



# PCECC for P2MP / SRv6

draft-dhody-pce-pcep-extension-pce-controller-p2mp-00

draft-dhody-pce-pcep-extension-pce-controller-srv6-00

Dhruv Dhody / Zhenbin Li

# Introduction

## RFC 8283 is published

- An architecture for use of PCE/PCEP in a network with central control.
- Introduces the architecture for PCE as a central controller and examines the motivations/applicability for PCEP as a control protocol in this environment.
- A PCE-based central controller can simplify the processing of a distributed control plane by blending it with elements of SDN and without necessarily completely replacing it.

## PCECC Extensions

- Basic PCECC [I-D.zhao-pce-pcep-extension-for-pce-controller] WG adoption call
- PCECC-SR [I-D.zhao-pce-pcep-extension-pce-controller-sr] WG adoption pending
- Both were presented in IETF 102

# Basic PCECC Mode

---

LSPs are provisioned as explicit label instructions at each hop on the end-to-end path.

---

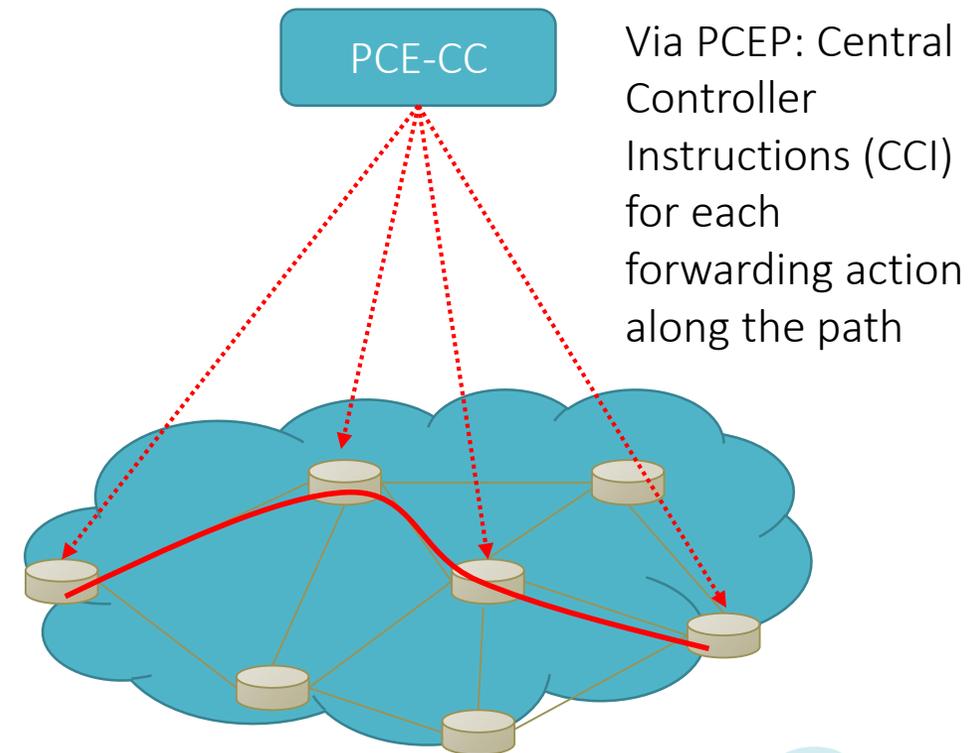
Each router along the path must be told what label forwarding instructions to program and what resources to reserve.

---

The controller uses PCEP to communicate with each router along the path of the end-to-end LSP.

