, and jitter on live traffic.

For the SR use case, since this document aims to define the control plane, it is to be noted that a relevant document for the data plane is [I-D.ietf-6man-ipv6-alt-mark] for Segment Routing over IPv6 data plane (SRv6).

The format of Enhanced Alternate Marking (EAM) Sub-TLV is defined as follows:
Fig. 8 Enhanced Alternate Marking Sub-TLV

Where:

Type: 5 (to be assigned by IANA).

Length: 4. It is the total length of the value field not including Type and Length fields.

FlowMonID: A 20-bit identifier to uniquely identify a monitored flow within the measurement domain. The definition is the same as described in section 5.3 of [I-D.ietf-6man-ipv6-alt-mark]. It is to be noted that PCE also needs to maintain the uniqueness of FlowMonID as described in [I-D.ietf-6man-ipv6-alt-mark].

Period: Time interval between two alternate marking period. The unit is second.

Flags: A 4-bits field. Two flags are currently assigned:

H: A flag indicating that the measurement is Hop-By-Hop.

E: A flag indicating that the measurement is End-to-End.

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

5. PCEP Messages

5.1. The PCInitiate Message

A PCInitiate message is a PCEP message sent by a PCE to a PCC to trigger LSP instantiation or deletion RFC 8281 [RFC8281].

For the PCE-initiated LSP with the IFIT feature enabled, IFIT-ATTRIBUTES TLV MUST be included in the LSPA object with the PCInitiate message.

The Routing Backus-Naur Form (RBNF) definition of the PCInitiate message RFC 8281 [RFC8281] is unchanged by this document.
5.2. The PCUpd Message

A PCUpd message is a PCEP message sent by a PCE to a PCC to update the LSP parameters RFC 8231 [RFC8231].

For PCE-initiated LSPs with the IFIT feature enabled, the IFIT-ATTRIBUTES TLV MUST be included in the LSPA object with the PCUpd message. The PCE can send this TLV to direct the PCC to change the IFIT parameters.

The RBNF definition of the PCUpd message RFC 8231 [RFC8231] is unchanged by this document.

5.3. The PCRpt Message

The PCRpt message RFC 8231 [RFC8231] is a PCEP message sent by a PCC to a PCE to report the status of one or more LSPs.

For PCE-initiated LSPs RFC 8281 [RFC8281], the PCC creates the LSP using the attributes communicated by the PCE and the local values for the unspecified parameters. After the successful instantiation of the LSP, the PCC automatically delegates the LSP to the PCE and generates a PCRpt message to provide the status report for the LSP.

The RBNF definition of the PCRpt message RFC 8231 [RFC8231] is unchanged by this document.

For both PCE-initiated and PCC-initiated LSPs, when the LSP is instantiated the IFIT methods are applied as specified for the corresponding data plane. [I-D.ietf-ippm-ioam-ipv6-options] and [I-D.ietf-6man-ipv6-alt-mark] are the relevant documents for Segment Routing over IPv6 data plane (SRv6).

6. Example of application to SR Policy

A PCC or PCE sets the IFIT-CAPABILITY TLV in the Open message during the PCEP initialization phase to indicate that it supports the IFIT procedures.

[I-D.ietf-pce-segment-routing-policy-cp] defines the PCEP extension to support Segment Routing Policy Candidate Paths and in this regard the SRPAG Association object is introduced.

The Examples of PCC Initiated SR Policy with single or multiple candidate-paths and PCE Initiated SR Policy with single or multiple candidate-paths are reported in [I-D.ietf-pce-segment-routing-policy-cp].
In case of PCC Initiated SR Policy, PCC sends PCReq message to the PCE, encoding the SRPAG ASSOCIATION object and IFIT-ATTRIBUTES TLV via the LSPA object. This is valid for both single and multiple candidate-paths. Finally PCE returns the path in PCRep message, and echoes back the SRPAG object that were used in the computation and IFIT LSPA TLVs too. Additionally, PCC sends PCRpt message to the PCE, including the LSP object and the SRPAG ASSOCIATION object and IFIT-ATTRIBUTES TLV via the LSPA object. Then PCE computes path and finally PCE updates the SR policy candidate path’s ERO using PCUpd message considering the IFIT LSPA TLVs too.

In case of PCE Initiated SR Policy, PCE sends PCInitiate message, containing the SRPAG Association object and IFIT-ATTRIBUTES TLV via the LSPA object. This is valid for both single and multiple candidate-paths. Then PCC uses the color, endpoint and preference from the SRPAG object to create a new candidate path considering the IFIT LSPA TLVs too. Finally PCC sends a PCRpt message back to the PCE to report the newly created Candidate Path. The PCRpt message contains the SRPAG Association object and IFIT-ATTRIBUTES information.

The procedure of enabling/disabling IFIT is simple, indeed the PCE can update the IFIT-ATTRIBUTES of the LSP by sending subsequent Path Computation Update Request (PCUpd) messages. PCE can update the IFIT-ATTRIBUTES of the LSP by sending Path Computation State Report (PCRpt) messages.

7. IANA Considerations

This document defines the new IFIT-CAPABILITY TLV and IFIT-ATTRIBUTES TLV.

7.1. PCEP TLV Type Indicators

IANA is requested to make the assignment from the "PCEP TLV Type Indicators" subregistry of the "Path Computation Element Protocol (PCEP) Numbers" registry as follows:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>TBD1</td>
<td>IFIT-CAPABILITY TLV</td>
<td>This document</td>
</tr>
<tr>
<td>TBD2</td>
<td>IFIT-ATTRIBUTES TLV</td>
<td>This document</td>
</tr>
</tbody>
</table>
7.2. IFIT-CAPABILITY TLV Flags field

This document specifies the IFIT-CAPABILITY TLV 32-bits Flags field. IANA is requested to create a registry to manage the value of the IFIT-CAPABILITY TLV’s Flags field within the "Path Computation Element Protocol (PCEP) Numbers" registry.

New values are to be assigned by Standards Action RFC 8126 [RFC8126]. Each bit should be tracked with the following qualities:

* Bit number (count from 0 as the most significant bit)
* Flag Name
* Reference

IANA is requested to set 5 new bits in the IFIT-CAPABILITY TLV Flags Field registry, as follows:

<table>
<thead>
<tr>
<th>Bit no.</th>
<th>Flag Name</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-26</td>
<td>Unassigned</td>
<td>This document</td>
</tr>
<tr>
<td>27</td>
<td>P: IOAM Pre-allocated Trace Option flag</td>
<td>This document</td>
</tr>
<tr>
<td>28</td>
<td>I: IOAM Incremental Trace Option flag</td>
<td>This document</td>
</tr>
<tr>
<td>29</td>
<td>D: IOAM Directly Export Option flag</td>
<td>This document</td>
</tr>
<tr>
<td>30</td>
<td>E: IOAM Edge-to-Edge Option</td>
<td>This document</td>
</tr>
<tr>
<td>31</td>
<td>M: Alternate Marking Flag</td>
<td>This document</td>
</tr>
</tbody>
</table>

7.3. IFIT-ATTRIBUTES Sub-TLV

This document also specifies the IFIT-ATTRIBUTES sub-TLVs. IANA is requested to create an "IFIT-ATTRIBUTES Sub-TLV Types" subregistry within the "Path Computation Element Protocol (PCEP) Numbers" registry.

IANA is requested to set the Registration Procedure for this registry to read as follows:
This document defines the following types:

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Reserved</td>
<td>This document</td>
</tr>
<tr>
<td>1</td>
<td>IOAM Pre-allocated Trace Option</td>
<td>This document</td>
</tr>
<tr>
<td>2</td>
<td>IOAM Incremental Trace Option</td>
<td>This document</td>
</tr>
<tr>
<td>3</td>
<td>IOAM Directly Export Option</td>
<td>This document</td>
</tr>
<tr>
<td>4</td>
<td>IOAM Edge-to-Edge Option</td>
<td>This document</td>
</tr>
<tr>
<td>5</td>
<td>Enhanced Alternate Marking</td>
<td>This document</td>
</tr>
<tr>
<td>6-65503</td>
<td>Unassigned</td>
<td>This document</td>
</tr>
<tr>
<td>65504-65535</td>
<td>Experimental Use</td>
<td>This document</td>
</tr>
</tbody>
</table>

7.4.  Enhanced Alternate Marking Sub-TLV Flags field

This document specifies the Enhanced Alternate Marking Sub-TLV 4-bits Flags field. IANA is requested to create a registry to manage the value of the Enhanced Alternate Marking Sub-TLV’s Flags field within the "Path Computation Element Protocol (PCEP) Numbers" registry.

New values are to be assigned by Standards Action RFC 8126 [RFC8126]. Each bit should be tracked with the following qualities:

* Bit number (count from 0 as the most significant bit)

* Flag Name

* Reference

IANA is requested to set 2 new bits in the IFIT-CAPABILITY TLV Flags Field registry, as follows:
### 7.5. PCEP Error Codes

This document defines a new Error-value for PCErr message of Error-Type 19 (Invalid Operation). IANA is requested to allocate a new Error-value within the "PCEP-ERROR Object Error Types and Values" subregistry of the "Path Computation Element Protocol (PCEP) Numbers" registry as follows:

<table>
<thead>
<tr>
<th>Error-Type</th>
<th>Meaning</th>
<th>Error-value</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>19</td>
<td>Invalid Operation capability not advertised</td>
<td>TBD3: IFIT</td>
<td>This document</td>
</tr>
</tbody>
</table>

### 8. Security Considerations

This document defines the new IFIT-CAPABILITY TLV and IFIT Attributes TLVs, which do not add any substantial new security concerns beyond those already discussed in RFC 8231 [RFC8231] and RFC 8281 [RFC8281] for stateful PCE operations. As per RFC 8231 [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) RFC 8253 [RFC8253], as per the recommendations and best current practices in BCP 195 RFC 7525 [RFC7525] (unless explicitly set aside in RFC 8253 [RFC8253]).

