

# Encoding of Data Structure (DS) in the Path Computation Element Communication Protocol (PCEP).

draft-dhody-pce-pcep-ds-00

Dhruv Dhody ([dhruvd@huawei.com](mailto:dhruvd@huawei.com))

Udayasree Palle ([udayasreepalle@huawei.com](mailto:udayasreepalle@huawei.com))

# Introduction

- Backward-Recursive PCE-Based Computation (BRPC) defines VSPT [Virtual Shortest Path Tree] as a default de-facto data structure for PCRep message in inter-domain scenarios.

```
VSPT(i): MP2P (multipoint-to-point) tree returned by PCE(i) to  
PCE(i-1):
```

```
          Root (TE LSP destination)  
         /      |      \  
    BN-en(1,i)  BN-en(2,i) ... BN-en(j,i).
```

```
    where [X-en(i)] is the number of  
    entry BNs in domain i and j<= [X-en(i)]
```

- Need for multiple Data Structure based different scenarios like P2MP, Protection, HPCE, OF, Synchronization
- PCEP is limited with just one Data Structure.
- Extend PCEP to allow multiple data structures.

# Motivation for this work

- [PCE-P2MP-PROCEDURES] describes the need for an extended VSPT for computation of the best core-tree.
- [RFC 6007] describes the need for disjoint VSPT in case of Synchronized Dependent Path Computations.
- VSPT does not fit into hierarchical PCE paradigm described in [PCE-HIERARCHY-FWK].
- VSPT does not work well with constraints like HOP-LIMIT.
- Since PCEP allow multiple Objective Function (OF); it is natural to extend PCEP to support multiple Data Structure based on path computation scenario

# PCEP Extension

- **DS-List TLV:** Support advertisement of supported Data Structure (DS) in Open Message.
- **DS Object:** In Request Message, it indicates the desired/required data structure to be applied by PCE. In Reply Message, it indicates the data structure that was used by the PCE during path computation and in the reply message.
- **New Flag in RP:** Indicating PCE to supply DS in reply message.
- **Message Format:** <DS> object is carried in PCReq / PCRep. It can be used in conjunction with SVEC as well as individual requests.

# Next Steps

- Evaluate the need within the WG and extend PCEP accordingly.
- IANA and security considerations needs analysis.

Questions  
&  
Comments?

Thanks!

# Backup Information

Inter Domain:

draft-zhao-pce-pcep-inter-domain-p2mp-procedures:

Note that the application of BRPC in the aforementioned procedure differs from the typical one since paths returned from a downstream PCE are not necessary pruned from the solution set by intermediate PCEs.

The reason for this is that if the PCE in a downstream domain does the pruning and returns the single optimal sub-path to its parent PCE, BRPC insures that the ingress PCE will get all the best optimal sub-paths for each LN (Leaf Border Nodes), but the combination of these single optimal sub-paths into a P2MP tree is not necessarily optimal even if each S2L (Source-to-Leaf) sub-path is optimal.

Without trimming, the ingress PCE will get all the possible S2L sub-paths set for LN, and eventually by looking through all the combinations, and taking one sub-path from each set to built one P2MP tree it finds the optimal tree.



# Backup Information

## Use of the Synchronization VECtor (SVEC) List for Synchronized Dependent Path Computations: RFC 6007

### 6.1. Disjoint VSPT

The BRPC procedure constructs a VSPT to inform the enquiring PCE of potential paths to the destination node.

In the end-to-end diverse path computation, diversity (or disjointness) information among the potential paths must be preserved in the VSPT to ensure an end-to-end disjoint path. In order to preserve diversity (or disjointness) information, disjoint VSPTs are sent in the PCEP PCRep message. The PCReq containing a SVEC object with the appropriate diverse flag set would signal that the PCE should compute a disjoint VSPT.

A definition of the disjoint VSPT is a collection of VSPTs, in which each VSPT contains a potential set of primary and secondary paths.

# Backup Information

PathList

PathList(i) = {

- For  $i=n$ : all possible paths meeting the constraints between BN-en(k,n) to destination
- For  $1 < i < n$ : [all possible paths meeting the constraints between BN-en(k,i) to BN-ex(l,i)] + PathList(i+1)
- For  $i=1$ : [all possible paths meeting the constraints between Source to BN-ex(l,1)] + PathList(2)