Stream: Internet Engineering Task Force (IETF)
RFC: 8800
Category: Standards Track
Published: July 2020
ISSN: 2070-1721

Authors:
S. Litkowski, *Cisco Systems, Inc.*
S. Sivabalan, *Ciena Corporation*
C. Barth, *Juniper Networks*
M. Negi, *RtBrick India*

RFC 8800
Path Computation Element Communication Protocol (PCEP) Extension for Label Switched Path (LSP) Diversity Constraint Signaling

Abstract

This document introduces a simple mechanism to associate a group of Label Switched Paths (LSPs) via an extension to the Path Computation Element Communication Protocol (PCEP) with the purpose of computing diverse (disjointed) paths for those LSPs. The proposed extension allows a Path Computation Client (PCC) to advertise to a Path Computation Element (PCE) that a particular LSP belongs to a particular Disjoint Association Group; thus, the PCE knows that the LSPs in the same group need to be disjoint from each other.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

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

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

[RFC8697] introduces a generic mechanism to create a grouping of LSPs in the context of a PCE that can then be used to define associations between a set of LSPs and a set of attributes (such as configuration parameters or behaviors) and is equally applicable to the active and passive modes of a stateful PCE [RFC8231] or a stateless PCE [RFC4655].

This document specifies a PCEP extension to signal that a set of LSPs in a particular group should use diverse (disjointed) paths, including the requested type of diversity. Sections 3 and 4 describe the property and use of a Disjoint Association Group. A PCC can use this extension to signal to a PCE that a particular LSP belongs to a particular Disjoint Association Group. When a PCE receives LSP states belonging to the same Disjoint Association Group from some PCCs, the PCE should ensure that the LSPs within the group are disjoint from each other.

1.1. Requirements Language

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

2. Terminology

The following terminology is used in this document.

DAT: Disjoint Association Type

DAG: Disjoint Association Group

MPLS: Multiprotocol Label Switching

OF: Objective Function

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 Communication Protocol

PLSP-ID: PCEP-specific identifier for the LSP

SRLG: Shared Risk Link Group

3. Motivation

Path diversity is a very common use case in today's IP/MPLS networks, especially for layer 2 transport over MPLS. A customer may request that the operator provide two end-to-end disjoint paths across the operator's IP/MPLS core. The customer may use these paths as primary/backup or active/active configuration.

Different levels of disjointness may be offered:

- Link disjointness: the paths of the associated LSPs should transit different links (but may use common nodes or different links that may have some shared fate).
- Node disjointness: the paths of the associated LSPs should transit different nodes (but may use different links that may have some shared fate).
- SRLG disjointness: the paths of the associated LSPs should transit different links that do not share fate (but may use common transit nodes).
- Node+SRLG disjointness: the paths of the associated LSPs should transit different links that do not have any common shared fate and should transit different nodes.

The associated LSPs may originate from the same or different head end(s) and may terminate at the same or different tail end(s).

4. Applicability

```
         -----------------------------------------
        /                                         \
       /        +------+                           \
      |         | PCE  |                            |
      |         +------+                            |
      |                                             |
      |          ***********************>           |
      | +------+           10             +------+  |
CE1 ****| PE 1 | ----- R1 ---- R2 ------- | PE 2 |**** CE2
      | +------+       |        |         +------+  |
      |                |        |                   |
      |                |        |                   |
      | +------+       |        |         +------+  |
CE3 ****| PE 3 | ----- R3 ---- R4 ------- | PE 4 |**** CE4
      | +------+ ***********************> +------+  |
      |                                             |
       \                                           /
        \_________________________________________/
```

Figure 1: Disjoint Paths with Different Head Ends and Tail Ends

In the figure above, let us consider that the customer wants to have two disjoint paths, one between CE1 and CE2 and one between CE3 and CE4. From an IP/MPLS network point view, in this example, the CEs are connected to different PEs to maximize their disjointness. When LSPs originate from different head ends, distributed computation of diverse paths can be difficult, whereas computation via a centralized PCE ensures path disjointness, correctness, and simplicity.

Section 5.4 describes the relationship between the Disjoint Association Group (DAG) and Synchronization VECtor (SVEC) object.

The PCEP extension for stateful PCE [RFC8231] defined new PCEP messages -- Path Computation Report (PCRpt), Path Computation Update (PCUpd), and Path Computation Initiate (PCInitiate) [RFC8281]. These messages use a PLSP-ID in the LSP object for identification. Moreover, to allow diversity between LSPs originating from different PCCs, the generic mechanism to create a grouping of LSPs that is equally applicable to the active and passive modes of a stateful PCE is described in [RFC8697].

