T3P or Modified LLDP?

Dec 08, 2009

Manoj Wadekar, QLogic Corp.
Ilango Ganga, Intel Corp.
Goals of presentation

• Understand differences between LLDP and VDP/T3P requirements
• Explore LLDP modifications that can support T3P enhancements
• Not a final proposal – but proposed direction
Steps for Configuring Edge Connections (VSIs)

1. Query available port profile types and obtain a port profile instance.
2. Push VM & VSI Configuration to VM Host.
3. Retrieve Port Configuration.
4. Switch Edge.

System Admin

Network Admin

VM Manager

Port Profile Database

VM Edge

Switch

VSI Discovery

L2 net(s)

Server Edge
Need for VDP/T3P

- EVB allows bridge to have “insight” into virtual server
- “Insight” results into information about VSIs supported inside station
- Bridge needs to maintain per VSI information
  - VDP to allow exchange of such information
  - T3P to allow transport of such information between two directly connected nodes
Revisiting VDP/T3P needs
(Ref: v0.1 presentation on 11/23/09)

<table>
<thead>
<tr>
<th>LLDP</th>
<th>VDP/T3P</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requires ALL TLVs in every LLDP exchange. Limited to 1500 bytes, no fragmentation.</td>
<td>Assumes subsets of TLVs can be passed in a frame. Total size of TLVs can significantly exceed 1500 bytes.</td>
</tr>
<tr>
<td>Assumes that information is exchanged (no state maintained).</td>
<td>Assumes that TLV exchanges can be stateful and that state can be maintained for each TLV type.</td>
</tr>
<tr>
<td>Delivery is confirmed by regularly retransmitting (and fast retransmit) the full set of information.</td>
<td>Successful delivery is confirmed by other approaches: • TLV-specific ACKs/NACKs may not be required • Transmission of a signature of the last TLV sent or of current state may not be required. • Etc.</td>
</tr>
<tr>
<td>Requires repetition of information that is not changing.</td>
<td>Only transmits TLVs that are new or changed.</td>
</tr>
</tbody>
</table>

Can these requirements be addressed with LLDP modifications?
**Few thoughts:**
**LLDP Modifications per EVB requirement**

<table>
<thead>
<tr>
<th>EVB Requirements</th>
<th>LLDP Modifications</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total size of TLVs can significantly exceed 1500 bytes.</td>
<td>Allow “LLDP Client” to submit multiple TLVs. Modified LLDP to transmit 1500B LLDPDU. Multiple LLDPDUs to be generated to exhaust all “LLDP Client” TLvs</td>
</tr>
<tr>
<td>Assumes that TLV exchanges can be stateful and that state can be maintained for each TLV type.</td>
<td>State to be maintained in “LLDP Client” e.g. VDP. No state is required in LLDP.</td>
</tr>
<tr>
<td>Successful delivery is confirmed by other approaches:</td>
<td>“Reliability” requirement can be moved to “LLDP Clients” while maintaining “Modified LLDP” simple. Is there strong reason to introduce ACK?</td>
</tr>
<tr>
<td>• TLV-specific ACKs/NACKs may not be required</td>
<td>• Transmission of a signature of the last TLV sent or of current state may not be required.</td>
</tr>
<tr>
<td>• Etc.</td>
<td>• Etc.</td>
</tr>
<tr>
<td>Only transmits TLVs that are new or changed.</td>
<td>Allow “LLDP Client” to submit TLVs only for new information.</td>
</tr>
</tbody>
</table>
Modified LLDP (single direction shown)

Queue up as many desired VSI changes as needed

1. LLDP state machine creates a PDU with Required TLVs and VDP TLVs (or LLDP Client TLVs). If all TLVs cannot fit into 1500B, then LLDP creates multiple LLDPDUs.

2. LLDP supports an “immediate PUSH” in case of MIB change. LLDP Timer is used.

3. LLDP delivers TLVs to appropriate “LLDP Clients”
Proposal in Nut Shell

• Reuse LLDP Framework and Protocol
• Explore modification to allow LLDP Clients to submit TLVs with information >1500B
  - LLDP to generate multiple LLDPDUs to exhaust submitted TLVs (each LLDPDU still 1500B)
  - Allow avoiding repetition of “Other TLVs” or “System TLVs” in each LLDPDU
    • Can be achieved by treating “Other TVLs” same as LLDP Client that queues one request per timer or per change
• Explore further “Reliability Needs”