

PCE Working Group
Internet-Draft
Intended status: Standards Track
Expires: March 15, 2018

S. Litkowski
Orange
S. Sivabalan
Cisco Systems, Inc.
C. Barth
Juniper Networks
D. Dhody
Huawei
September 11, 2017

Path Computation Element communication Protocol extension for signaling
LSP diversity constraint

[draft-ietf-pce-association-diversity-02](#)

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 paths for those LSPs. The proposed extension allows a PCC to advertise to a PCE the belonging of a particular LSP to a disjoint-group, thus the PCE knows that LSPs in the same group must be disjoint from each other.

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 March 15, 2018.

Copyright Notice

Copyright (c) 2017 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	2
1.1.	Requirements Language	3
2.	Terminology	3
3.	Motivation	4
4.	Protocol extension	7
4.1.	Association group	7
4.2.	Mandatory TLV	7
4.3.	Optional TLVs	9
4.4.	Disjointness objective functions	9
4.5.	P-flag considerations	9
4.6.	Disjointness computation issues	12
5.	Security Considerations	13
6.	IANA Considerations	13
6.1.	Association object Type Indicators	13
6.2.	PCEP TLVs	14
6.3.	NO-PATH-VECTOR bit Flags	14
6.4.	PCEP-ERROR codes	14
7.	Manageability Considerations	15
7.1.	Control of Function and Policy	15
7.2.	Information and Data Models	15
7.3.	Liveness Detection and Monitoring	15
7.4.	Verify Correct Operations	15
7.5.	Requirements On Other Protocols	15
7.6.	Impact On Network Operations	15
8.	Acknowledgments	15
9.	References	16
9.1.	Normative References	16
9.2.	Informative References	16
	Authors' Addresses	17

[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](#)].

PCEP Extensions for Stateful PCE Model [[I-D.ietf-pce-stateful-pce](#)] describes a set of extensions to PCEP to enable active control of MPLS-TE and GMPLS tunnels. [[I-D.ietf-pce-pce-initiated-lsp](#)] 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.

[[I-D.ietf-pce-association-group](#)] introduces a generic mechanism to create a grouping of LSPs which can then be used to define associations between a set of LSPs and a set of attributes (such as configuration parameters or behaviours) and is equally applicable to the active and passive modes of a stateful PCE [[I-D.ietf-pce-stateful-pce](#)] or a stateless PCE [[RFC5440](#)].

This document specifies a PCEP extension to signal that a particular group of LSPs should use diverse paths including the requested type of diversity. A PCC can use this extension to signal to a PCE the belonging of a particular LSP to a disjoint-group. When a PCE receives LSP states belonging to the same disjoint-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", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [[RFC2119](#)].

[2.](#) Terminology

The following terminology is used in this document.

LSR: Label Switch Router.

MPLS: Multiprotocol Label Switching.

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.

SRLG: Shared Risk Link Group.

3. Motivation

Path diversity is a very common use case today in 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 IP/MPLS core. The customer may use those paths as primary/backup or active/active.

Different level of disjointness may be offered:

- o 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).
- o Node disjointness: the paths of the associated LSPs should transit different nodes (but may use different links that may have some shared fate).
- o SRLG disjointness: the paths of the associated LSPs should transit different links that do not share fate (but may use common transit nodes).
- o 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 from different head-end(s) and may terminate at the same or different tail-end(s).