Using the extension to PCEP defined in this document, the PCC uses the extension defined in [RFC8697] to indicate that a group of LSPs are required to be disjoint; such indication should include disjointness parameters like the type of disjointness, the Disjoint Association Group identifiers, and any customization parameters according to the configured local policy.

The management of the Disjoint Association Group IDs will be a key point for the operator as the Association ID field is limited to 65535. The local configuration of the IPv4/IPv6 Association Source, or Global Association Source/Extended Association ID, can overcome this limitation, as described in [RFC8697]. When a PCC or PCE initiates all the LSPs in a particular Disjoint Association Group, it can set the IPv4/IPv6 Association Source as one of its own IP address. When disjoint LSPs are initiated from different head ends, the Association Source could be the PCE address or any other unique value to identify the DAG.

```
        Initiate Disjoint LSPs
                 |
                 |                       PCReq/PCRpt
                 V                        {DAG Y}
              +-----+                ----------------> +-----+
   _ _ _ _ _ _| PCE |               |                  | PCE |
  |           +-----+               |      ----------> +-----+
  | PCInitiate                      |     |    PCReq/PCRpt
  |{DAG X}                          |     | {DAG Y}
  |                                 |     |
  |              .-----.            |     |         .-----.
  |             (       )           | +-----+      (       )
  |         .--(         )--.       | |PCC 2|--.--(         )--.
  V        (                 )      | +-----+ (                 )
+---+     (                  )      |        (                  )
|PCC|----(   (G)MPLS network )   +-----+    ( (G)MPLS network   )
+---+    (                   )   |PCC 1|-----(                  )
{DAG X}   (                 )    +-----+      (                )
           '--(         )--'                   (           )--'
               (       )                         (        )
                '-----'                            '-----'

Case 1: Disjointness initiated by   Case 2: Disjointness initiated by
    PCE and enforced by PCC              PCC and enforced by PCE
```

Figure 2: Sample Use Cases for Carrying Disjoint Association Group over PCEP Session

The Disjoint Association Group within a PCEP messages is used for:

- Configuration: Used to communicate the configured disjoint requirement to a PCEP peer.
- Status: Used to communicate the status of the computed disjointness.

5. Protocol Extension

5.1. Association Group

As per [RFC8697], LSPs are associated with other LSPs with which they interact by adding them to a common association group. As described in [RFC8697], the association group is uniquely identified by the combination of the following fields in the ASSOCIATION object: Association Type, Association ID, Association Source, and (if present) Global Association Source or Extended Association ID.

This document defines a new Association type, called "Disjoint Association" (2), based on the generic ASSOCIATION object. This new Association type is also called "DAT", for "Disjoint Association Type".

[RFC8697] specifies the mechanism for the capability advertisement of the Association types supported by a PCEP speaker by defining an ASSOC-Type-List TLV to be carried within an OPEN object. This capability exchange for the DAT (2) **MUST** be done before using the disjoint association. Thus, the PCEP speaker **MUST** include the DAT in the ASSOC-Type-List TLV and **MUST** receive the same from the PCEP peer before using the Disjoint Association Group (DAG) in PCEP messages.

This Association type is considered to be both dynamic and operator-configured in nature. As per [RFC8697], the association group could be manually created by the operator on the PCEP peers, and the LSPs belonging to this association are conveyed via PCEP messages to the PCEP peer; alternately, the association group could be created dynamically by the PCEP speaker, and both the association group information and the LSPs belonging to the association group are conveyed to the PCEP peer. The Operator-configured Association Range **MUST** be set for this association-type to mark a range of Association Identifiers that are used for operator-configured associations to avoid any Association Identifier clash within the scope of the Association Source. (Refer to [RFC8697].)

A Disjoint Association Group can have two or more LSPs, but a PCE may be limited in the number of LSPs it can take into account when computing disjointness. If a PCE receives more LSPs in the group than it can handle in its computation algorithm, it **SHOULD** apply disjointness computation to only a subset of LSPs in the group. The subset of disjoint LSPs will be decided by PCE as a local policy. Local polices **MAY** define the computational behavior for the other LSPs in the group. For example, the PCE may provide no path, a shortest path, or a constrained path based on relaxing disjointness, etc. The disjoint status of the computed path is informed to the PCC via the DISJOINTNESS-STATUS TLV (see Section 5.2).