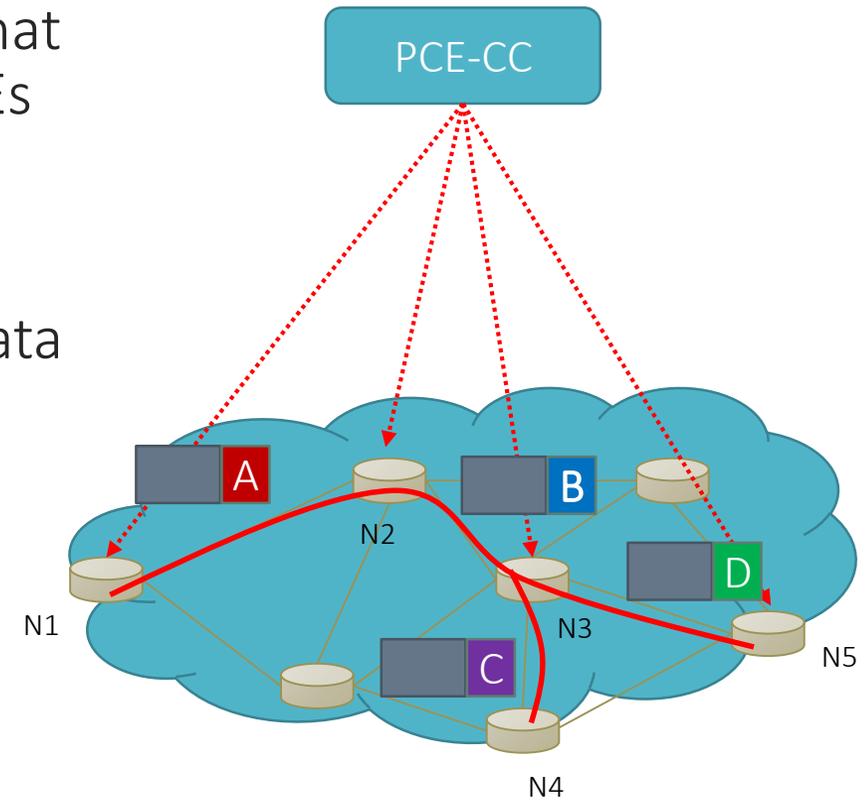
---

PCECC will take responsibility for managing some part of the MPLS label space for each of the routers that it controls



# PCECC-P2MP

- [I-D.ietf-pce-stateful-pce-p2mp] specify the extensions that are necessary in order for the deployment of stateful PCEs to support P2MP TE LSPs
- PCECC – P2MP additional requirement
  - A branch node is an LSR that replicates the incoming data on to one or more outgoing interfaces.
- At the branch node N3 -
  - Incoming Label : B
  - Outgoing Label: (C, towards N4) and (D, towards N5)
  - 3 instances of CCI object in PCInitiate message for each label



# PCECC-P2MP

- A new M-bit is added in PCECC-CAPABILITY
- CCI Object Type = 1 for MPLS Label is already defined
- For P2MP branch node, multiple CCI objects with outgoing flag is included.
- No change in RBNF required; <cci-list> takes care of it!

```
<PCInitiate Message> ::= <Common Header>  
                           <PCE-initiated-lsp-list>
```

Where:

```
<Common Header> is defined in [RFC5440]
```

```
<PCE-initiated-lsp-list> ::= <PCE-initiated-lsp-request>  
                             [<PCE-initiated-lsp-list>]
```

```
<PCE-initiated-lsp-request> ::=  
    (<PCE-initiated-lsp-instantiation> |  
     <PCE-initiated-lsp-deletion> |  
     <PCE-initiated-lsp-central-control>)
```

```
<PCE-initiated-lsp-central-control> ::= <SRP>  
                                         (<LSP>  
                                          <cci-list>) |  
                                         (<FEC>  
                                          <CCI>)
```

```
<cci-list> ::= <CCI>  
               [<cci-list>]
```

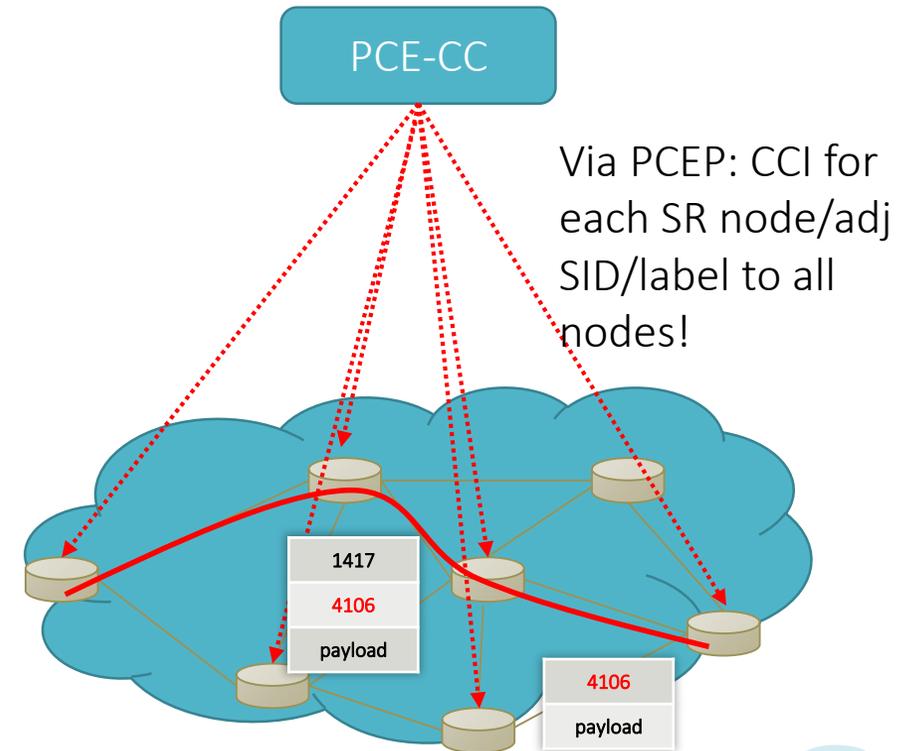
# SR PCECC Mode

PCECC can use PCEP for SR SID (Segment Identifier) distribution on the SR nodes.