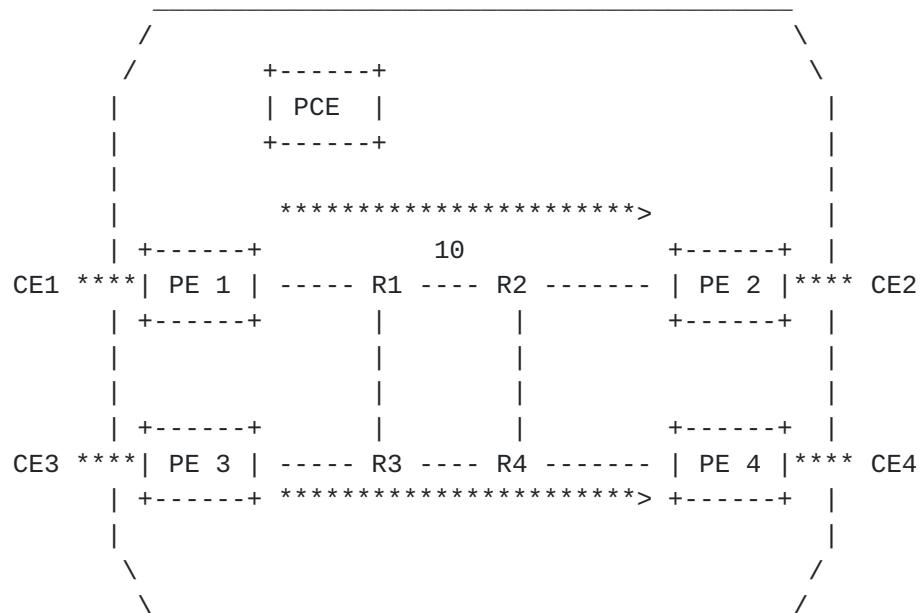


Figure 1 - Disjoint paths with different head-ends and tail-ends

In the figure above, the customer wants to have two disjoint paths between CE1/CE2 and CE3/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.

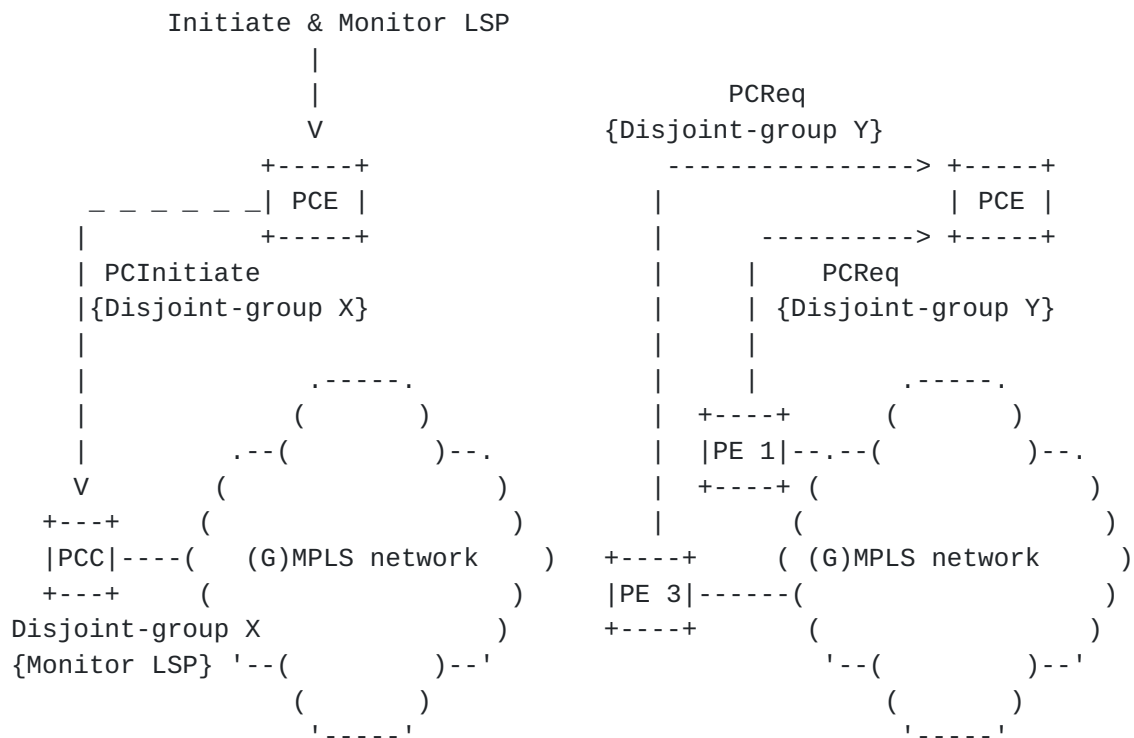
The SVEC (Synchronization VECtor) object [[RFC5440](#)] allows a PCC to request the synchronization of a set of dependent or independent path computation requests. As per [[RFC5440](#)], the SVEC object is optional and may be carried within a PCReq message. It uses the Request-ID-number carried within the respective RP (Request Parameters) object to identify the computation request that should be synchronized.

