                    PCEP 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.

Table of Contents

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 ccollected, 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 transimission. 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 constriants should be extended for PCEP. Moreover, the information along the deterministic path should be provided to the PCC after the path conputation 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
defiend 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 PCInitiate 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.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        Queuing Identifier      |    Queuing Algorithm Type     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            Queuing Parameters Sub-TLV (variable)              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
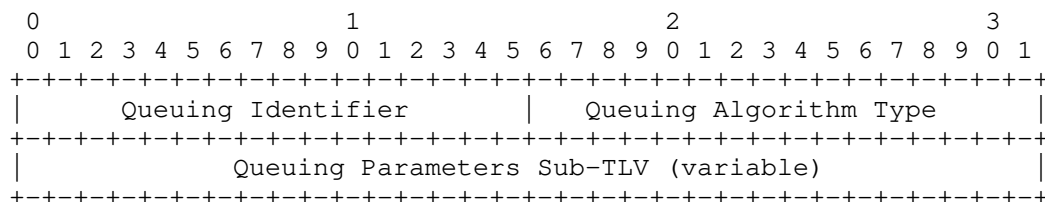
Figure 1: Queuing Information Structure

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
[IIEEE802.1Qbv].

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): indicuates 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-
   srtsn] 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                           |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
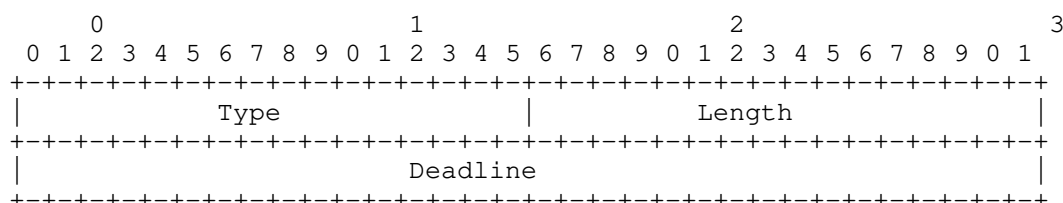
                       Figure 2: Deadline Sub-TLV

   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
   clycle 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.

```
       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             |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                        Cycle Profile ID                       |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                           Cycle ID                            |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
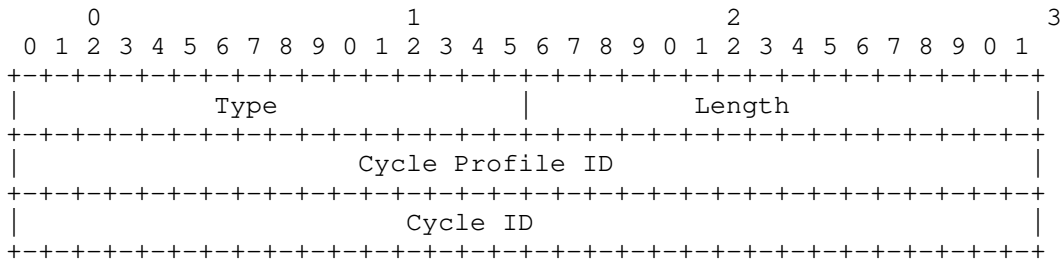
Figure 3: Cycle Sub-TLV

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]
              "LSP Extended Flags", July 2021, <https://www.rfc-
              editor.org/info/draft-ietf-pce-lsp-extended-flags>.

   [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate
              Requirement Levels", BCP 14, RFC 2119,
              DOI 10.17487/RFC2119, March 1997,
              <https://www.rfc-editor.org/info/rfc2119>.

   [RFC4655] "A Path Computation Element (PCE)-Based Architecture",
              August 2006, <https://www.rfc-editor.org/info/RFC4655>.

   [RFC4915] "Multi-Topology (MT) Routing in OSPF", June 2007,
              <https://www.rfc-editor.org/info/RFC4915>.

   [RFC5120] "M-ISIS: Multi Topology (MT) Routing in Intermediate
              System to Intermediate Systems (IS-ISs)", February 2008,
              <https://www.rfc-editor.org/info/RFC5120>.

   [RFC5440] "Path Computation Element (PCE) Communication Protocol
              (PCEP)", March 2009,
              <https://www.rfc-editor.org/info/RFC5440>.

   [RFC6549] "OSPFv2 Multi-Instance Extensions", March 2012,
              <https://www.rfc-editor.org/info/RFC6549>.

   [RFC7752] "North-Bound Distribution of Link-State and Traffic
              Engineering (TE) Information Using BGP", March 2016,
              <https://www.rfc-editor.org/info/RFC7752>.

   [RFC8174] "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key
              Words", May 2017,
              <https://www.rfc-editor.org/info/RFC8174>.

   [RFC8231] "Path Computation Element Communication Protocol (PCEP)
              Extensions for Stateful PCE", September 2017,
              <https://www.rfc-editor.org/info/RFC8231>.

   [RFC8655] "DetNet Architecture", June 2017,
              <https://www.rfc-editor.org/info/RFC8655>.

   [RFC8664] "SR-PCE", August 2020,
              <https://www.rfc-editor.org/info/RFC8664>.

Authors' Addresses

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

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