There are different types of disjointness identified by the flags (T, S, N, and L) in the DISJOINTNESS-CONFIGURATION TLV (see Section 5.2). All LSPs in a particular Disjoint Association Group **MUST** use the same combination of T, S, N, and L flags in the DISJOINTNESS-CONFIGURATION TLV. If a PCEP peer receives a PCEP message for LSPs belonging to the same Disjoint Association Group but having an inconsistent combination of T, S, N, and L flags, the PCEP peer **MUST NOT** add the LSPs to the Disjoint Association Group and **MUST** reply with a PCErr with Error-Type 26 (Association Error) and Error-value 6 (Association information mismatch).

A particular LSP **MAY** be associated to multiple Disjoint Association Groups, but in that case, the PCE **SHOULD** try to consider all the Disjoint Association Groups during path computation, if possible. Otherwise, a local policy **MAY** define the computational behavior. If a PCE does not support such a path computation, it **MUST NOT** add the LSP into the association group and **MUST** return a PCErr with Error-Type 26 (Association Error) and Error-value 7 (Cannot join the association group).

5.2. Disjoint TLVs

The Disjoint Association Group (ASSOCIATION object with Association type = 2 for DAT) **MUST** carry the following TLV:

- DISJOINTNESS-CONFIGURATION TLV: Used to communicate some disjointness configuration parameters. This is applicable for all PCEP messages that include DAG.

In addition, the Disjoint Association Group (ASSOCIATION object with Association type = 2 for DAT) **MAY** carry the following TLVs:

- DISJOINTNESS-STATUS TLV: Used to communicate the status of the computed disjointness. This is applicable for messages from a PCE to a PCC only (i.e., PCUpd, PCInitiate, or PCRep messages).
- VENDOR-INFORMATION-TLV: Used to communicate arbitrary vendor-specific behavioral information, described in [RFC7470].
- OF-List TLV: Used to communicate the disjointness objective function. See Section 5.3.