The PCEP extension for stateful PCE [[I-D.ietf-pce-stateful-pce](#)] defined new PCEP messages - PCRpt, PCUpd and PCInitiate. These messages uses 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 as described in [[I-D.ietf-pce-association-group](#)] equally applicable to the active and passive modes of a stateful PCE is better suited.

Using PCEP, the PCC MUST indicate that disjoint path computation is required, such indication SHOULD include disjointness parameters such as the type of disjointness, the disjoint group-id, and any

customization parameters according to the configured local policy. As mentioned previously, the extension described in [\[I-D.ietf-pce-association-group\]](#) is well suited to associated a set of LSPs with a particular disjoint-group.

The management of the disjoint group-ids will be a key point for the operator as the Association ID field is limited to 65535. The local configuration of IPv4/IPv6 association source, or Global Association Source/Extended Association ID should allow to overcome this limitation. For example, when a PCC or PCE initiates all the LSPs in a particular disjoint-group, it can set the IPv4/IPv6 association source can be set to one of its IP address. When disjoint LSPs are initiated from different head-ends, a unique association identifier SHOULD be used for those LSPs: this can be achieved by setting the IPv4/IPv6 source address to a common value (zero value can be used) as well as the Association ID.



Case 1: Disjointness initiated by PCE and enforced by PCC

Case 2: Disjointness initiated by PCC and enforced by PCE

Figure 2 - Sample use-cases for carrying disjoint-group over PCEP session

Using the disjoint-group within a PCUpd or PCInitiate may have two purposes:

- o Information: in case the PCE is performing the path computation, it may communicate to the PCC the locally used parameters in the attribute-list of the LSP.
- o Configuration: in case the PCE is updating the diversity parameters.

4. Protocol extension

4.1. Association group

As per [[I-D.ietf-pce-association-group](#)], LSPs are associated with other LSPs with which they interact by adding them to a common association group. The Association ID will be used to identify the disjoint group a set of LSPs belongs to. This document defines a new Association type, based on the generic Association object -

- o Association type = TBD1 ("Disjointness Association Type").

A disjoint group can have two or more LSPs. But a PCE may be limited in how many 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 the implementation.

Local policies on the PCC or PCE 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.

Associating a particular LSP to multiple disjoint groups is authorized from a protocol perspective, however there is no insurance that the PCE will be able to compute properly the multi-disjointness constraint.

4.2. Mandatory TLV

The disjoint group MUST carry the following TLV:

- o DISJOINTNESS-INFORMATION-TLV: Used to communicate some disjointness specific parameters.

The DISJOINTNESS-INFORMATION-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 = [TBD2]           |           Length           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|           Flags                     |T|P|S|N|L|
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

Flags:

- * L (Link diverse) bit: when set, this indicates that the computed paths within the disjoint group MUST NOT have any link in common.
- * N (Node diverse) bit: when set, this indicates that the computed paths within the disjoint group MUST NOT have any node in common.
- * S (SRLG diverse) bit: when set, this indicates that the computed paths within the disjoint 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 satisfies all constraints and objective functions first without considering the diversity constraint. This means that an LSP with P flag set should be placed as if the disjointness constraint has not been configured, while the other LSP in the association with P flag unset should be placed by taking into account the disjointness constraint. Setting P flag changes the relationship between LSPs to a unidirectional relationship (LSP 1 with P=0 depends of LSP 2 with P=1, but LSP 2 with P=1 does not depend of LSP 1 with P=0).
- * T (Strict disjointness) bit: when set, if disjoint paths cannot be found, PCE should return no path for LSPs that could not be be disjoint. When unset, PCE is allowed to relax disjointness by using either applying a requested objective function or any other behavior if no objective function is requested (e.g.: using a lower disjoint type (link instead of node) or relaxing disjointness constraint at all).

If a PCEP speaker receives a disjoint-group without DISJOINTNESS-INFORMATION-TLV, it SHOULD reply with a PCErr Error-type=6 (Mandatory Object missing) and Error-value=TBD7.

4.3. Optional TLVs

The disjoint group MAY carry some optional TLVs including but not limited to:

- o VENDOR-INFORMATION-TLV: Used to communicate arbitrary vendor specific behavioral information, described in [[RFC7150](#)].

4.4. Disjointness objective functions

An objective function 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 Group Object. The PCEP OF-List TLV allow 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. The OF-Code to maximize diversity are specified in ([[I-D.dhody-pce-of-diverse](#)])).