Implementation of IFIT methods (IOAM and Alternate Marking) are mindful of security and privacy concerns, as explained in [I-D.ietf-ippm-ioam-data] and RFC 8321 [RFC8321]. Anyway incorrect IFIT parameters in the IFIT-ATTRIBUTES sub-TLVs SHOULD NOT have an adverse effect on the LSP as well as on the network, since it affects only the operation of the telemetry methodology.

IFIT data MUST be propagated in a limited domain in order to avoid malicious attacks and solutions to ensure this requirement are respectively discussed in [I-D.ietf-ippm-ioam-data] and [I-D.ietf-6man-ipv6-alt-mark].
IFIT methods (IOAM and Alternate Marking) are applied within a controlled domain where the network nodes are locally administered. A limited administrative domain provides the network administrator with the means to select, monitor and control the access to the network, making it a trusted domain also for the PCEP extensions defined in this document.

9. Contributors

The following people provided relevant contributions to this document:

Yali Wang
Huawei
Email: wangyali11@huawei.com

Huanan Chen
independent
Email: -

Dhruv Doody
Huawei Technologies
Email: dhruv.ietf@gmail.com

10. Acknowledgements

The authors of this document would like to thank Huaimo Chen for the comments and review of this document.

11. References

11.1. Normative References

[I-D.ietf-6man-ipv6-alt-mark]

[I-D.ietf-ippm-ioam-data]
[I-D.ietf-ippm-ioam-direct-export]

[I-D.ietf-ippm-ioam-flags]
Mizrahi, T., Brockners, F., Bhandari, S., Gafni, B., and M. Spiegel, "In-situ OAM Loopback and Active Flags", draft-ietf-ippm-ioam-flags-09 (work in progress), June 2022.

[I-D.ietf-ippm-ioam-ipv6-options]


Internet-Draft                  pcep-ifit                      July 2022

11.2. Informative References


Authors’ Addresses

Hang Yuan
UnionPay
1899 Gu-Tang Rd., Pudong
Shanghai
China

Email: yuanhang@unionpay.com

Xuerong Wang
China Telecom
Guangzhou
China

Email: wangxr4@chinatelecom.cn

Tianran Zhou
Huawei
156 Beiqing Rd., Haidian District
Beijing
China

Email: zhoutianran@huawei.com
Weidong Li  
Huawei  
156 Beiqing Rd., Haidian District  
Beijing  
China  

Email: poly.li@huawei.com

Giuseppe Fioccola  
Huawei  
Riesstrasse, 25  
Munich  
Germany  

Email: giuseppe.fioccola@huawei.com
Abstract

This document updates, simplifies and clarifies certain aspects of the PCEP protocol. The content of this document has been compiled based on several interop exercises.

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.

Status of This Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 2 January 2023.
1. Introduction

The PCEP protocol started off being purely stateless with PCReq and PCReply messages, see Path Computation Element (PCE) Communication Protocol (PCEP) [RFC5440]. Stateless PCEP operates in a "pull" model, i.e., PCC has to periodically ask the PCE for updates to the path, even if the path has not changed.

Stateful PCEP was later introduced in PCEP Extensions for the Stateful PCE Model [RFC8231]. Stateful PCEP operates in a "push" model, where the PCC can register with PCE to receive future updates
about the path, and there is no need to ask the PCE periodically. The current document serves to optimize the original procedure in [RFC8231] to optionally drop the PCReq and PCReply exchange, which greatly simplifies implementation and optimizes the protocol.

Due to different interpretations of PCEP standards, it was found that implementations often had to adjust their behavior in order to interoperate. The current document serves to clarify certain aspects of PCEP to make it easier to produce interoperable implementations of PCEP.

2. Terminology

The following terminologies are used in this document:

PCC: Path Computation Client. Any client application requesting a path computation to be performed by a Path Computation Element.

PCE: Path Computation Element. An entity (component, application, or network node) that is capable of computing a network path or route based on a network graph and applying computational constraints.

PCEP: Path Computation Element Protocol.

MBB: Make-Before-Break. A procedure during which the head-end of a traffic-engineered path wishes to move traffic to a new path without losing any traffic, by first "making" a new path and then "breaking" the old path.

Association parameters: As described in [RFC8697], the combination of the mandatory fields Association type, Association ID and Association Source in the ASSOCIATION object uniquely identify the association group. If the optional TLVs - Global Association Source or Extended Association ID are included, then they MUST be included in combination with mandatory fields to uniquely identify the association group.

Association information: As described in [RFC8697], the ASSOCIATION object could also include other optional TLVs based on the association types, that provides 'information' related to the association type.

ERO: Explicit Route Object is the path of the LSP encoded into a PCEP object. To represent an empty ERO object, i.e., without any subobjects, we use the notation "ERO={}". To represent an ERO object containing some given sequence of subobjects, we use the notation "ERO=(A)".
3. PCEP LSP Database

We use the concept of the LSP-DB, as a database of actual LSP state in the network, to illustrate the internal state of PCEP speakers in response to various PCEP messages.

Note that the term "LSP", which stands for "Label Switched Path", if taken too literally would restrict our discussion to MPLS dataplane only. We take the term "LSP" to apply to non-MPLS paths as well, to avoid changing the name. Alternatively, we could rename LSP to "Instance".

3.1. Structure

LSP-DB contains two types of objects: LSPs and Tunnels. An LSP is identified by the LSP-IDENTIFIERS TLV. A Tunnel is identified by the PLSP-ID in the LSP object and/or the SYMBOLIC-NAME. See [RFC8231].

A Tunnel may or may not be an actual tunnel on the router. For example, working and protect paths can be implemented as a single tunnel interface, but in PCEP we would refer to them as two different Tunnels, because they would have different PLSP-IDs.

An LSP can be thought of as a instance of a Tunnel. In steady-state, a Tunnel has only one LSP, but during make-before-break (see [RFC3209]) it can have multiple LSPs, to represent both new and old instances that exist simultaneously for a time.

3.2. Synchronization

Both PCE and PCC maintain their separate copies of the LSP-DB. The PCE LSP-DB is only modified by PCRpt messages, no other PCEP message may modify the PCE LSP-DB. The PCC LSP-DB is built from actual forwarding state that PCC has installed. PCC uses PCRpt messages to synchronize its local LSP-DB to the PCE.

The PCE MUST always act on the latest state of the PCE LSP DB. Note that this does not mean that the PCE cannot use information from outside of LSP-DB. For example, the PCE can use other mechanisms to collect traffic statistics and use them in the computation. However, these traffic statistics are not part of the LSP-DB, but only reference it.

The LSP-DB on both the PCC and the PCE only stores the actual state in the network, it does not store the desired state. For example, consider the case of PCE Initiated LSP, configured on the PCE. When the operator modifies the configuration of this LSP, that is a change in desired state. The actual state has not yet changed, so LSP-DB is...
not modified yet. The LSP-DB is only modified after the PCE sends PCInit/PCUpd message to the PCC and the PCC decides to act on that message. When the PCC acts on a message from a PCE, it would update its own PCC LSP DB and send a PCRpt to the PCE(s) to synchronize the change. When the PCE receives the PCRpt msg, it updates its own PCE LSP DB. After this, the PCC LSP-DB and PCE LSP-DB are in sync.

3.3. Stateful Bringup

[RFC8231] Section 5.8.2, allows delegation of an LSP in operationally down state, but at the same time mandates the use of PCReq, before sending PCRpt. In this document, we would like to make it clear that sending PCReq is optional.

We shall refer to the process of sending PCReq before PCRpt as "stateless bringup". In reality, stateless bringup introduces overhead and is not possible to enforce from the PCE, because the stateless PCE is not required to keep any per-LSP state about previous PCReq messages. It was found that many vendors choose to ignore this requirement and send the PCRpt directly, without going through PCReq. Even though this behavior is against [RFC8231], it offers some advantages and simplifications, as will be explained in this section. This document therefore updates [RFC8231].

Even though all the major vendors today are moving to the stateful PCE model, it does not deprecate the need for stateless PCEP. The key property of stateless PCEP is that PCReq messages do not modify the state of the PCE LSP-DB. Therefore, PCReq messages are useful for many OAM ping/traceroute applications where the PCC wishes to probe the network topology without having any effect on the existing LSPs.

3.3.1. Updates to RFC 8231

[RFC8231] Section 5.8.2, says "The only explicit way for a PCC to request a path from the PCE is to send a PCReq message. The PCRpt message MUST NOT be used by the PCC to attempt to request a path from the PCE." In this document we update [RFC8231] to remove the quoted text.

As part of the new bringup procedure, the PCC MAY delegate an empty LSP (no ERO or empty ERO) to the PCE and then wait for the PCE to send PCUpd, without sending PCReq. We shall refer to this process as "stateful bringup". The PCE MUST support the original stateless bringup, for backward compatibility purposes. Supporting stateful bringup should not require introducing any new behavior on the PCE, because as mentioned earlier, the PCE does not modify LSP-DB state
based on PCReq messages. So whether the PCE has received a PCReq or not, it would process the PCRpt all the same.

An example of stateful bringup follows. In our example the PCC starts off by using LSP-ID of 0. The value 0 does not hold any special meaning, any other 16-bit value could have been used.

