## Method For Discovering Network Topology

**SWRA203 Version 1.4**

<table>
<thead>
<tr>
<th>Version</th>
<th>Description</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>Initial Release.</td>
<td>09/15/2005</td>
</tr>
<tr>
<td>1.1</td>
<td>Changed logo on title page, changed copyright on page footer.</td>
<td>02/27/2006</td>
</tr>
<tr>
<td>1.2</td>
<td>Updated for 1.4.0 Release</td>
<td>12/14/2006</td>
</tr>
<tr>
<td>1.3</td>
<td>Updated for 2.1.0 Release</td>
<td>06/16/2008</td>
</tr>
<tr>
<td>1.4</td>
<td>Updated for 2.2.0 Release</td>
<td>04/03/2009</td>
</tr>
</tbody>
</table>
Table of Contents

1. PURPOSE.........................................................................................................................1
2. METHOD............................................................................................................................1
3. EXAMPLE USING Z-TOOL ............................................................................................1
4. SEQUENCE DIAGRAMS .................................................................................................5
5. PACKET DETAILS ...........................................................................................................6
1. Purpose

This document describes a method for determining the network topology for any given network using a subset of the Z-Stack ZDO (Zigbee Device Object) layer APIs. Please read the ZStack Developer’s Guide and the ZStack API document before continuing.

2. Method

By using `ZDP_IEEEAddrReq()`, one can start the process by issuing this request to the PAN coordinator since it always has short address 0x0000. The `ReqType` parameter should be set to 1, to obtain the list of devices that have associated to the coordinator. The `StartIndex` parameter can be used if the list contains more devices that can fit into one message.

By registering for the IEEE address request callback at the application level (see Z-stack Developer’s Guide section on ZDO Message Requests), one can obtain the list of associated devices which will include their short addresses, as well as the IEEE address of the coordinator. The `NumAssocDev` parameter in the IEEE address response structure represents the number of devices that are associated with the coordinator.

Once the first list from the coordinator is returned, one can repeat the above procedure targeted at each short address in the list. By iterating this procedure over all parents and children, the application will be able to “map out” the topology of the entire network.

3. Example using Z-Tool

In this example, the contrived network shown in Figure 1 was created, and visualized using the Daintree Sensor Network Analyzer application. The `NLME_PermitJoiningRequest()` function was used on the coordinator device to force the formation of the network in this manner. By passing in the parameter of 0x00 to `NLME_PermiitJoiningRequest`, the user can deny other devices from associating to a particular router or coordinator. Passing in a parameter of 0xFF permits association.

NOTE: The use of NLME_PermitJoiningRequest in this example is used to force a joining topology and is not the normal use of this function. The NLME_PermitJoiningRequest function is used to turn off associations in the coordinator or any router and should be used to allow and disallow new devices joining the network.

Sections 4 and 5 contain information about the packet flow and the information included in a ZDO IEEE Address Response packet.
The method described above in Section 2 can be demonstrated by exercising the ZDO MT (Monitor Test) functions using Z-Tool (for Z-Tool help, select Help in the Z-Tool menu). These are compiled in via the MT_ZDO_FUNC and MT_ZDO_CB_FUNC compile options (see the ZStack Compile Options document for further details).
Figure 2 shows a screen capture of an instance of Z-Tool connected to the coordinator. One should issue the ZDO_IEEE_ADDR_REQ with the ShortAddr parameter set to 0, ReqType set to EXTENDED, StartIndex set to 0. The response is then displayed in Z-Tool, and we see that the coordinator has one device associated to it with short address = 0xFAA6 (AssocDevList: 0xFAA6).

ZDO_IEEE_ADDR_REQ is then issued to short address 0xFAA6, and we see that the router has one device associated to it with short address = 0xB922 (see Figure 3).
ZDO_IEEE_ADDR_REQ can then also be sent to short address 0xB922, and the user will see that no devices are associated with it. Thus, this is the end of the line, and the device discovery procedure is concluded.
4. Sequence Diagrams

Figure 4. Sequence Diagram for ZDO_IEEE_ADDR_REQ and ZDO_IEEE_ADDR_RSP

Figure 4 shows the chain of ZDO_IEEE_ADDR_REQ calls on the left hand side of the device tree, and corresponding ZDO_IEEE_ADDR_RSP calls on the right hand side.
5. Packet Details

Figure 5 shows the over-the-air packets that occur during these transactions, and Figure 6 shows a snapshot of one of the ZDO IEEE Address Response packets that is decoded by Daintree to show the associated device list of Router 0xFAA6.

Figure 5. Over-the-air Traffic

Figure 6. Router 0xFAA6’s Associated Device List