4.5. P-flag considerations

As mentioned in [Section 4.2](#), the P-flag (when set) indicates that the computed path of the LSP SHOULD satisfies all constraints and objective functions first without considering the diversity constraint. 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 group (for instance the primary path) while other paths are allowed to be longer (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 fulfil the customer requirement.

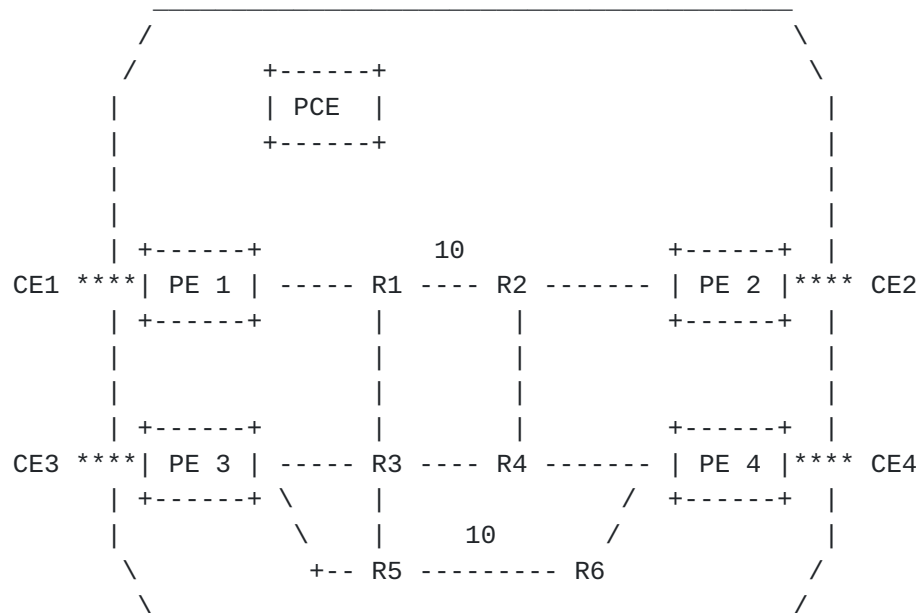


Figure 3

In the figure above, a customer has two dual homed sites (CE1/CE3 and CE2/CE4). This customer wants two disjoint paths between the two sites. Due to physical meshings, the customer wants to use CE1 and CE2 as primary (for instance CE3 and CE4 are hosted in a remote site for redundancy purpose compared the customer hosts).

Without any hint (constraint) provided, the PCE may compute the two disjoint LSPs as a batch leading to PE1->PE2 using a 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 (RTD 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 disjoint from this path. The second LSP will be placed on PE3->R5->R6->PE4 as it is allowed to be longer.

Driving the PCE disjointness computation may be done in other ways by for instance setting a metric boundary reflecting an RTD boundary. Other constraints may also be used.

The P-Flag allows a simple expression 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 to find a disjoint path. In the example above, if we consider that the router R5 is down, if PE1->PE2 has the P-flag set, there is no room available to place PE3->PE4 (the disjointness constraint cannot be fulfilled). If PE->PE2 has the P-flag unset, the algorithm may be able to place PE1->PE2 on R1->R2 link leaving a room for PE3->PE4 using the R3->R4 link. When using 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.

Multiple LSPs in the same disjoint 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 others LSPs in the group that have the P-flag unset.