PCC has no LSP yet, but wants to establish a path. PCC sends PCRpt (R-FLAG=0, D-flag=1, OPER-FLAG=DOWN, PLSP-ID=100, LSP-ID=0, ERO={}).

<table>
<thead>
<tr>
<th>TUNNEL</th>
<th>LSP</th>
</tr>
</thead>
<tbody>
<tr>
<td>PLSP-ID=100</td>
<td>LSP-ID=0, D-flag=1, OPER=DOWN, ERO={}</td>
</tr>
</tbody>
</table>

Figure 1: Content of LSP DB

PCC received a PCUpd from the PCE and has decided to install the ERO={A} from that PCUpd. PCC sends PCRpt (R-FLAG=0, D-flag=1, OPER-FLAG=UP, PLSP-ID=100, LSP-ID=0, ERO={A}).

<table>
<thead>
<tr>
<th>TUNNEL</th>
<th>LSP</th>
</tr>
</thead>
<tbody>
<tr>
<td>PLSP-ID=100</td>
<td>LSP-ID=0, D-flag=1, OPER=UP, ERO={A}</td>
</tr>
</tbody>
</table>

Figure 2: Content of LSP DB

3.4. Successful MBB

Below we give an example of doing MBB to switch the Tunnel from one path to another. We represent the path encoded into the ERO object as ERO={A} and ERO={B}.

PCC has an existing LSP in UP state, with LSP-ID=2. PCC sends PCRpt (R-FLAG=0, PLSP-ID=100, LSP-ID=2, ERO={A}, OPER-FLAG=UP).

<table>
<thead>
<tr>
<th>TUNNEL</th>
<th>LSP</th>
</tr>
</thead>
<tbody>
<tr>
<td>PLSP-ID=100</td>
<td>LSP-ID=2, ERO={A}, OPER=UP</td>
</tr>
</tbody>
</table>

Figure 3: Content of LSP DB
PCC initiates the MBB procedure by creating a new LSP with LSP-ID=3. It does not matter what triggered the creation of the new LSP, it could have been due to a new path received via PCUpd (if the given Tunnel is delegated), or it could have been local computation on the PCC (if the Tunnel is locally computed on the PCC), or it could have been a change in configuration on the PCC (if the Tunnel’s path is explicitly configured on the PCC). It is important to emphasize that the procedure for updating the LSP-DB is common, regardless of the trigger that caused the change.

PCC sends PCRpt(R-FLAG=0, PLSP-ID=100, LSP-ID=3, ERO={B}, OPER-FLAG=UP).

```
+-----------------+---------------------------------------------+
| TUNNEL          | LSP                                         |
+-----------------+---------------------------------------------+
| PLSP-ID=100     | LSP-ID=2, ERO={A}, OPER=UP                  |
|                 | LSP-ID=3, ERO={B}, OPER=UP                  |
+---------------------------------------------------------------+
```

Figure 4: Content of LSP DB

After traffic has successfully switched to the new LSP, the PCC cleans up the old LSP. PCC sends PCRpt(R-FLAG=1, PLSP-ID=100, LSP-ID=2).

```
+-----------------+---------------------------------------------+
| TUNNEL          | LSP                                         |
+-----------------+---------------------------------------------+
| PLSP-ID=100     | LSP-ID=3, ERO={B}, OPER=UP                  |
+---------------------------------------------------------------+
```

Figure 5: Content of LSP DB

3.5. Aborted MBB

The MBB process can abort when the newly created LSP is destroyed before it is installed as traffic carrying. This scenario is described below.

PCC has an existing LSP in UP state, with LSP-ID=2. PCC sends PCRpt(R-FLAG=0, OPER-FLAG=UP, PLSP-ID=100, LSP-ID=2).

```
+-----------------+---------------------------------------------+
| TUNNEL          | LSP                                         |
+-----------------+---------------------------------------------+
| PLSP-ID=100     | LSP-ID=2, OPER=UP                           |
+---------------------------------------------------------------+
```

MBB procedure is initiated, a new LSP is created with LSP-ID=3. LSP is currently being established, so its oper state is DOWN. PCC sends PCRpt(R-FLAG=0, OPER-FLAG=DOWN, PLSP-ID=100, LSP-ID=3).

<table>
<thead>
<tr>
<th>TUNNEL</th>
<th>LSP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PLSP-ID=100</td>
</tr>
<tr>
<td></td>
<td>LSP-ID=2, OPER=UP</td>
</tr>
<tr>
<td></td>
<td>LSP-ID=3, OPER=DOWN</td>
</tr>
</tbody>
</table>

MBB procedure is aborted. PCC sends PCRpt(R-FLAG=1, PLSP-ID=100, LSP-ID=3).

<table>
<thead>
<tr>
<th>TUNNEL</th>
<th>LSP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PLSP-ID=100</td>
</tr>
<tr>
<td></td>
<td>LSP-ID=2, OPER=UP</td>
</tr>
</tbody>
</table>

PCEP Association Database

PCEP Association is a group of zero or more LSPs.

The PCE ASSO DB is populated by PCRpt messages and/or via configuration on the PCE itself. An Association is identified by the Association Parameters. The Association parameters contain many fields, so for convenience we will group all the fields into a single value. We will use ASSO_PARAM=A, ASSO_PARAM=B, to refer to different PCEP Associations: A and B, respectively.

4.1. 2 LSPs in same Association

Below, we give an example to illustrate how LSPs join the same Association.

PCC creates the first LSP. PCC sends PCRpt(R-FLAG=0, PLSP-ID=100, LSP-ID=1, ASSO_PARAM=A, ASSO_R_FLAG=0).
PCC creates the second LSP. PCC sends PCRpt(R-FLAG=0, PLSP-ID=200, LSP-ID=1, ASSO_PARAM=A, ASSO_R_FLAG=0).

<table>
<thead>
<tr>
<th>ASSO</th>
<th>LSP</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASSO_PARAM=A</td>
<td>PLSP-ID=100, LSP-ID=1</td>
</tr>
<tr>
<td></td>
<td>PLSP-ID=200, LSP-ID=1</td>
</tr>
</tbody>
</table>

Figure 10: Content of PCE ASSO DB

PCC updates the first LSP, the PCC is NOT REQUIRED to send the ASSOCIATION object in this PCRpt, since the LSP is already in the Association. PCC sends PCRpt(R-FLAG=0, PLSP-ID=100, LSP-ID=1). The content of the PCE ASSO DB is unchanged. Note that the PCC sends the ASSOCIATION OBJECT in the first PCRpt during SYNC state, even if it has already issued a PCRpt with the association object sometime in the past with this PCE. The synchronization steps outlined in [RFC8697] are to be followed.

<table>
<thead>
<tr>
<th>ASSO</th>
<th>LSP</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASSO_PARAM=A</td>
<td>PLSP-ID=100, LSP-ID=1</td>
</tr>
<tr>
<td></td>
<td>PLSP-ID=200, LSP-ID=1</td>
</tr>
</tbody>
</table>

Figure 11: Content of PCE ASSO DB

PCC decides to delete the second LSP. PCC sends PCRpt(R-FLAG=1, PLSP-ID=200, LSP-ID=1).

<table>
<thead>
<tr>
<th>ASSO</th>
<th>LSP</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASSO_PARAM=A</td>
<td>PLSP-ID=100, LSP-ID=1</td>
</tr>
</tbody>
</table>

Figure 12: Content of PCE ASSO DB
PCC decides to remove the first LSP from the Association, but not delete the LSP itself. PCC sends PCRpt(R-FLAG=0, PLSP-ID=100, LSP-ID=1, ASSO_PARAM=A, ASSO_R_FLAG=1). The PCE ASSO DB is now empty.

<table>
<thead>
<tr>
<th>ASSO</th>
<th>LSP</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASSO_PARAM=A</td>
<td></td>
</tr>
</tbody>
</table>

Figure 13: Content of PCE ASSO DB

4.2. Switch Association during MBB

Below, we give an example to illustrate how a Tunnel goes through MBB and switches from Association A to Association B.

Each new LSP (identified by the LSP-ID) does not inherit the Association membership of any previous LSPs within the same Tunnel. This is so that a Tunnel can have two LSPs that are in different Associations, this may be done when switching from one Association to another.

PCC creates the first LSP. PCC sends PCRpt(R-FLAG=0, PLSP-ID=100, LSP-ID=1, ASSO_PARAM=A, ASSO_R_FLAG=0).

<table>
<thead>
<tr>
<th>ASSO</th>
<th>LSP</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASSO_PARAM=A</td>
<td>PLSP-ID=100, LSP-ID=1</td>
</tr>
</tbody>
</table>

Figure 14: Content of PCE ASSO DB

PCC creates the MBB LSP in a different Association. PCC sends PCRpt(R-FLAG=0, PLSP-ID=100, LSP-ID=2, ASSO_PARAM=B, ASSO_R_FLAG=0).

<table>
<thead>
<tr>
<th>ASSO</th>
<th>LSP</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASSO_PARAM=A</td>
<td>PLSP-ID=100, LSP-ID=1</td>
</tr>
<tr>
<td>ASSO_PARAM=B</td>
<td>PLSP-ID=100, LSP-ID=2</td>
</tr>
</tbody>
</table>

Figure 15: Content of PCE ASSO DB
PCC deletes the old LSP. PCC sends PCRpt(R-FLAG=1, PLSP-ID=100, LSP-ID=1).