The DISJOINTNESS-CONFIGURATION TLV is shown in the following figure:

```
 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 = 46             |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                 Flags                               |T|P|S|N|L|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 3: DISJOINTNESS-CONFIGURATION TLV

Type: 46

Length: Fixed value of 4 bytes.

Flags:

L (Link Diverse) bit: When set, this indicates that the computed paths within the Disjoint Association Group **MUST NOT** have any link in common.

N (Node Diverse) bit: When set, this indicates that the computed paths within the Disjoint Association Group **MUST NOT** have any node in common.

S (SRLG Diverse) bit: When set, this indicates that the computed paths within the Disjoint Association Group **MUST NOT** share any SRLG (Shared Risk Link Group).

P (Shortest Path) bit: When set, this indicates that the computed path of the LSP **SHOULD** satisfy all the constraints and objective functions first without considering the diversity constraint. This means that all of the LSPs with P flag set in the association group are computed first, as if the disjointness constraint has not been configured; then, with those LSPs fixed, the other LSPs with P flag unset in the association group are computed by taking into account the disjointness constraint. The role of P flag is further described with examples in Section 5.5.

T (Strict Disjointness) bit: When set, if disjoint paths cannot be found, the PCE **MUST** return no path for LSPs that could not be disjoint. When unset, the PCE is allowed to relax disjointness by either applying a requested objective function (cf. Section 5.3) or using the local policy if no objective function is requested (e.g., using a lower disjoint type (link instead of node) or fully relaxing disjointness constraint). See Section 5.6 for further details.

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

If a PCEP speaker receives a Disjoint Association Group (ASSOCIATION object with Association type = 2 for DAT) without the DISJOINTNESS-CONFIGURATION TLV, it **SHOULD** reply with a PCErr Error-Type 6 (Mandatory Object missing) and Error-value 15 (DISJOINTNESS-CONFIGURATION TLV missing).

The DISJOINTNESS-STATUS TLV uses the same format as the DISJOINTNESS-CONFIGURATION TLV with a different type 47 (in the TLV). The L, N, and S flags are set if the respective disjointness criterion was requested and the computed paths meet it. The P flag indicates that the computed path is the shortest path (computed first without taking disjointness constraints into consideration but considering other constraints).

The T flag has no meaning in the DISJOINTNESS-STATUS TLV and **MUST NOT** be set while sending and **MUST** be ignored on receipt.

Any document defining a new flag for the DISJOINTNESS-CONFIGURATION TLV automatically defines a new flag with the same name and in the same location in DISJOINTNESS-STATUS TLV; the semantics of the flag in the DISJOINTNESS-STATUS TLV **MUST** be specified in the document that specifies the flag in the DISJOINTNESS-CONFIGURATION TLV.

5.3. Disjointness Objective Functions

An objective function (OF) **MAY** be applied to the disjointness computation to drive the PCE computation behavior. In this case, the OF-List TLV (defined in [RFC5541]) is used as an optional TLV in the ASSOCIATION object. Whereas the PCEP OF-List TLV allows multiple OF-codes inside the TLV, a sender **SHOULD** include a single OF-code in the OF-List TLV when included in the Association Group, and the receiver **MUST** consider the first OF-code only and ignore others if included.

To minimize the common shared resources (Node, Link, or SRLG) between a set of paths during path computation, three new OF-codes are defined:

MSL

Name: Minimize the number of Shared (common) Links.
Objective Function Code: 15
Description: Find a set of paths such that it passes through the least number of shared (common) links.

- A network comprises a set of N links {Li, (i=1...N)}.
- A path P passes through K links {Lpi,(i=1...K)}.
- A set of paths {P1...Pm} have L links that are common to more than one path {Lci, (i=1...L)}.
- Find a set of paths such that the value of L is minimized.

MSS

Name: Minimize the number of Shared (common) SRLGs.
Objective Function Code: 16
Description: Find a set of paths such that it passes through the least number of shared (common) SRLGs.

- A network comprises a set of N links {Li, (i=1...N)}.
- A path P passes through K links {Lpi,(i=1...K)} belonging to unique M SRLGs {Spi, (i=1..M)}.
- A set of paths {P1...Pm} have L SRLGs that are common to more than one path {Sci, (i=1...L)}.
- Find a set of paths such that the value of L is minimized.

MSN

Name: Minimize the number of Shared (common) Nodes.
Objective Function Code: 17
Description: Find a set of paths such that they pass through the least number of shared (common) nodes.

- A network comprises a set of N nodes {Ni, (i=1...N)}.
- A path P passes through K nodes {Npi,(i=1...K)}.
- A set of paths {P1...Pm} have L nodes that are common to more than one path {Nci, (i=1...L)}.
- Find a set of paths such that the value of L is minimized.