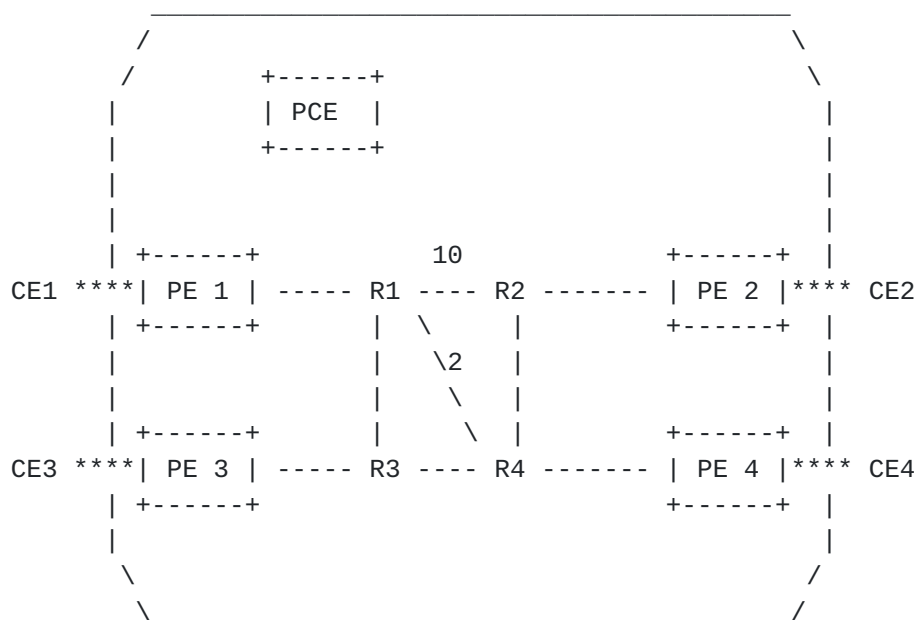


Figure 4

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 disjoint and may use a longer 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 or even use boths (using both may happen in case Segment Routing TE is used, allowing ECMP).

If the implementation elects only one path, there is a chance that picking up one path may prevent 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 P-flag is set for an LSP and when ECMPs are available, an implementation MAY select a path that allows disjointness.

4.6. 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-bit is set (Strict disjointness requested), if disjointness cannot be found for one or more LSPs, the PCE SHOULD reply with a PCUpd message containing an empty ERO. 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 NO-PATH-VECTOR TLV:

bit "TBD3": when set, the PCE indicates that it could not find a disjoint path for this LSP.

bit "TBD4": when set, the PCE indicates that it does not support the requested disjointness computation.

The NO-PATH-VECTOR TLV MAY also be used when T-bit is unset and when the PCE cannot provide a path for an LSP in the disjoint group.

When the T-bit is unset, the PCE is allowed to relax the constraint if it cannot be satisfied. This document introduces a new RELAXED-CONSTRAINT-TLV that MAY be used by a PCE to indicate that some constraints have been relaxed.