+---------------------------------------------------------------+
| ASSO            | LSP                                         |
| ASSO_PARAM=B    | PLSP-ID=100, LSP-ID=2                       |
+---------------------------------------------------------------+

Figure 16: Content of PCE ASSO DB

5. Computation Constraints

For any PCEP object that does not have an explicit removal flag, the absence of that object indicates removal of the constraint specified by that object. For example, suppose the first state-report contains an LSPA object with some affinity constraints. Then if a subsequent state-report does not contain an LSPA object, then this means that the previously specified affinity constraints do not apply anymore. Same applies to all PCEP objects, like METRIC, BANDWIDTH, etc., which do not have an explicit flag for removal. This simply ensures that it is possible to remove a constraint without using an explicit removal flag.

6. Use of SR-RRO and SRv6-RRO objects


In practice RRO data is the result of signalling via a protocol such as RSVP-TE, which allows collection of per-hop information along the path. The ERO and RRO values may be different as the path encoded in the ERO may differ than the RRO such as during protection conditions or if the ERO contains loose hops which are expanded upon. As Segment Routing LSP does not perform any signalling, the values of an SR-ERO/SRv6-ERO and SR-RRO/SRv6-RRO (respectively) are in practice the same, therefore some implementations have omitted the RRO when reporting a SR-TE LSP while others continue to send both ERO and RRO values.

The following applies to SR-TE only. If both ERO and RRO are present for the same LSP, it SHOULD be interpreted as the RRO being the actual path the LSP is taking but MAY interpret only the ERO as the actual path. In the absence of RRO a PCE MUST interpret the ERO as the actual path for the LSP. Until SR-TE introduces some form of
signaling similar to RSVP-TE, the use of RRO is discouraged for SR-TE LSPs.

7. Security Considerations

None at this time.

8. IANA Considerations

None at this time.

9. Acknowledgement

The authors would like to thank Adrian Farrel for useful review comments.

10. Normative References


Appendix A. Contributors

Dhruv Dhody
Huawei Technologies
Divyashree Techno Park, Whitefield
Bangalore, Karnataka  560066
India

Email: dhrv.ietf@gmail.com

Andrew Stone
Nokia
Ottawa, Canada

Email: andrew.stone@nokia.com

Samuel Sidor
Cisco Systems
Bratislava, Slovakia

Email: ssidor@cisco.com

Mahendra Singh Negi
RtBrick Inc
N-17L, 18th Cross Rd, HSR Layout
Bangalore, Karnataka  560102
India

Email: mahend.ietf@gmail.com

Authors’ Addresses
Abstract

This document proposes a set of extensions for Path Computation Element Communication Protocol (PCEP) for Circuit Style Policies - Segment-Routing Policy designed to satisfy requirements for connection-oriented transport services. New TLV is introduced to control path recomputation and new flag to add ability to request path with strict hops only.

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.

Status of This Memo

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

1. Introduction ............................................. 3
2. Terminology .............................................. 3
3. Overview of Extensions to PCEP .............................. 4
  3.1. LSP-EXTENDED-FLAG TLV ............................ 4
  3.2. PATH-RECOMPUTATION TLV ............................ 4
4. Operation ............................................... 5
  4.1. Strict path enforcement .............................. 5
  4.2. Path recomputation .................................. 6
5. Security Considerations .................................. 6
6. IANA Considerations ..................................... 6
  6.1. LSP-EXTENDED-FLAG TLV ............................ 7
  6.2. PATH-RECOMPUTATION TLV ............................ 7
7. References .............................................. 7
  7.1. Normative References ............................... 7
  7.2. Informative References ............................. 8
Authors’ Addresses ........................................ 8
1. Introduction

Usage of Segment-routing and PCEP in connection-oriented transport services require path persistancy and hop-by-hop behavior for PCE computed paths.

Circuit-Style Policy introduced in [I-D.schmutzer-pce-cs-sr-policy] requires PCEP extensions, which are covered in this document.

This document:

* Introduces possibility to request strict path from the PCE by extending LSP-EXTENDED-FLAG TLV

* Adding new TLV to encode information about disabling path recomputation for specific path to the PCE, to be carried inside the LSPA object, which is defined in [RFC5440].

* Clarifies usage of existing O-flag from RP object in Segment-routing

PCEP extensions described in this document are applicable to RSVP-TE and SR-TE.

2. Terminology

The following terminologies are used in this document:

ERO: Explicit Route Object

IGP: Interior Gateway Protocol

LSP: Label Switched Path.

LSPA: Label Switched Path Attributes.

OTN: Optical Transport Network.

PCC: Path Computation Client

PCE: Path Computation Element

PCEP: Path Computation Element Protocol.

SDH: Synchronous Digital Hierarchy

SID: Segment Identifier
SONET: Synchronous Optical Network

SR: Segment Routing.


3. Overview of Extensions to PCEP

3.1. LSP-EXTENDED-FLAG TLV

O-flag is proposed in the LSP-EXTENDED-FLAG TLV, which was introduced in 5.1.2 of [I-D.ietf-pce-lsp-extended-flags] and extended with E-flag in [I-D.peng-pce-entropy-label-position]

