# Stateful PCE for P2MP LSP

draft-palle-pce-stateful-pce-p2mp-08

draft-palle-pce-stateful-pce-initiated-p2mp-lsp-07


| | |
|---|---|
| Udayasree Palle (Huawei) | Dhruv Dhody (Huawei) |
| Yosuke Tanaka (NTT) | Zafar Ali (Cisco) |
| Vishnu Pavan Beeram (Juniper) | |

# Stateful PCE for P2MP

Stateful PCE is equally applicable for P2MP TE LSP

- For global optimization
- Restoration and recovery
- Protection for P2MP

P2MP Path Computation are much more CPU intensive, delegating full control to a specialized PCE can be useful

For P2MP, where the size of message is much large, stateful PCE allow referring to existing LSPs via an PLSP-ID.

PCE-Initiated P2MP LSP dynamic changes based on the application demands (IPTV, MVPN) including add/del of the leaves for existing P2MP LSP.

# Stateful Operations

Capability Advertisement

- PCE Capability advertisement via IGP

P2MP LSP State Synchronization

P2MP LSP Update

P2MP LSP Report

P2MP LSP Delegation

P2MP LSP Initiation

- And deletion

Add / Remove leaves to existing P2MP LSP

# PCEP Extension

## Capability Advertisement

- 3 new bits added to STATEFUL-PCE-CAPABILITY TLV
  - N (P2MP-CAPABILITY)
  - M (P2MP-LSP-UPDATE-CAPABILITY)
  - P (P2MP-LSP-INSTANTIATION-CAPABILITY)
- *Similar bits added in PCE-CAP-FLAGS sub-TLV too (PCE discovery via IGP)*

## LSP Object

- New Flags P2MP (N) and Fragmentation (F) bits
- PLSP-ID identify a (full) P2MP TE LSP uniquely.
- P2MP-LSP-IDENTIFIER TLV
  - Identify RSVP signaled P2MP LSP-ID
  - IPv4 and IPv6

# PCEP Extension

## S2LS (Source to Leaves)

- Report state of one or more leaves encoded within the END-POINTS object.
- O in LSP - operational status of the full P2MP TE LSP & O in S2L - the operational status of a group of leaves encoded within the END-POINTS object.

## Support for Passive and active stateful PCE mode

## Message Fragmentation

- P2MP PCRpt, PCUpd and PCInitiate may not fit into a single PCEP message.
- Use a new F-bit in the LSP object.

# Leaf Type & Operational Status

| The P2MP END-POINTS object for specifying address of P2MP leaves are grouped based on leaf types. | | | |
|---|---|---|---|
| New leaves to add (leaf type = 1) | Old leaves to remove (leaf type = 2) | Old leaves whose path can be modified/reoptimized (leaf type = 3) | Old leaves whose path must be left unchanged (leaf type = 4) |

- When reporting the status of a P2MP TE LSP, the destinations are grouped in END-POINTS object based on the operational status (O field in S2LS object) and leaf type (in END-POINTS).
  - This way the leaves that share the same operational status are grouped together!

- For reporting the status of delegated P2MP TE LSP, leaf-type = 3, where as for non-delegated P2MP TE LSP, leaf-type = 4 is used.
- For delegated P2MP TE LSP configuration changes are reported via PCRpt message. For example, adding of new leaves END-POINTS (leaf-type = 1) is used where as removing of old leaves (leaf-type = 2) is used.

# Recent Updates

## Last presented during IETF 90 (Toronto) – aligned to the latest stateful drafts

| | | |
|---|---|---|
| Addition of Stateful P2MP PCE capability in IGP | Update in PCRpt/ PCUpd/ PcInitiate/ PCReq message format<br>• SERO / SRRO<br>• Intended and actual path | Error Handling<br>• S2LS / ENDPOINT object missing<br>• Fragmentation error<br>• Backward Compatibility |

# Next Steps

No pending comments!

- More reviews are welcome!

Good base to be worked on by the WG

- Only missing piece in the WG adopted stateful PCE drafts
- WG adoption call?

# Questions & Comments?

# Backup Slides

# Messages

The format of PCRpt message is as follows:

```
<PCRpt Message> ::= <Common Header>
                    <state-report-list>
Where:

<state-report-list> ::= <state-report>
                        [<state-report-list>]

<state-report> ::= [<SRP>]
                   <LSP>
                   <end-point-path-pair-list>
                   <attribute-list>
Where:

<end-point-path-pair-list>::=
                   [<END-POINTS>]
                   [<S2LS>]
                   <intended_path>
                   [<actual_path>]
                   [<end-point-path-pair-list>]

<intended_path> ::= (<ERO>|<SERO>)
            [<intended_path>]

<actual_path> ::= (<RRO>|<SRRO>)
            [<actual_path>]
```

<attribute-list> is defined in [RFC5440] and extended by PCEP extensions.

```
<PCUpd Message> ::= <Common Header>
                    <update-request-list>

Where:

<update-request-list> ::= <update-request>
                          [<update-request-list>]

<update-request> ::= <SRP>
                     <LSP>
                     <end-point-path-pair-list>

<attribute-list>
   Where:

<end-point-path-pair-list>::=
                [<END-POINTS>]
                <path>
                [<end-point-path-pair-list>]

<path> ::= (<ERO>|<SERO>)
           [<path>]
```

<attribute-list> is defined in [RFC5440] and extended by PCEP extensions.

# Messages

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

<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-instantiation> ::= <SRP>
                                      <LSP>
                                      <end-point-path-pair-list>
                                      [<attribute-list>]

<PCE-initiated-lsp-deletion> ::= <SRP>
                                 <LSP>

Where:

<end-point-path-pair-list>::=
                  [<END-POINTS>]
                  <path>
                  [<end-point-path-pair-list>]

<path> ::= (<ERO>|<SERO>)
           [<path>]
```

# Messages

```
<PCReq Message>::= <Common Header>
                   <request>
where:
<request>::= <RP>
             <end-point-rro-pair-list>
             [<LSP>]
             [<OF>]
             [<LSPA>]
             [<BANDWIDTH>]
             [<metric-list>]
             [<IRO>]
             [<LOAD-BALANCING>]

where:
<end-point-rro-pair-list>::=<END-POINTS>[<RRO-List>][<BANDWIDTH>]
                            [<end-point-rro-pair-list>]

<RRO-List>::=(<RRO>|<SRRO>)[<BANDWIDTH>][<RRO-List>]
<metric-list>::=<METRIC>[<metric-list>]
```

```
<PCRep Message>::= <Common Header>
                   <response>

<response>::=<RP>
             [<end-point-path-pair-list>]
             [<NO-PATH>]
             [<attribute-list>]

where:

<end-point-path-pair-list>::=
                [<END-POINTS>]<path>[<end-point-path-pair-list>]

<path> ::= (<ERO>|<SERO>) [<path>]

<attribute-list>::=[<LSP>]
                   [<OF>]
                   [<LSPA>]
                   [<BANDWIDTH>]
                   [<metric-list>]
                   [<IRO>]
```

# Thanks!