If the OF-List TLV is included in the ASSOCIATION object, the first OF-code inside the OF object **MUST** be one of the disjoint OFs defined in this document. If this condition is not met, the PCEP speaker **MUST** respond with a PCErr message with Error-Type 10 (Reception of an invalid object) and Error-value 32 (Incompatible OF code).

5.4. Relationship to SVEC

[RFC5440] defines a mechanism for the synchronization of a set of path computation requests by using the SVEC object, which specifies the list of synchronized requests that can be either dependent or independent. The SVEC object identifies the relationship between the set of path computation requests, identified by 'Request-ID-number' in the RP (Request Parameters) object. [RFC6007] further clarifies the use of the SVEC list for synchronized path computations when computing dependent requests and describes a number of usage scenarios for SVEC lists within single-domain and multi-domain environments.

The SVEC object includes a Flags field that indicates the potential dependency between the set of path computation requests in a similar way as the Flags field in the TLVs defined in this document. The path computation request in the Path Computation Request (PCReq) message **MAY** use both the SVEC and ASSOCIATION objects to identify the related path computation request, as well as the DAG. The PCE **MUST** try to find a path that meets both the constraints. It is possible that the diversity requirement in the association group is different from the one in the SVEC object. The PCE **MUST** consider both the objects (including the flags set inside the objects) as per the processing rules and aim to find a path that meets both of these constraints. In case no such path is possible, the PCE **MUST** send a Path Computation Reply (PCRep) with a NO-PATH object indicating path computation failure, as per [RFC5440]. It should be noted that the LSPs in the association group can be fully same or partially overlapping with the LSPs grouped by the SVEC object in PCReq message.

Some examples of usage are listed below:

- PCReq with SVEC object with node-diverse bit=1 (LSP1,LSP2) and DAG with S=1 (LSP1,LSP2) - both node- and SRLG-diverse path between LSP1 and LSP2.
- PCReq with SVEC object with link-diverse bit=1 (LSP1,LSP2) and DAG with L=1 (LSP1,LSP3) - link-diverse paths between LSP1 and LSP2 and between LSP1 and LSP3. If the DAG is part of the stateful database, any future change in LSP3 will have an impact on LSP1. But any future change in LSP2 will have no impact on LSP1, as LSP2 is part of SVEC object (which is considered once on receipt of the PCReq message only).

5.4.1. SVEC and OF

This document defines three new OF-codes in Section 5.3 to maximize diversity as much as possible. In other words, new OF-codes allow specification of minimization of common shared resources (Node, Link, or SRLG) among a set of paths during path computation.

It may be interesting to note that the diversity flags in the SVEC object and OF for diversity can be used together. Some examples of usage are listed below:

- SVEC object with node-diverse bit=1 - ensure full node diversity.
- SVEC object with node-diverse bit=1 and OF=MSS - full node diversity with as much SRLG diversity as possible.

- SVEC object with domain-diverse bit=1 [RFC8685]; node-diverse bit=1, and OF=MSS - full domain and node diversity with as much SRLG diversity as possible.
- SVEC object with node-diverse bit=1 and OF=MSN - ensure full node diversity.

In the last example above, it is interesting to note that "OF" becomes redundant as "SVEC object" ensures full node diversity; however, this specification does not prohibit redundant constraints while using "SVEC object" and "OF" together for diversity.

5.5. P Flag Considerations

As mentioned in Section 5.2, the P flag (when set) indicates that the computed path of the LSP **SHOULD** satisfy all constraints and objective functions first without considering the diversity constraint.

This means that an LSP with the P flag set should be placed first, as if the disjointness constraint has not been configured, while the other LSPs in the association with the P flag unset should be placed by taking into account the disjointness constraint. Setting the P flag changes the relationship between LSPs to a one-sided relationship (LSP 1 with P=0 depends on LSP 2 with P=1, but LSP 2 with P=1 does not depend on LSP 1 with P=0). Multiple LSPs in the same Disjoint Association Group may have the P flag set. In such a case, those LSPs may not be disjoint from each other but will be disjoint from other LSPs in the group that have the P flag unset.

This could be required in some primary/backup scenarios where the primary path should use the more optimal path available (taking into account the other constraints). When disjointness is computed, it is important for the algorithm to know that it should try to optimize the path of one or more LSPs in the Disjoint Association Group (for instance, the primary path), while other paths are allowed to be costlier (compared to a similar path without the disjointness constraint). Without such a hint, the disjointness algorithm may set a path for all LSPs that may not completely fulfill the customer's requirement.