The format of the LSP-EXTENDED-FLAG is as follows:

```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Type=TBD1           |          Length               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                           |O|E|
//                 LSP Extended Flags                          //
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 1: LSP-EXTENDED-FLAG TLV Format

Type (16 bits): the value is TBD1 by IANA.

Length (16 bits): multiple of 4 octets.

O (Strict-Path): If set to 1, this indicates to the PCE that a path exclusively made of strict hops is required. Strict hop definition can be found in Section 4.1.

3.2. PATH-RECOMPUTATION TLV

This document defines new TLV for the LSPA Object for encoding information whether path recomputation is allowed for delegated LSP. The TLV is optional. If the TLV is included in LSPA object, the PCE MUST NOT recompute path in cases specified by flags in the TLV.
Type (16 bits): the value is TBD2 by IANA.

Length (16 bits): 4 octets

Reserved: MUST be set to zero by the sender and MUST be ignored by the receiver.

Flags: 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.

* P (Permanent): If set to 1, the PCE MUST NOT recompute path even if current path is not satisfying path computation constraints. Otherwise, if this flag is cleared, then the PCE MAY recompute path if original path is invalidated.

* F (Force): If set to 1, the PCE MUST NOT update path. If flag is cleared, the PCE MAY update path based on explicit request from operator.

4. Operation

4.1. Strict path enforcement

PCC MAY set the O flag in LSP-EXTENDED-FLAG TLV in PCRpt message to the PCE to indicate that a path exclusively made of strict hops is required.

O flag cleared or LSP-EXTENDED-FLAG TLV not included indicates that a loose path is acceptable.

In PCUpdate or PCInitiate messages, PCE MAY set O bit if strict path is provided.

The flag is applicable only for stateful messages. Existing O flag in RP object MAY be used to indicate similar behavior in PCReq and PCRep messages as described in as described in Section 7.4.1 of [RFC5440].
If O flag is set to 1 for both stateful and stateless messages for SR paths introduced in [RFC8664], PCE MUST use only SIDs, which will use explicitly specified adjacencies for packet forwarding. For example Adjacency SIDs MAY be used, but Prefix SIDs MUST NOT be used (even if there is only one adjacency). the PCE MUST use Adjacency SIDs only.

4.2. Path recomputation

PCC MAY set flags in PATH-RECOMPUTATION TLV to control path computation behavior on PCE side. If TLV is not included, then the PCE MAY use local policy to trigger path-computation or LSP path update.

The presence of the TLV is blocking path recomputation based on various triggers like topology update, any periodic update or changed state of other LSPs in the network. LSP path MAY be modified if forwarded packets will still use same path - for example if same path can be encoded using Adjacency and Prefix SIDs, then PCE MAY switch between various representations of same path.

If P flag is cleared, the PCE MAY recompute if current path is not considered valid, for example after topology update resulting in path not satisfying LSP’s path constraints, but it MUST NOT recompute path if current path is not optimal.

If P flag is set, the PCE MUST NOT recompute path during LSP lifetime even if path is invalidated. Only exception is explicit request from operator to recompute path.

If F flag is cleared, path update triggered manually by operator or any northbound interface of PCE MAY be done. If flag is set the PCE CAN update path only to tear down LSP by sending PCUpdate message with empty ERO.

TLV MAY be included in PCInitiate and PCUpdate messages to indicate, which triggers will be disabled on the PCE. PCC should reflect flag values in PCRpt messages to forward requirement to other PCEs in the network.

5. Security Considerations

No additional security measure is required.

6. IANA Considerations
6.1. LSP-EXTENDED-FLAG TLV

[I-D.ietf-pce-lsp-extended-flags] defines the LSP-EXTENDED-FLAG TLV. IANA is requested to make the following assignment from the "LSP-EXTENDED-FLAG TLV Flag Field" registry:

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>TBD1</td>
<td>Strict-Path Flag (O)</td>
<td>This document</td>
</tr>
</tbody>
</table>

Table 1

6.2. PATH-RECOMPUTATION TLV

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

<table>
<thead>
<tr>
<th>TLV Type</th>
<th>TLV Name</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>TBD2</td>
<td>PATH-RECOMPUTATION TLV</td>
<td>This document</td>
</tr>
</tbody>
</table>

Table 2

7. References

7.1. Normative References

[I-D.ietf-pce-lsp-extended-flags]

[I-D.peng-pce-entropy-label-position]
7.2. Informative References

[I-D.schmutzer-pce-cs-sr-policy]

[SDH]

[SONET]

Authors’ Addresses

Sidor, et al.
Expires 7 January 2023
PCE Extension for DetNet Bounded Latency

draft-xiong-pce-detnet-bounded-latency-00

Abstract

In certain networks, such as Deterministic Networking (DetNet), it is required to consider the bounded latency for path selection. This document describes the extensions to PCEP to carry bounded latency constraints and distribute deterministic paths for end-to-end path computation in DetNet service.

Status of This Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 25 August 2022.

Copyright Notice

Copyright (c) 2022 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.
1. Introduction

[RFC5440] describes the Path Computation Element Protocol (PCEP) which is used between a Path Computation Element (PCE) and a Path Computation Client (PCC) (or other PCE) to enable computation of Multi-protocol Label Switching (MPLS) for Traffic Engineering Label Switched Path (TE LSP). PCEP Extensions for the Stateful PCE Model [RFC8231] describes a set of extensions to PCEP to enable active control of MPLS-TE and Generalized MPLS (GMPLS) tunnels. As depicted in [RFC4655], a PCE MUST be able to compute the path of a TE LSP by operating on the TED and considering bandwidth and other constraints applicable to the TE LSP service request. The constraint parameters are provided such as metric, bandwidth, delay, affinity, etc. However these parameters can’t meet the DetNet requirements.
According to [RFC8655], Deterministic Networking (DetNet) operates at the IP layer and delivers service which provides extremely low data loss rates and bounded latency within a network domain. The bounded latency indicates the minimum and maximum end-to-end latency from source to destination and bounded jitter (packet delay variation). The computing method of end-to-end delay bounds is defined in [draft-ietf-detnet-bounded-latency]. It is the sum of the 6 delays in DetNet bounded latency model. And these delays should be measured and collected, but the related mechanisms are out of this document. The end-to-end delay bounds can also be computed as the sum of non queuing delay bound and queuing delay bound along the path. The upper bounds of non queuing delay are constant and depend on the specific network and the value of queuing delay bound depends on the queuing mechanisms deployed along the path.

As per [draft-ietf-detnet-controller-plane-framework], explicit path should be calculated and established in control plane to guarantee the deterministic transmission. When the PCE is deployed, the path computation should be applicable for DetNet networks. It is required that bounded latency including minimum and maximum end-to-end latency and bounded delay variation are considered during the deterministic path selection for PCE. The bounded latency constraints should be extended for PCEP. Moreover, the information along the deterministic path should be provided to the PCC after the path computation such as queuing parameters.

This document describes the extensions to PCEP to carry bounded latency constraints and distribute deterministic paths for end-to-end path computation in DetNet service.

1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

2. Terminology

The terminology is defined as [RFC8655] and [RFC5440].

3. PCEP Extensions

3.1. METRIC Object

The METRIC object is defined in Section 7.8 of [RFC5440], comprising metric-value and metric-type (T field), and a flags field, comprising a number of bit flags (B bit and C bit). This document defines two types for the METRIC object.
3.1.1. End-to-End Bounded Latency Metric

[RFC8233] has proposed the Path Delay metric type of the METRIC object to represent the sum of the Link Delay metric of all links along a P2P path. This document proposes the End-to-End Bounded Latency metric in PCEP to represent the sum of Output delay, Link delay, Frame preemption delay, Processing delay, Regulation delay and Queuing delay as defined in [draft-ietf-detnet-bounded-latency] along a deterministic path. Or the End-to-End Bounded Latency metric can be encoded as the sum of non queuing delay bound and queuing delay bound along the deterministic path. The extensions for End-to-End Bounded Latency Metric are as following shown:

* T=TBD1: End-to-End Bounded Latency Metric.

* The value of End-to-End Bounded Latency Metric is the encoding in units of microseconds with 32 bits.

* The B bit MUST be set to suggest a maximum bound for the end-to-end latency of deterministic path. The end-to-end latency must be less than or equal to the value.

A PCC MAY use the End-to-End Bounded Latency metric in a Path Computation Request (PCReq) message to request a deterministic path meeting the end-to-end latency requirement. A PCE MAY use the End-to-End Bounded Latency metric in a Path Computation Reply (PCRep) message along with a NO-PATH object in the case where the PCE cannot compute a path meeting this constraint. A PCE can also use this metric to send the computed end-to-end bounded latency to the PCC.

3.1.2. End-to-End Bounded Jitter Metric

RFC8233 has proposed the Path Delay Variation metric type of the METRIC object to represent the sum of the Link Delay Variation metric of all links along the path. This document proposes the End-to-end Bounded Jitter metric in PCEP to represent the difference between the end-to-end upper bounded latecny and the end-to-end lower bounded latecny along a deterministic path. The extensions for End-to-End Bounded Jitter Metric are as following shown:

* T=TBD2: End-to-End Bounded Jitter Metric.

* The value of End-to-End Bounded Jitter Metric is the encoding in units of microseconds with 32 bits.

* The B bit MUST be set to suggest a maximum bound for the end-to-end jitter of deterministic path. The end-to-end jitter must be less than or equal to the value.
A PCC MAY use the End-to-End Bounded Jitter metric in a PCReq message to request a deterministic path meeting the end-to-end delay variation requirement. A PCE MAY use the End-to-End Bounded Jitter metric in a PCRep message along with a NO-PATH object in the case where the PCE cannot compute a path meeting this constraint. A PCE can also use this metric to send the computed end-to-end bounded Jitter to the PCC.

3.2. LSP Object

The LSP Object is defined in Section 7.3 of [RFC8231]. This document defined a new flag (D-flag) to present the deterministic path for the LSP-EXTENDED-FLAG TLV carried in LSP Object as defined in [draft-ietf-pce-lsp-extended-flags].

D (Request for Deterministic Path) : If the bit is set to 1, it indicates that the PCC requests PCE to compute the deterministic path. A PCE would also set this bit to 1 to indicate that the deterministic path is included by PCE and encoded in the PCRep, PCUpd or PCCInitiate message.

3.3. ERO Object

The Explicit Route Object (ERO) is defined in RFC5440 to encode the path of a TE LSP through the network. SR-ERO subobject is used for SR-TE path which consists of one or more SIDs as defined in [RFC8664]. SRv6-ERO subobject is used for SRv6 path as defined in [draft-ietf-pce-segment-routing-ipv6]. This document defines deterministic path information for ERO, SR-ERO and SRv6-ERO subobjects.

3.3.1. Queue Information Structure

As defined in [draft-ietf-detnet-bounded-latency], the end-to-end delay bounds can be presented as the sum of non queuing delay bound and queuing delay bound along the path. The upper bounds of non queuing delay are constant and depend on the specific network, but the value of queuing delay bound depends on the queuing mechanisms deployed along the deterministic path. So to meet the requirements of the end-to-end delay, the PCE should select a queuing mechanism and configure the related parameters to the PCC. This document proposes the Queuing Information Structure carried in ERO or SR-ERO as shown in Figure 2.
Figure 1: Queuing Information Structure

Queuing Identifier (16bits): indicates the unique identifier of a queue for the node forwarding a DetNet flow.

Queuing Algorithm Type (16bits): indicates the type of queuing algorithm and each type represents the corresponding queuing mechanisms. The type can be defined refer to the queuing mechanisms which have been discussed such as [draft-ietf-detnet-bounded-latency]. More types can be defined due to the new queuing mechanisms.

Queuing Algorithm Type = 1: indicates the Time Aware Shaping [IEEE802.1Qbv].

Queuing Algorithm Type = 2: indicates the Credit-Based Shaper[IEEE802.1Q-2014] with Asynchronous Traffic Shaping[IEEE802.1Qcr].

Queuing Algorithm Type = 3: indicates the Guaranteed-Service IntServ [RFC2212].

Queuing Algorithm Type = 4: indicates the Cyclic Queuing and Forwarding [IEEE802.1Qch].

Queuing Algorithm Type = 5: indicates the Deadline Based Forwarding [draft-peng-detnet-deadline-based-forwarding].

Queuing Algorithm Type = 6: indicates the Multiple Cyclic Buffers Queuing Mechanism [draft-dang-queuing-with-multiple-cyclic-buffers].

Queuing Parameters Sub-TLV (variable): indicates the corresponding Queuing Parameters. The current Sub-TLVs including Deadline Sub-TLV and Cycle Sub-TLV are proposed as following sections.
3.3.1.1. Deadline Sub-TLV

Deadline Sub-TLV is optional for the Queuing Information Structure. The deadline-based queue mechanism has been proposed in [draft-stein-srtnsn] and [draft-peng-detnet-deadline-based-forwarding]. The deadlines along the path should be computed at PCE and configured to the PCC, and then inserted into the packet headers. When the Queuing Algorithm Type is set to indicate the deadline-based queuing mechanisms, the Deadline Sub-TLV should be used to carry the deadline parameters.

0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            Type               |          Length               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Deadline                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

Figure 2: Deadline Sub-TLV

Type (16bits): TBD3, indicates the type of Deadline Sub-TLV.

Length (16bits): indicated the length of Deadline Sub-TLV.

Deadline (32bits): indicates the deadline time for a node to forward a DetNet flow.

3.3.1.2. Cycle Sub-TLV

Cycle Sub-TLV is optional for the Queuing Information Structure. The cyclic-based queue mechanism has been proposed in [IEEE802.1Qch] and improved in [draft-dang-queuing-with-multiple-cyclic-buffers]. The cycle along the path should be computed at PCE and configured to the PCC, and then inserted into the packet headers. When the Queuing Algorithm Type is set to indicate the cycle-based queuing mechanisms, the Cycle Sub-TLV should be used to carry the cycle parameters.
Type (16bits): TBD4, indicates the type of Cycle Sub-TLV.

Length (16bits): indicated the length of Cycle Sub-TLV.

Cycle Profile ID (32bits): indicates the profile ID which the cyclic queue applied at a node.

Cycle ID (32bits): indicates the Cycle ID for a node to forward a DetNet flow.

4. Acknowledgements

TBA

5. IANA Considerations

TBA

6. Security Considerations

TBA

7. References

7.1. Normative References

[draft-ietf-pce-lsp-extended-flags]

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119,


Authors’ Addresses

Quan Xiong (editor)
ZTE Corporation
China
Email: xiong.quan@zte.com.cn

Peng Liu
China Mobile
Beijing
China
Email: liupengyjy@chinamobile.com
Abstract

This document proposes a set of extensions for PCEP to support the identifier of Network Resource Partition (NRP-ID) as the constraint of network slicing during path computation.

Status of This Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 7 January 2023.

Copyright Notice

Copyright (c) 2022 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.
1. Introduction

[RFC5440] describes the Path Computation Element Protocol (PCEP) which is used between a Path Computation Element (PCE) and a Path Computation Client (PCC) (or other PCE) to enable computation of Multi-protocol Label Switching (MPLS) for Traffic Engineering Label Switched Path (TE LSP). PCEP Extensions for the Stateful PCE Model [RFC8231] describes a set of extensions to PCEP to enable active control of MPLS-TE and Generalized MPLS (GMPLS) tunnels. As depicted in [RFC4655], a PCE MUST be able to compute the path of a TE LSP by operating on the TED and considering bandwidth and other constraints applicable to the TE LSP service request. The constraint parameters are provided such as metric, bandwidth, delay, affinity, etc. However these parameters can’t meet the network slicing requirements.
According to 5G context, network slicing is the collection of a set of technologies to support network service differentiation and meeting the diversified requirements from vertical industries. As defined in [I-D.ietf-teas-ietf-network-slice-definition], an IETF network slice is a logical network topology connecting a number of endpoints using a set of shared or dedicated network resources that are used to satisfy specific Service Level Objectives (SLOs). As defined in [I-D.ietf-teas-ietf-network-slices], a Network Resource Partition (NRP) is a collection of resources (bufferage, queuing, scheduling, etc.) in the underlay network to support the IETF Network Slice service (or any other service that needs logical network structures with required characteristics to be created). And as per [I-D.ietf-teas-ns-ip-mpls], NRP Identifier (NRP-ID) indicates an identifier that is globally unique within an NRP domain and that can be used in the control or management plane to identify the resources associated with the NRP. The NRP-ID could be used to identify the slice and network resource and viewed as constraints of network slicing when PCE is deployed. PCE MUST take the identifier of slicing into consideration during path computation.

This document proposes a set of extensions for PCEP to support the NRP-ID as the constraint of network slicing during path computation.

2. Conventions used in this document

2.1. Terminology

The terminology is defined as [RFC5440] and [I-D.ietf-teas-ietf-network-slice-definition].

2.2. 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.

3. PCEP Extensions for Network Resource Partition

As defined in [RFC5440], the LSPA object is used to specify the LSP attributes to be taken into account by the PCE during path computation such as constraints. This document proposes new TLV for the LSPA object to carry TE constraints for network slicing.
3.1. NRP-ID TLV

The NRP-ID TLV is optional and is defined to carry the slice specific constraint. PCEP message needs to carry NRP-ID to limit the network resources for path calculation within a NRP domain.

The format of the NRP-ID TLV is shown as Figure 1:

