Stateful PCE for P2MP LSP

draft-palle-pce-stateful-pce-p2mp-08
draft-palle-pce-stateful-pce-initiated-p2mp-lsp-07

<table>
<thead>
<tr>
<th>Udayasree Palle (Huawei)</th>
<th>Dhruv Dhody (Huawei)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yosuke Tanaka (NTT)</td>
<td>Zafar Ali (Cisco)</td>
</tr>
<tr>
<td>Vishnu Pavan Beeram (Juniper)</td>
<td></td>
</tr>
</tbody>
</table>

6th April 2016

95th IETF @ BA
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
  - 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 PCIInitiate 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

<table>
<thead>
<tr>
<th>Addition of Stateful P2MP PCE capability in IGP</th>
<th>Update in PCRpt/PCUpd/PcInitiate/PCReq message format</th>
</tr>
</thead>
<tbody>
<tr>
<td>• SERO / SRRO</td>
<td>• Intended and actual path</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Error Handling</th>
</tr>
</thead>
<tbody>
<tr>
<td>• S2LS / ENDPOINT object missing</td>
</tr>
<tr>
<td>• Fragmentation error</td>
</tr>
<tr>
<td>• Backward Compatibility</td>
</tr>
</tbody>
</table>
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
The format of PCRpt message is as follows:

\[
\text{<PCRpt Message>} ::= \langle \text{Common Header} \rangle \\
\langle \text{state-report-list} \rangle
\]

Where:

\[
\text{<state-report-list>} ::= \langle \text{state-report} \rangle \\
[\langle \text{state-report-list} \rangle]
\]

\[
\text{<state-report>} ::= [\langle \text{SRP} \rangle] \\
\langle \text{LSP} \rangle \\
\text{<end-point-path-pair-list>}
\]

Where:

\[
\text{<end-point-path-pair-list>} ::= [\langle \text{END-POINTS} \rangle] \\
[\langle \text{S2LS} \rangle] \\
\text{<intended_path>} \\
[\langle \text{actual_path} \rangle] \\
[\langle \text{end-point-path-pair-list} \rangle]
\]

\[
\text{<intended_path>} ::= (\langle \text{ERO} \rangle|\langle \text{SERO} \rangle) \\
[\langle \text{intended_path} \rangle]
\]

\[
\text{<actual_path>} ::= (\langle \text{RRO} \rangle|\langle \text{SRRO} \rangle) \\
[\langle \text{actual_path} \rangle]
\]

\[
\text{<attribute-list>}
\]

Where:

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

\[
\text{<PCUpd Message>} ::= \langle \text{Common Header} \rangle \\
\langle \text{update-request-list} \rangle
\]

Where:

\[
\text{<update-request-list>} ::= \langle \text{update-request} \rangle \\
[\langle \text{update-request-list} \rangle]
\]

\[
\text{<update-request>} ::= \langle \text{SRP} \rangle \\
\langle \text{LSP} \rangle \\
\text{<end-point-path-pair-list>}
\]

\[
\text{<attribute-list>}
\]

Where:

\[
\text{<attribute-list> is defined in [RFC5440] and extended by PCEP extensions.}
\]
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>]
        [<OP>]
        [<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>
where:
<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>]
    [<OP>]
    [<LSPA>]
    [BANDWIDTH>]
    [metric-list>]
    [IRO>]

Thanks!