```
         -----------------------------------------
        /                                         \
       /        +------+                           \
      |         | PCE  |                            |
      |         +------+                            |
      |                                             |
      |                                             |
      | +------+           10           +------+  |
CE1 ****| PE 1 | ----- R1 ---- R2 ------- | PE 2 |**** CE2
      | +------+       |        |         +------+  |
      |                |        |                   |
      |                |        |                   |
      | +------+       |        |         +------+  |
CE3 ****| PE 3 | ----- R3 ---- R4 ------- | PE 4 |**** CE4
      | +------+ \     |               /  +------+  |
      |           \    |     10       /             |
       \           +-- R5 --------- R6             /
        \_________________________________________/
```

Figure 4: Example Topology with Six Internal Routers

Note: In Figure 4, the cost of all the links is 1, unless explicitly marked otherwise.

In the figure above, a customer has two dual-homed sites (CE1/CE3 and CE2/CE4). Let us consider that this customer wants two link disjoint paths between the two sites. Due to physical meshing, the customer wants to use CE1 and CE2 as the primary (and CE3 and CE4 are hosted in a remote site for redundancy purpose).

Without any hint (constraint) provided, the PCE may compute the two link disjoint LSPs together, leading to PE1->PE2 using path PE1->R1->R2->PE2 and PE3->PE4 using PE3->R3->R4->PE4. In this case, even if the disjointness constraint is fulfilled, the path from PE1 to PE2 does not use the best optimal path available in the network (path delay may be higher); the customer requirement is thus not completely fulfilled.

The usage of the P flag allows the PCE to know that a particular LSP should be tied to the best path, as if the disjointness constraint was not requested.

In our example, if the P flag is set to the LSP PE1->PE2, the PCE should use the path PE1->R1->R3->R4->R2->PE2 for this LSP, while the other LSP should be link disjoint from this path. The second LSP will be placed on PE3->R5->R6->PE4, as it is allowed to be costlier.

Driving the PCE disjointness computation may be done in other ways, for instance, setting a metric boundary reflecting a path delay boundary. Other constraints may also be used.

The P flag allows to simply express that the disjointness constraint should not make the LSP worst.

Any constraint added to a path disjointness computation may reduce the chance to find suitable paths. The usage of the P flag, as any other constraint, may prevent finding a disjoint path. In the example above, if we consider that router R5 is down and if PE1->PE2 has the P flag set, there is no room available to place PE3->PE4 (the link disjointness constraint cannot be fulfilled). If PE1->PE2 has the P flag unset, the algorithm may be able to place PE1->PE2 on the R1->R2 link leaving room for PE3->PE4 using the R3->R4 link. When using the P flag or any additional constraint on top of the disjointness constraint, the user should be aware that there is less chance to fulfill the disjointness constraint.