```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Type=TBD1             |            Length=4           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           NRP-ID                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 1: NRP-ID TLV

The code point for the TLV type is TBD1. The TLV length is 4 octets.

NRP-ID (32 bits): indicates a NRP Identifier as defined in [I-D.ietf-teas-ns-ip-mpls].

3.2. Protocol Operation

As defined in [I-D.ietf-teas-ns-ip-mpls], NRP state aware TE (NRP-TE) should implement the TE path selection that takes into account the available network resources associated with a specific NRP. The NRP-ID TLV should be carried in PCEP messages when computing NRP state aware TE paths. The PCE may maintain network resources per path and the NRP state within the resource pool identified by NRP-ID.

In a PCReq message, a PCC MAY request the PCC to compute the NRP-TE path and insert a NRP-ID TLV to indicate the resources within a NRP domain. The PCE will perform path computation based on the intra-domain or inter-domain sub-topology identified by the specific NRP-ID, that can be used to find the corresponding customized topology or referenced topology, and corresponding resources. The PCE may reply the PCC with NRP-ID TLV carried in PCRep message and the headend may insert the NRP-ID into an encapsulated data packet. In case of unsuccessful path computation when the NRP-ID constraint could not be satisfied, the the PCRep message may contain a NO-PATH object.
In a PCInit/PCUpd message, the PCE MAY compute the optimal NRP-TE path and carry the NRP-ID TLV so as to provide the network slicing information. If a PCC is unable to recognize a NRP-ID value passed in an LSP PCInit/PCUpd request, the PCC must keep the LSP in DOWN state, and include an LSP Error Code value of "Unsupported NRP" [Value to be assigned by IANA] in LSP State Report message.

4. Security Considerations

TBA

5. Acknowledgements

TBA

6. IANA Considerations

IANA is requested to make allocations from the registry, as follows:

<table>
<thead>
<tr>
<th>Type</th>
<th>TLV</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>TBD1</td>
<td>NRP-ID TLV</td>
<td>[this document]</td>
</tr>
</tbody>
</table>

Table 1

7. Normative References

[I-D.ietf-teas-ietf-network-slice-definition]

[I-D.ietf-teas-ietf-network-slices]

[I-D.ietf-teas-ns-ip-mpls]


Authors’ Addresses

Quan Xiong
ZTE Corporation
No.6 Huashi Park Rd
Wuhan
Hubei, 430223
China
Email: xiong.quan@zte.com.cn

Shaofu Peng
ZTE Corporation
No.50 Software Avenue
Nanjing
Jiangsu, 210012
China

Path Computation Element Communication Protocol (PCEP) Extensions to Redundancy Policy
draft-yang-pce-pcep-redundancy-policy-00

Abstract

PCEP is used to provide a communication between a PCC and a PCE. This document defines the extensions to PCEP to support the redundancy paths computation. Specifically, two new TLVs are defined to support the request of redundancy path computation and protection method, and one TLV is defined to distribute the Candidate Path Flag of an SR Policy.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in .

Status of This Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 12 January 2023.

Copyright Notice

Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.
1. Introduction

Redundancy protection [I-D.ietf-spring-sr-redundancy-protection] is a generalized protection mechanism by replicating and transmitting copies of flow packets on the redundancy node over multiple different and disjoint paths, and further eliminating the redundant packets at the merging node. To support redundancy protection in Segment Routing, Redundancy Policy[I-D.geng-spring-redundancy-policy] is provided to instantiate the segment lists of more than one disjoint forwarding paths. This document extends the PCEP protocols to support the request of redundancy paths computation and protection method, and further distribute the flag of redundancy policy to instantiate more than one segment lists for redundancy forwarding.
2. Terminology

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

3. RP Object

The RP (Request Parameters) object defined in [RFC5440] is used to specify various characteristics of the path computation request and MUST be carried within each PCReq and PCRep messages. The format of RP object is as follows:

```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Flags                    |O|B|R| Pri |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Request-ID-number                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
//                      Optional TLVs                          //
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Request Parameters Object

3.1. Redundancy Protection TLV

In order to request PCE to compute multiple redundancy forwarding paths with the intention of redundancy protection, this document defines a new TLV named Redundancy Protection TLV. The format of Redundancy Protection TLV is shown as follows.

```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Type = TBD1          |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Flag       |     Number    |            Reserved           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Redundancy Protection TLV
Where:

* Type: to be assigned by IANA.
* Length: 16-bit value to indicate the length of the value portion in bytes.
* Flag: 8-bit bitmap to indicate the redundancy constraint of path computation that PCC requires.

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

where:

a) R-Flag: One bit Redundancy Flag is used to indicate whether PCC requires the common path computation or a redundancy path computation. When redundancy flag bit is set to 0, it means PCC requests a common path computation. When redundancy flag bit is set to 1, it means PCC requests a redundancy path computation.

b) U-Flag: Unused and undefined

* Number: 8-bit value to indicate how many redundancy forwarding paths that PCC requires. The range of the number is recommended from 2 to 8.
* Reserved: 16-bit of reserved bits. SHOULD be set to zero on transmission and MUST be ignored on receipt.

When PCC requests a redundancy path computation, it MUST include the Redundancy Flag TLV in the RP object in PCReq message. When PCC includes the Redundancy Flag TLV in a path computation request, PCE would reply with the required number of redundancy forwarding paths and the set of Redundancy Flag associated with the computed paths.

3.2. Protection Type TLV

As specified in [I-D.geng-spring-redundancy-policy], multiple candidate paths can co-exist with different types of protection. In order to differentiate the types of protection, a new TLV named Protection Type TLV is defined. The format of Protection Type TLV is shown as follows.
Protection Type TLV

where:

* Type: to be assigned by IANA.

* Length: 16-bit value to indicate the length of the value portion in bytes.

* Protection: 4-bit value to indicate the protection type of path computation that PCC requires. The following Table gives the values and corresponding protection types.

```
+--------------+-------------------------+
<table>
<thead>
<tr>
<th>Value</th>
<th>Protection Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>No protection</td>
</tr>
<tr>
<td>1</td>
<td>Backup Protection</td>
</tr>
<tr>
<td>2</td>
<td>Redundancy Protection</td>
</tr>
<tr>
<td>3-15</td>
<td>Undefined</td>
</tr>
</tbody>
</table>
```

Protection Type Values

* Reserved: 24-bit of reserved bits. SHOULD be set to zero on transmission and MUST be ignored on receipt.