When used, the RELAXED-CONSTRAINT-TLV SHOULD be included in the LSP Object of a PCUpd message. The RELAXED-CONSTRAINT-TLV has the following format:


```

      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 = [TBD5]               |               Length               |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| Object-Class#1| OT#1 |Res|P|I|   Object Length (bytes)   |               |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|               |               |               |               |               |
//               (Object body)               //               |
|               |               |               |               |               |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
      ...
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| Object-Class#n| OT#n |Res|P|I|   Object Length (bytes)   |               |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|               |               |               |               |               |
//               (Object body)               //               |
|               |               |               |               |               |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

All LSPs in a particular disjoint group MUST use the same combination of T,S,N,L flags in the DISJOINTNESS-INFORMATION-TLV. If a PCE receives PCRpt messages for LSPs belonging to the same disjoint group but having an inconsistent combination of T,S,N,L flags, the PCE SHOULD NOT try to compute disjointness path and SHOULD reply a PCErr with Error-type 5 (Policy Violation) and Error-Value TBD6 (Inconsistent parameters in DISJOINTNESS-INFORMATION TLV) to all PCCs involved in the disjoint group.

5. Security Considerations

This document defines one new type for association, which do not add any new security concerns beyond those discussed in [\[RFC5440\]](#), [\[I-D.ietf-pce-stateful-pce\]](#) and [\[I-D.ietf-pce-association-group\]](#) in itself.

6. IANA Considerations

6.1. Association object Type Indicators

This document defines the following new association type originally defined in [\[I-D.ietf-pce-association-group\]](#).

Value	Name	Reference
TBD1	Disjoint-group	Association
Type	[This I.D.]	

6.2. PCEP TLVs

This document defines the following new PCEP TLVs:

Value	Name	Reference
TBD2	DISJOINTNESS-INFORMATION-TLV	[This I.D.]
TBD5	RELAXED-CONSTRAINT-TLV	[This I.D.]

IANA is requested to manage the space of flags carried in the DISJOINTNESS-INFORMATION TLV defined in this document, numbering them from 0 as the least significant bit.

New bit numbers may be allocated in future.

IANA is requested to allocate the following bit numbers in the DISJOINTNESS-INFORMATION-TLV flag space:

Bit Number	Name	Reference
0	Link disjointness	[This I.D.]
1	Node disjointness	[This I.D.]
2	SRLG disjointness	[This I.D.]
3	Shortest-path	[This I.D.]
4	Strict disjointness	[This I.D.]

6.3. NO-PATH-VECTOR bit Flags

This documents defines new bits for the NO-PATH-VECTOR TLV in the "NO-PATH-VECTOR TLV Flag Field" sub-registry of the "Path Computation Element Protocol (PCEP) Numbers" registry:

Bit Number	Name	Reference
TBD3	Disjoint path not found	[This I.D.]
TBD4	Requested disjointness computation not supported	[This I.D.]

6.4. PCEP-ERROR codes

IANA is requested to allocate new Error Types and Error Values within the " PCEP-ERROR Object Error Types and Values" sub-registry of the PCEP Numbers registry, as follows:

Error-Type	Meaning
5	Policy violation Error-value=TBD6: Inconsistent parameters in DISJOINTNESS-INFORMATION TLV
6	Mandatory Object missing Error-value=TBD7: DISJOINTNESS-INFORMATION TLV missing

7. Manageability Considerations

7.1. Control of Function and Policy

An operator MUST be allowed to configure the disjointness associations and parameters at PCEP peers and associate it with the LSPs.

7.2. Information and Data Models

[RFC7420] describes the PCEP MIB, there are no new MIB Objects for this document.

7.3. 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](#)].

7.4. Verify Correct Operations

Mechanisms defined in this document do not imply any new operation verification requirements in addition to those already listed in [[RFC5440](#)].

7.5. Requirements On Other Protocols

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

7.6. Impact On Network Operations

Mechanisms defined in this document do not have any impact on network operations in addition to those already listed in [[RFC5440](#)].

8. Acknowledgments

A special thanks to author of [[I-D.ietf-pce-association-group](#)], this document borrow some of the text from it. Authors would also like to thank Adrian Farrel for his useful comments.

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>>.
- [RFC4655] Farrel, A., Vasseur, J., 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>>.
- [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>>.
- [I-D.ietf-pce-association-group]
Minei, I., Crabbe, E., Sivabalan, S., Ananthakrishnan, H., Dhody, D., and Y. Tanaka, "PCEP Extensions for Establishing Relationships Between Sets of LSPs", [draft-ietf-pce-association-group-04](#) (work in progress), August 2017.
- [I-D.ietf-pce-stateful-pce]
Crabbe, E., Minei, I., Medved, J., and R. Varga, "PCEP Extensions for Stateful PCE", [draft-ietf-pce-stateful-pce-21](#) (work in progress), June 2017.
- [I-D.dhody-pce-of-diverse]
Dhody, D. and Q. Wu, "PCE support for Maximizing Diversity", [draft-dhody-pce-of-diverse-07](#) (work in progress), May 2017.

9.2. Informative References

- [RFC7150] Zhang, F. and A. Farrel, "Conveying Vendor-Specific Constraints in the Path Computation Element Communication Protocol", [RFC 7150](#), DOI 10.17487/RFC7150, March 2014, <<https://www.rfc-editor.org/info/rfc7150>>.

[RFC7420] Koushik, A., Stephan, E., Zhao, Q., King, D., and J. Hardwick, "Path Computation Element Communication Protocol (PCEP) Management Information Base (MIB) Module", [RFC 7420](#), DOI 10.17487/RFC7420, December 2014, <<https://www.rfc-editor.org/info/rfc7420>>.

[I-D.ietf-pce-pce-initiated-lsp] Crabbe, E., Minei, I., Sivabalan, S., and R. Varga, "PCEP Extensions for PCE-initiated LSP Setup in a Stateful PCE Model", [draft-ietf-pce-pce-initiated-lsp-10](#) (work in progress), June 2017.

Authors' Addresses

Stephane Litkowski
Orange

EMail: stephane.litkowski@orange.com

Siva Sivabalan
Cisco Systems, Inc.
2000 Innovation Drive
Kanata, Ontario K2K 3E8
Canada

EMail: msiva@cisco.com

Colby Barth
Juniper Networks

EMail: cbarth@juniper.net

Dhruv Dhody
Huawei

EMail: dhruv.dhody@huawei.com

