MOP Extension & Capabilities

draft-rahul-roll-mop-ext-01
- Rahul, Pascal @ IETF105, Montreal
Need of MOP-extension?

• Mode of Operation (MOP)
  - Mandates primitives to be supported by the 6LRs
  - 3-bits in size
  - Already exhausted

<table>
<thead>
<tr>
<th>MOP</th>
<th>Used for</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>No downward routes</td>
</tr>
<tr>
<td>1</td>
<td>Non-storing</td>
</tr>
<tr>
<td>2</td>
<td>Storing with no mcast</td>
</tr>
<tr>
<td>3</td>
<td>Storing with mcast</td>
</tr>
<tr>
<td>4</td>
<td>P2P-RPL</td>
</tr>
<tr>
<td>5, 6, 7 (Unused)</td>
<td>(AODV-RPL, P-DAO-NS, P-DAO-Storing)</td>
</tr>
</tbody>
</table>
MOP Extension

- MOPex Option
  - New RPL Control message option
  - Applicable only if base DIO-MOP = 0x7
  - Final MOP = base MOP + MOPex

<table>
<thead>
<tr>
<th>Base MOP</th>
<th>MOPex</th>
<th>Final MOP</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>NA</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>NA</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>:</td>
<td>:</td>
</tr>
<tr>
<td>6</td>
<td>NA</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>7</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>8</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>:</td>
<td>:</td>
</tr>
</tbody>
</table>

Table 1: Final MOP calculation

Figure 1: Extended MOP Option
Introducing Capabilities

- Capabilities indicate the set of features supported
  - Could be mandatory or *optional*
    * Specs defining new capability indicate whether it is mandatory/optional

- Why MOP is not sufficient?
  - MOP *mandates* primitives needed by the routers
  - Unlike MOP, Capabilities can be *negotiated*,
    * using DIO/DAO/DAO-ACK
Capabilities (Caps) Option

- Defined as new RPL Control message option
  - Can be part of DIO/DAO/ACK

```
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1

+-----------------------------------------+
| Type = TODO | Capabilities Flags |
+-----------------------------------------+

Figure 2: Capabilities Option
```
Use-case for Root

• Used by root
  – In DIO: Inform all the 6LR/6LN of root’s capabilities
  – In DAO-ACK: Inform the 6LR/6LN about accepted capabilities from Root.
Use-case for 6LR/6LN

- **Used by 6LR/6LN**
  - In DIO: Inform its child nodes about its capabilities (for 6LRs)
  - In DAO: Inform parent/ancestors/root of this 6LR’s capabilities
Points to ponder

• CAPs can work with existing MOPs
  - CAPs and MOPs are not dependent on each other

• Reducing CAPs control overhead
  - Eliding mechanism similar to DIO Configuration Option?

• How to reduce control overhead of MOPex?
Turnon-6LoRH, a use-case

- Can be handled without changing RFC 6550?
  - Note that this is not what is suggested in the draft, currently.
ACK

• Thanks to Georgious for the review
• Updates
  – Clarification: what if MOPex option is absent but the base MOP is 7.
  – Made explicit: CAP and MOPex are mutually exclusive
  – Added detailed IANA considerations