4. PCEP Extensions for Redundancy Policy

As per [I-D.ietf-pce-segment-routing-policy-cp], the mapping between PCEP Associations and SR Policies is always one-to-one, and the mapping between PCEP Tunnels and SR Policy Candidate Paths may be either one-to-one or many-to-one. Regarding Redundancy Policy, the mapping between PCEP Associations and Redundancy Policy is always one-to-one. PCEP Tunnels and Redundancy Policy Candidate Paths are always many-to-one. The definitions of SR Policy Association Type
This document introduces a new SR Policy Candidate Path Attribute called Flag, which identify the Flag of SR Policy Candidate Path within the context of an SR Policy. This Flag identifier MUST NOT change for a given LSP during its lifetime. When these rules are not satisfied, the PCE MUST send a PCErr message with Error-Type = 26 "Association Error", Error Value = TBD4 "SR Policy Candidate Path Flag Mismatch".

4.1. SR Policy Candidate Path Flag TLV

A new SR Policy Association Type TLV [I-D.ietf-pce-segment-routing-policy-cp] called SR Policy Candidate Path Flag TLV is defined to indicate the Flag of a candidate path. The format of SR Policy Candidate Path Flag TLV is shown in following.

\[
\begin{array}{cccc}
0 & 1 & 2 & 3 \\
\hline
0 & 1 & 2 & 3 \\
\hline
| Flag | Length |
\end{array}
\]

where:
SRPOLICY-CPPATH-Flag TLV

* Type: to be allocated by IANA.

* Length: specifies the length of the value field not including Type and Length fields.

* Flag: 8-bit bitmap of Flag. A new registry "SR Policy Candidate Path Flags" is created. One flag is defined at this writing:

\[
\begin{array}{cccc}
0 & 1 & 2 & 3 \\
\hline
0 & 1 & 2 & 3 \\
\hline
|R|U|U|U|U|U|U|U|
\end{array}
\]
where:

a) R-Flag: One bit Redundancy Flag is used to indicate the type of candidate path. When R Flag is set, it represents the candidate path is used for the redundancy forwarding.

b) U-Flag: Unused and undefined

* RESERVED: 3-octet of reserved bits. SHOULD be set to zero on transmission and MUST be ignored on receipt.

4.2. Path Binding TLV

Since Redundancy Policy can be optionally associated with the Binding Segment, specifically the Redundancy Segment, according to [I-D.ietf-pce-segment-routing-policy-cp], the functionality of specified-BSID-only is not mandatory to be enabled. It means that the given Redundancy Segment is not required to be allocated and programmed for the LSP to be operationally up. When there is a Redundancy Segment associated with Redundancy Policy, TE-PATH-BINDING TLV [I-D.ietf-pce-binding-label-sid] is used to distribute Redundancy Segment as the Binding Segment of Redundancy Policy.

5. IANA Considerations

5.1. New TLV Type

This document defines three new TLVs.

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>TBD1</td>
<td>Redundancy Protection TLV</td>
<td>This document</td>
</tr>
<tr>
<td>TBD2</td>
<td>Protection Type TLV</td>
<td>This document</td>
</tr>
<tr>
<td>TBD3</td>
<td>SR Policy Candidate Path Flag TLV</td>
<td>This document</td>
</tr>
</tbody>
</table>
5.2. PCEP Errors

This document defines one new Error-Value within the "Association Error" Error-Type. IANA is requested to allocate new error values within the "PCEP-ERROR Object Error Types and Values" sub-registry of the PCEP Numbers registry, as follows:

<table>
<thead>
<tr>
<th>Error-Type</th>
<th>Meaning</th>
<th>Error-value</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>26</td>
<td>Association Error</td>
<td>TBD4: SR Policy Candidate Path Flag Mismatch</td>
<td>This I-D</td>
</tr>
</tbody>
</table>

6. Security Considerations

TBD

7. References

7.1. Normative References

[I-D.geng-spring-redundancy-policy]

[I-D.ietf-pce-segment-routing-policy-cp]
7.2. Informative References

[I-D.ietf-pce-binding-label-sid]

Authors’ Addresses

Fan Yang
Huawei
156 Beiqing Rd.
Beijing
100095
China
Email: shirley.yangfan@huawei.com
Abstract

PCEP is used to provide a communication between a PCC and a PCE. This document defines the extensions to PCEP to support the bounded-latency path computation. Specifically, two new objects and three new TLVs are defined for the transmission of bounded latency information between PCC and PCE to guarantee the bounded latency transmission in control plane.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

Status of This Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 12 January 2023.

Copyright Notice

Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.
Table of Contents

1. Introduction ................................................. 2
2. Terminology ................................................ 3
3. Object Formats ............................................. 3
   3.1. Open Object ........................................... 3
       3.1.1. Bounded Latency Capability TLV .................... 3
   3.2. RP Object ............................................. 6
       3.2.1. BLI Type TLV ..................................... 6
   3.3. Traffic Model Object .................................. 8
   3.4. BLI Object ............................................ 9
       3.4.1. BLI List TLV ..................................... 10
       3.4.2. Shared BLI TLV .................................. 11
4. SR Policy for BLI ........................................... 11
5. IANA Considerations ......................................... 12
   5.1. New TLV Type .......................................... 12
   5.2. New Object ............................................ 12
6. Security Considerations ...................................... 12
7. Acknowledgements ........................................... 12
8. Normative References ....................................... 12
Authors’ Addresses ........................................... 14

1. Introduction

[RFC8665] provides the overall architecture for Deterministic Networking (DetNet), which provides the capability to carry specified unicast or multicast data flows with extremely low data loss rates and bounded end-to-end latency within a network domain. Based on this, [draft-finn-detnet-bounded-latency] proposed a timing model for sources, destinations, and DetNet transit nodes. Using the model, it provides a methodology to compute end-to-end latency and backlog bounds for various queuing methods.

[RFC5440] describes the Path Computation Element Protocol (PCEP) for communications between a Path Computation Client (PCC) and a Path Computation Element (PCE), or between two PCEs. PCEP defines the interaction and data format of path calculation requests and path computation replies between PCC and PCE. [RFC8231] specifies extension to PCEP to enable stateful control of LSPs within and across PCEP.
sessions in compliance with [RFC4657]. [I-D.yzz-detnet-enhanced-data-plane] enhances the DetNet data plane by introducing Bounded Latency Information (BLI) which facilitates DetNet transit nodes to guarantee the bounded latency transmission in data plane. Based on that, [I-D.geng-spring-sr-enhanced-detnet] defines how to leverage Segment Routing (SR) and Segment Routing over IPv6 (SRv6) to implement bounded latency.

When a PCE is used to compute paths using PCEP, it is important that the PCE understands the bounded latency requirement and the head end of the path also need to understands the bounded latency information associated with the candidate path.

This document defines the extensions to PCEP to support the bounded-latency path computation. Specifically, two new objects and three new TLVs are defined for the transmission of bounded latency information between PCC and PCE to guarantee the bounded latency transmission in control plane.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

3. Object Formats

3.1. Open Object

[RFC5440] defines the Open object in open message used to specify the PCEP version, Keepalive frequency, DeadTimer, PCEP session ID, and other communication parameters. The Open object may also contain a set of TLVs used to convey various session characteristics.

3.1.1. Bounded Latency Capability TLV

During the PCEP initialization phase, PCEP speakers SHOULD advertise their support of Bounded Latency features, for this reason this document defines the Bounded Latency capability TLV.

A PCEP speaker includes the Bounded Latency capability TLV in the Open object to advertise its support for Bounded Latency features. The format of the Bounded Latency capability TLV is formatted as follows:
Type: To be assigned by IANA.

Length: 16 bits value to indicate the length of the value portion in bytes.

Type-Flag: 16 bits of flags to indicate which kind of BLI Type the speaker supports. A new registry "Bounded Latency Type Flags" is expected to be created. Table 1 shows the assignment of Bounded Latency Type Flags. The speaker sets the defined bit in flag to indicate that it supports this Type of BLI. The undefined bits MUST be set to zero by the sender and MUST be ignored by the receiver.