- SR SID is just another central controller instruction (CCI)
- A new CCI Object Type for SR is defined

PCECC needs to be in control of label space to make SR SID allocation

- Node/Prefix
- Adjacency

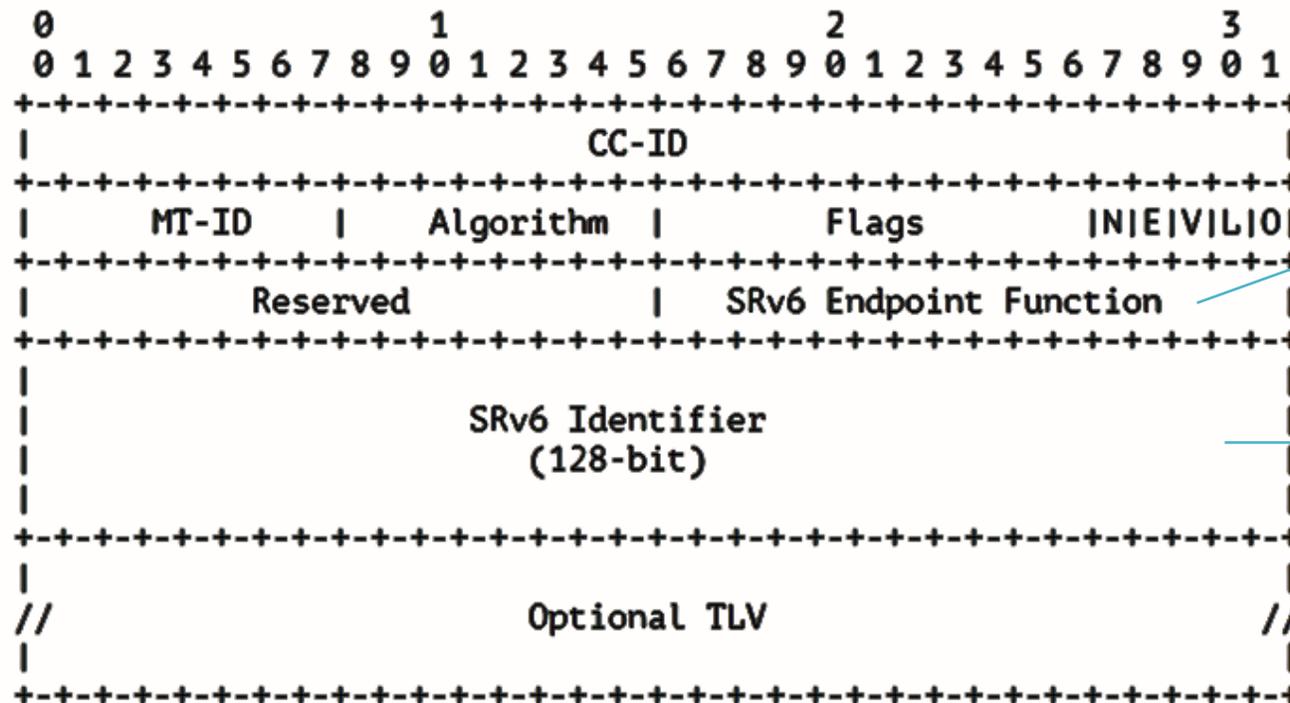


# PCECC-SRv6

- [I-D.negi-pce-segment-routing-ipv6] extends the PCEP-SR procedure to include support for SRv6 paths.
- PCECC-SRv6 require support for SRv6 SID allocation and distribution via PCEP.
- A new CCI object type for SRv6 is needed.
- A new I-bit is added in PCECC-CAPABILITY for SR in IPv6 and a new PST is defined.
- On receiving the SRv6 node/prefix SID allocation:
  - For the local SID, update SID with associated END function in "My Local SID Table".
  - For the non-local SID, use the local routing information to determine the next-hop and download the forwarding instructions accordingly.
- Similarly each adjacency is assigned a SRv6 adj SID

# PCECC-SRv6

CCI Object-Type is TBD for SRv6 as below -



These fields are already defined for SR-MPLS!

Associated "function" of the SRv6 SID

128-bit SRv6 Segment

Question: is it worth encoding LOC:FUNC separately?

# Question & Next Steps

---

Are these features useful?

*They align to the use-cases in TEAS!*

---

Are these extensions on the right track?

*They seems to require minimal changes to PCECC!*

---

Reviews?

*Lets refine the work!*

---



Thank You!