```
          -----------------------------------------
         /                                         \
        /        +------+                           \
       |         | PCE  |                            |
       |         +------+                            |
       |                                             |
       |                                             |
       | +------+           10             +------+  |
 CE1 ****| PE 1 | ----- R1 ---- R2 ------- | PE 2 |**** CE2
       | +------+       |  \     |         +------+  |
       |                |   \2   |                   |
       |                |    \   |                   |
       | +------+       |     \  |         +------+  |
 CE3 ****| PE 3 | ----- R3 ---- R4 ------- | PE 4 |**** CE4
       | +------+                          +------+  |
       |                                             |
        \                                           /
         \_________________________________________/
```

Figure 5: Example Topology with Four Internal Routers

Note: In Figure 5, the cost of all the links is 1, unless explicitly marked otherwise.

In the figure above, we still consider the same previous requirements, so PE1->PE2 LSP should be optimized (P flag set), while PE3->PE4 should be link disjoint and may use a costlier path.

Regarding PE1->PE2, there are two paths that are satisfying the constraints (ECMP): PE1->R1->R4->R2->PE2 (path 1) and PE1->R1->R3->R4->R2->PE2 (path 2). An implementation may choose one of the paths.

If the implementation elects only one path, there is a chance that picking up one path may prevent link disjointness. In our example, if path 2 is used for PE1->PE2, there is no room left for PE3->PE4, while if path 1 is used, PE3->PE4 can be placed on R3->R4 link.

When the P flag is set for an LSP and when ECMPs are available, an implementation should aim to select a path that allows disjointness.

5.6. Disjointness Computation Issues

There may be some cases where the PCE is not able to provide a set of disjoint paths for one or more LSPs in the association.

When the T flag is set (Strict disjointness), if disjointness cannot be ensured for one or more LSPs, the PCE **MUST** reply to a PCReq with a PCRep message containing a NO-PATH object. In case of a PCRpt message, the PCE **MUST** return a PCErr message with Error-Type 26 (Association Error) and Error-value 7 (Cannot join the association group).

In case of a network event leading to an impossible strict disjointness, the PCE **MUST** send a PCUpd message containing an empty Explicit Route Object (ERO) to the corresponding PCCs. In addition to the empty ERO object, the PCE **MAY** add the NO-PATH-VECTOR TLV [RFC5440] in the LSP object.

This document adds new bits in the Flags field of the NO-PATH-VECTOR TLV:

- bit 11: When set, the PCE indicates that it could not find a disjoint path for this LSP.
- bit 10: When set, the PCE indicates that it does not support the requested disjointness computation.

When the T flag is unset, the PCE is allowed to relax disjointness by applying a requested objective function (Section 5.3) if specified. Otherwise, if no objective function is specified, the PCE is allowed to reduce the required level of disjointness as it deems fit. The actual level of disjointness of the paths computed by the PCE can be reported through the DISJOINTNESS-STATUS TLV by setting the appropriate flags in the TLV. While the DISJOINTNESS-CONFIGURATION TLV defines the desired level of disjointness required by configuration, the DISJOINTNESS-STATUS TLV defines the achieved level of disjointness computed.

There are some cases where the PCE may need to completely relax the disjointness constraint in order to provide a path to all the LSPs that are part of the association. A mechanism that allows the PCE to fully relax a constraint is considered by the authors as more global to PCEP rather than linked to the disjointness use case. As a consequence, it is considered out of scope of the document. See [PCE-OPTIONAL] for a proposed mechanism.

6. Security Considerations

This document defines one new PCEP Association type, which by itself does not add any new security concerns beyond those discussed in [RFC5440], [RFC8231], [RFC7470], and [RFC8697]. But adding of a spurious LSP into the Disjoint Association Group could lead to recomputation and setup of all LSPs in the group, which could be used to overwhelm the PCE and the network.

A spurious LSP can have flags that are inconsistent with those of the legitimate LSPs of the group and thus cause LSP allocation for the legitimate LSPs to fail with an error. Also, certain combinations of flags (notably, the 'T' bit) can result in conflicts that cannot be resolved.

Also, as stated in [RFC8697], much of the information carried in the ASSOCIATION object reflects information that can also be derived from the LSP database, but association provides a much easier grouping of related LSPs and messages. This holds true for the DAT as well; thus, this could provide an adversary with the opportunity to eavesdrop on the relationship between the LSPs and understand the network topology.

Thus, securing the PCEP session using Transport Layer Security (TLS) [RFC8253], as per the recommendations and best current practices in BCP 195 [RFC7525], is **RECOMMENDED**.

7. IANA Considerations

7.1. Association Type

This document defines a new Association type, originally described in [RFC8697]. IANA has assigned the following new value in the "ASSOCIATION Type Field" subregistry [RFC8697] within the "Path Computation Element Protocol (PCEP) Numbers" registry:

Type	Name	Reference
2	Disjoint Association	RFC 8800

Table 1: ASSOCIATION Type Field

7.2. PCEP TLVs

This document defines two new PCEP TLVs. IANA has assigned the following values in the "PCEP TLV Type Indicators" subregistry within the "Path Computation Element Protocol (PCEP) Numbers" registry:

TLV Type	TLV Name	Reference
46	DISJOINTNESS-CONFIGURATION	RFC 8800
47	DISJOINTNESS-STATUS	RFC 8800

Table 2: PCEP TLV Type Indicators

IANA has created a new subregistry, named "DISJOINTNESS-CONFIGURATION TLV Flag Field", within the "Path Computation Element Protocol (PCEP) Numbers" registry to manage the Flags field in the DISJOINTNESS-CONFIGURATION TLV. New values are to be assigned by Standards Action [RFC8126]. Each bit should be tracked with the following qualities:

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

The initial contents of this subregistry are shown below:

Bit	Name	Reference
31	L - Link Diverse	RFC 8800
30	N - Node Diverse	RFC 8800
29	S - SRLG Diverse	RFC 8800
28	P - Shortest Path	RFC 8800
27	T - Strict Disjointness	RFC 8800
0-26	Unassigned	

Table 3: DISJOINTNESS-CONFIGURATION TLV Flag Field