Format-Flag: 16 bits of flags to indicate which kind of BLI Format the speaker supports. A new registry "Bounded Latency Format Flags" is expected to be created. Table 2 shows the assignment of Bounded Latency Format Flags. The speaker sets the defined bit in flag to indicate that it supports this Format of BLI. The undefined bits MUST be set to zero by the sender and MUST be ignored by the receiver.
<table>
<thead>
<tr>
<th>Bit</th>
<th>BLI Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Reserved</td>
</tr>
<tr>
<td>1</td>
<td>Time resource ID</td>
</tr>
<tr>
<td>2</td>
<td>Priority</td>
</tr>
<tr>
<td>3</td>
<td>End-to-end delay budget</td>
</tr>
<tr>
<td>4</td>
<td>Local delay budget</td>
</tr>
<tr>
<td>5</td>
<td>Reserved</td>
</tr>
<tr>
<td>6</td>
<td>Reserved</td>
</tr>
<tr>
<td>7</td>
<td>End-to-end delay variation budget</td>
</tr>
<tr>
<td>8</td>
<td>Local delay variation budget</td>
</tr>
<tr>
<td>9-15</td>
<td>undefined</td>
</tr>
</tbody>
</table>

Table 1: Bounded Latency Type flag and the corresponding BLI type

<table>
<thead>
<tr>
<th>Bit</th>
<th>BLI Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Reserved</td>
</tr>
<tr>
<td>1</td>
<td>32-bit unsigned Integer</td>
</tr>
<tr>
<td>2</td>
<td>16-bit unsigned Integer</td>
</tr>
<tr>
<td>3</td>
<td>8-bit unsigned Integer</td>
</tr>
<tr>
<td>4-15</td>
<td>undefined</td>
</tr>
</tbody>
</table>

Table 2: Bounded Latency Format flag and the corresponding BLI format
3.2. RP Object

The RP (Request Parameters) object is defined in [RFC5440], used to specify various characteristics of the path computation request and MUST be carried within each PCReq and PCRep messages. The format of RP object is as follows:

```
+---------+---------+---------+---------+---------+---------+---------+---------+
| Flags   | O | B | R | Pri |
+---------+---------+---------+---------+---------+---------+---------+---------+
| Request-ID-number |
+---------+---------+---------+---------+---------+---------+---------+---------+
// Optional TLVs //
+---------+---------+---------+---------+---------+---------+---------+---------+
```

The detail information about the fields in the RP object is defined in section 7.4 of [RFC5440].

3.2.1. BLI Type TLV

In order to specify the type and format of the BLI associated with candidate path, this document defines a new TLV named BLI type TLV. The BLI type TLV is formatted as follow:

```
+---------+---------+---------+---------+---------+---------+---------+---------+
| Type = TBD2 | Length=4 |
+---------+---------+---------+---------+---------+---------+---------+---------+
| BLI Type | BLI Format | Reserved |
+---------+---------+---------+---------+---------+---------+---------+---------+
```

Where:

Type: to be assigned by IANA.

Length: 16 bits value to indicate the length of the value portion in bytes. The value of this field is 4.

BLI Type: 8 bits value to indicate the type of BLI that PCC desires. Table 3 shows the values and their corresponding BLI types.
<table>
<thead>
<tr>
<th>BLI Type Value</th>
<th>Bounded Latency Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Reserved</td>
</tr>
<tr>
<td>1</td>
<td>Time resource ID</td>
</tr>
<tr>
<td>2</td>
<td>Priority</td>
</tr>
<tr>
<td>3</td>
<td>End-to-end delay variation budget</td>
</tr>
<tr>
<td>4</td>
<td>Local delay budget</td>
</tr>
<tr>
<td>5</td>
<td>End-to-end queue delay budget</td>
</tr>
<tr>
<td>6</td>
<td>Local queue delay budget</td>
</tr>
</tbody>
</table>

Table 3: BLI Type value and their corresponding types

BLI Format: 8 bits value to indicate the format of BLI that PCC desires. Table 4 shows the values and their corresponding BLI formats.

<table>
<thead>
<tr>
<th>Format Value</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>32-bit unsigned Integer</td>
</tr>
<tr>
<td>2</td>
<td>16-bit unsigned Integer</td>
</tr>
<tr>
<td>3</td>
<td>8-bit unsigned Integer</td>
</tr>
</tbody>
</table>

Table 4: BLI Format and their corresponding formats

When PCC needs to request a bounded-latency path, it MUST include the BLI Type TLV in the RP object in PCReq message. If a PCC includes an BLI Type TLV on a path calculation request, then the PCE will reply the specific type of BLI associated with computed path.
3.3. Traffic Model Object

The Traffic Model Object is optional in the PCReq message and used to specify the traffic model for the bounded-latency path computation. The traffic model object contains a set of fields used to specify the traffic features. [RFC9016] defines the traffic specification of the DetNet flow, which includes a set of attributes to specify how the DetNet Ingress transmits packets for the DetNet flow. Based on that, this document proposes the Traffic Model Object to describe the DetNet flow for bounded-latency path computation.

Traffic Model Object-Class is TBD3;
Traffic Model Object-Type is 1.

The format of the Traffic Model Object is shown in below:

```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Traffic ID          |            Flags              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     MinPacketsPerInterval     |     MaxPacketsPerInterval     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         MinPayloadSize        |       MaxPayloadSize          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           Interval                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         MinBandwidth                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          MaxLatency                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      MaxLatencyVariation                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
//                       Optional TLVs                         //
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Where:

Traffic ID: The only identification of the specify traffic in PCC. When the PCC need to request a path computation for a traffic, it MUST assign a 16-bit traffic identifier to specify the traffic in local.
Flags: 16 bits of flags. A new registry "Traffic Model Flags" is expected to be created. At the writing time, all flags are unused and undefined.

MinPacketsPerInterval: the minimum number of packets that the Ingress will transmit in one Interval.

MaxPacketsPerInterval: the maximum number of packets that the Ingress will transmit in one Interval.

MinPayloadSize: the minimum payload size that the Ingress will transmit.

MaxPayloadSize: the maximum payload size that the Ingress will transmit.

Interval: the period of time in which the traffic specification is specified.

MinBandwith: the minimum bandwidth that has to be guaranteed for the DetNet traffic.

MaxLatency: the end-to-end maximum latency for a single packet of the DetNet traffic.

MaxLatencyVariation: the difference between the minimum and the maximum end-to-end, one-way latency.

The Traffic Model object body has a variable length and may contain TLVs for the additional attributes of the traffic model. At the writing time there is no TLV defined for Traffic Mode Object.

3.4. BLI Object

In order to support the bounded-latency path computation, a new kind of object named BLI object is defined in this document to indicate the bounded latency information of a candidate path.

The BLI object is optionally carried within a PCRep message so as to indicate the requirement and resource allocation for the candidate path. When a PCC request a bounded-latency path computation and the PCE find out a path satisfying the set of constraints, the PCE MUST include the BLI object in PCRep message.

BLI Object-Class is TBD4.

BLI Object-Type is 1.
The format of BLI object is as follows:

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

The BLI object body has a variable length and may contain TLVs for the different kinds of BLI. This document defines two kinds of BLI TLV for different scenarios.

### 3.4.1. BLI List TLV

When all of the nodes in the Explicit Route Object (ERO) [RFC5440] request different BLI to guarantee bounded latency, a BLI list TLV is defined.

The BLI list sub-TLV is formatted as follows.

```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            Type=TBD5          |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        BLI List [m]                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             ...                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        BLI List [1]                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Where:

- **Type**: to be assigned by IANA.
- **Length**: 16 bits length value to indicate the length of BLI list in octet.
- **BLI List [1... m]**: 32 bits length bounded latency information, representing the nth BLI in the BLI list.

The BLI in the BLI List corresponds to the node in the ERO object one by one. The length of the BLI List depends on the number of nodes in the ERO object.
3.4.2. Shared BLI TLV

When all of the nodes in the ERO indicated by the sub-object list request BLI to guarantee bounded latency with the same BLI value, the Shared BLI TLV is defined.

The Shared BLI TLV is defined as follows:

```
+-----------------+-----------------+-----------------+-----------------+
| Type=TBD6       | Length          |
+-----------------+-----------------+
| BLI             |
+-----------------+-----------------+
```

Where:

Type: to be assigned by IANA.

Length: 16 bits value to indicate the length of BLI in octet.

BLI: 32 bits value of Bounded Latency Information to guarantee the bounded latency.

4. SR Policy for BLI

[I-D.ietf-pce-segment-routing-policy-cp] proposes extension to PCEP to support association among candidate paths of a given SR policy. For the bounded latency path, the additional bounded latency information associated with the candidate path SHOULD be carried with SR Policy. Therefore, the additional BLI TLV SHOULD be defined to indicate the bounded-latency requirement and resources allocation for the nodes along the candidate path. For different scenario, different BLI TLV need to be carried by SR policy.

When all of the nodes/adjacencies in the explicit path indicated by the segment list request different BLI to guarantee bounded latency, a BLI list TLV is need to be carried by SR Policy. The BLI list TLV is defined in section 3.4.1.

When all of the nodes/adjacencies in the explicit path indicated by the segment list request BLI to guarantee bounded latency with the same BLI value, a Per-segment BLI TLV is need to be carried by SR Policy. The Per-segment BLI TLV is defined in section 3.4.2.
5. IANA Considerations

This document defines four new TLVs and two new Object.

5.1. New TLV Type

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>TBD1</td>
<td>Bounded-Latency Capability TLV</td>
<td>This document</td>
</tr>
<tr>
<td>TBD2</td>
<td>BLI Type TLV</td>
<td>This document</td>
</tr>
<tr>
<td>TBD5</td>
<td>BLI list TLV</td>
<td>This document</td>
</tr>
<tr>
<td>TBD6</td>
<td>Shared BLI TLV</td>
<td>This document</td>
</tr>
</tbody>
</table>

5.2. New Object

IANA is requested to make the assignment from the "PCEP Object" sub-registry as follows:

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>TBD3</td>
<td>Traffic Model Object</td>
<td>This document</td>
</tr>
<tr>
<td>TBD4</td>
<td>BLI Object</td>
<td>This document</td>
</tr>
</tbody>
</table>

6. Security Considerations

TBD

7. Acknowledgements

8. Normative References

[I-D.geng-spring-sr-enhanced-detnet]

[I-D.ietf-pce-segment-routing-policy-cp]

[I-D.yzz-detnet-enhanced-data-plane]


Authors’ Addresses

Li Zhang
Huawei
Email: zhangli344@huawei.com

Xuesong Geng
Huawei
Email: gengxuesong@huawei.com

Tianran Zhou
Huawei
Email: zhoutianran@huawei.com