Stateful PCE for P2MP LSP

draft-palle-pce-stateful-pce-p2mp-02

draft-palle-pce-stateful-pce-initiated-p2mp-lsp-01

Udayasree Palle (Huawei)	Dhruv Dhody (Huawei)
Yosuke Tanaka (NTT)	Yuji Kamite (NTT)

Motivation

Applicability

- [I-D.ietf-pce-stateful-pce-app] presents several use cases, demonstrating scenarios that benefit from the deployment of a stateful PCE including optimization, recovery, etc.
  - These scenarios apply equally to P2P and P2MP TE LSPs.
- When P2MP TE LSP placement needs to change in response to application demands, it is useful to support dynamic creation and tear down of P2MP TE LSPs via PCE.

Message Encoding

- For stateless PCE, any modification of P2MP tree requires encoding of all leaves along with the paths in PCReq message.
- Stateful PCE with P2MP capability, convey only the modifications (the other information can be retrieved from the PLSP ID) - *An added advantage!*

Protocol Extension

Capability Advertisement

- Stateful PCE Capability TLV [I-D.ietf-pce-stateful-pce]
- P2MP capable TLV [RFC6006]
- When both TLVs are used together - indicates a stateful PCE with P2MP capability.
- *Also via IGP auto discovery*

LSP Object

- New Flags ¨C 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

S2L (Source to Leaf)

- 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.

No change in operations (from P2P)

- LSP state synchronization
- LSP delegation
- LSP update

PCEP Message Extension

```
<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>]
                   <S2L>
                   <path>
                   [<end-point-path-pair-list>]

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

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 S2L 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.

PCE Initiated P2MP LSP

Capability Advertisement

- Stateful PCE Capability TLV (I bit)
- P2MP capable TLV
- When used together indicates a stateful PCE with P2MP instantiation capability

P2MP LSP Instantiation

- P2MP (N bit)
- Create (C bit)
- When used together indicate PCE-Initiated P2MP LSP

Add/Prune leaves

- PCUpd message with leaf type = 1 for adding of new leaves
- leaf type = 2 for pruning of old leaves

```
<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>]
```

No change in operations (from P2P)

- LSP instantiation
- LSP deletion
- LSP delegation and cleanup

Message Fragmentation

P2MP PCRpt, PCUpd and PCIntiate may not fit into a single PCEP message.

The new F-bit is used in the LSP object to signal that it was too large to fit into a single message and will be fragmented into multiple messages.

Each message except the last one, will have the F-bit set in the LSP object to signify it has been fragmented into multiple messages.

Should use the same PLSP-ID and SRP-ID-number for all fragmented message.

Request to [I-D.ietf-pce-stateful-pce]

Passive Stateful PCE

- The Request and Response message should support LSP object, so that it is possible to refer to a LSP with a unique identifier and simplify the PCEP message exchange.
- For example, incase of modification of one leaf in a P2MP tree, there should be no need to carry the full P2MP tree in PCReq message.

END-POINTS object (optional) in PCRpt

- For P2MP this is needed, adding for P2P as well (?)

Questions & Comments?

Thanks!