7.3. Objective Functions

This document defines three new objective functions. IANA has made the following allocations in the "Objective Function" subregistry within the "Path Computation Element Protocol (PCEP) Numbers" registry:

Code Point	Name	Reference
15	Minimize the number of Shared Links (MSL)	RFC 8800
16	Minimize the number of Shared SRLGs (MSS)	RFC 8800
17	Minimize the number of Shared Nodes (MSN)	RFC 8800

Table 4: Objective Function

7.4. NO-PATH-VECTOR Bit Flags

This document defines new bits for the NO-PATH-VECTOR TLV in the "NO-PATH-VECTOR TLV Flag Field" subregistry of the "Path Computation Element Protocol (PCEP) Numbers" registry. IANA has made the following allocations:

Bit Number	Name	Reference
11	Disjoint path not found	RFC 8800
10	Requested disjoint computation not supported	RFC 8800

Table 5: NO-PATH-VECTOR TLV Flag Field

7.5. PCEP-ERROR Codes

This document defines two new Error-values within existing Error-Types related to disjoint association. IANA has allocated the following new Error-values in the "PCEP-ERROR Object Error Types and Values" subregistry within the "Path Computation Element Protocol (PCEP) Numbers" registry:

Error-Type	Meaning	Error-value	Reference
6	Mandatory Object missing		[RFC5440]
		15: DISJOINTNESS-CONFIGURATION TLV missing	RFC 8800
10	Reception of an invalid object		[RFC5440]
		32: Incompatible OF code	RFC 8800

Table 6: PCEP-ERROR Object Error Types and Values

8. Manageability Considerations

8.1. Control of Function and Policy

An operator **SHOULD** be allowed to configure the Disjoint Association Groups and disjoint parameters at the PCEP peers and associate them with the LSPs. The operator **MUST** be allowed to set the Operator-configured Association Range. The operator **SHOULD** be allowed to set the local policies to define various disjoint computational behavior at the PCE.

8.2. Information and Data Models

An implementation **SHOULD** allow the operator to view the disjoint associations configured or created dynamically. Furthermore, implementations **SHOULD** allow to view disjoint associations reported by each peer and the current set of LSPs in this association. The PCEP YANG module [PCEP-YANG] includes association group information.

8.3. Liveness Detection and Monitoring

Mechanisms defined in this document do not imply any new liveness detection and monitoring requirements in addition to those already listed in [RFC5440].

8.4. Verification of Correct Operations

Apart from the operation verification requirements already listed in [RFC5440], a PCEP implementation **SHOULD** provide parameters related to disjoint path computation, such as number of DAG, number of disjoint path computation failures, etc. A PCEP implementation **SHOULD** log failure events (e.g., incompatible Flags).

8.5. Requirements on Other Protocols

Mechanisms defined in this document do not imply any new requirements on other protocols.

8.6. Impact on Network Operations

Mechanisms defined in Section 8.6 of [RFC5440] also apply to PCEP extensions defined in this document. Additionally, a PCEP implementation **SHOULD** allow a limit to be placed on the number of LSPs that can belong to a DAG.

9. References

9.1. Normative References

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

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

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

[RFC7470] Zhang, F. and A. Farrel, "Conveying Vendor-Specific Constraints in the Path Computation Element Communication Protocol", RFC 7470, DOI 10.17487/RFC7470, March 2015, <https://www.rfc-editor.org/info/rfc7470>.

[RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <https://www.rfc-editor.org/info/rfc8126>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

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

[RFC8253] Lopez, D., Gonzalez de Dios, O., Wu, Q., and D. Dhody, "PCEPS: Usage of TLS to Provide a Secure Transport for the Path Computation Element Communication Protocol (PCEP)", RFC 8253, DOI 10.17487/RFC8253, October 2017, <https://www.rfc-editor.org/info/rfc8253>.

[RFC8685] Zhang, F., Zhao, Q., Gonzalez de Dios, O., Casellas, R., and D. King, "Path Computation Element Communication Protocol (PCEP) Extensions for the Hierarchical Path Computation Element (H-PCE) Architecture", RFC 8685, DOI 10.17487/RFC8685, December 2019, <https://www.rfc-editor.org/info/rfc8685>.

[RFC8697] Minei, I., Crabbe, E., Sivabalan, S., Ananthakrishnan, H., Dhody, D., and Y. Tanaka, "Path Computation Element Communication Protocol (PCEP) Extensions for Establishing Relationships between Sets of Label Switched Paths (LSPs)", RFC 8697, DOI 10.17487/RFC8697, January 2020, <https://www.rfc-editor.org/info/rfc8697>.

9.2. Informative References

[PCE-OPTIONAL] Li, C., Zheng, H., and S. Litkowski, "Extension for Stateful PCE to allow Optional Processing of PCEP Objects", Work in Progress, Internet-Draft, draft-dhody-pce-stateful-pce-optional-06, 9 July 2020, <https://tools.ietf.org/html/draft-dhody-pce-stateful-pce-optional-06>.

[PCEP-YANG] Dhody, D., Hardwick, J., Beeram, V., and J. Tantsura, "A YANG Data Model for Path Computation Element Communications Protocol (PCEP)", Work in Progress, Internet-Draft, draft-ietf-pce-pcep-yang-14, 7 July 2020, <https://tools.ietf.org/html/draft-ietf-pce-pcep-yang-14>.

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

[RFC6007] Nishioka, I. and D. King, "Use of the Synchronization VECtor (SVEC) List for Synchronized Dependent Path Computations", RFC 6007, DOI 10.17487/RFC6007, September 2010, <https://www.rfc-editor.org/info/rfc6007>.

[RFC7525] Sheffer, Y., Holz, R., and P. Saint-Andre, "Recommendations for Secure Use of Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)", BCP 195, RFC 7525, DOI 10.17487/RFC7525, May 2015, <https://www.rfc-editor.org/info/rfc7525>.

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

Acknowledgments

A special thanks to the authors of [RFC8697]; this document borrows some text from it. The authors would also like to thank Adrian Farrel and Julien Meuric for the valuable comments.

Thanks to Emmanuel Baccelli for the RTGDIR review.

Thanks to Dale Worley for a detailed GENART review.

Thanks to Alvaro Retana, Benjamin Kaduk, Suresh Krishnan, Roman Danyliw, Alissa Cooper, and Éric Vyncke for the IESG review.

Contributors

Dhruv Dhody
Huawei Technologies
Divyashree Techno Park, Whitefiled
Bangalore 560066
Karnataka
India
Email: dhruv.ietf@gmail.com

Authors' Addresses

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

Siva Sivabalan
Ciena Corporation
Email: msiva282@gmail.com

Colby Barth
Juniper Networks
Email: cbarth@juniper.net

Mahendra Singh Negi
RtBrick India
N-17L, Floor-1, 18th Cross Rd, HSR Layout Sector-3
Bangalore 560102
Karnataka
India
Email: mahend.ietf@gmail.com