AN4815

Zigbee® Packet Analysis

Introduction

This application note provides details about how to configure and use supported sniffing tools along with Microchip MCU-based sniffer hardware platforms. In Zigbee® networking, a sniffing tool (for example, Wireshark Network Protocol Analyzer (Wireshark)) is important during the development and testing phase to capture and analyze the frames exchanged in the network. Wireshark is more significant in networks with the Zigbee products from different vendors to test and verify as they are interoperable with one another. This application note mainly focuses on the packet capture using the Wireshark.

Wireshark is a free and open-source packet analyzer. The wireless network sniffer environment is set up by running the Wireshark on the PC. The following are the uses of the Wireshark:

- Network troubleshooting
- Analysis
- Software and communications protocol development
- Education

The Wireshark Sniffer Interface Tool connects the Wireshark Graphical User Interface (GUI) and the sniffer firmware running on the ZigBit USB stick. It enables communication between the Wireshark Sniffer Interface Tool application running on the PC and sniffer hardware. The Wireshark Sniffer Interface Tool is capable of real-time capture of frame formats supported by the Zigbee protocol and the IEEE® 802.15.4 standard. It also provides parsed information of different fields and sub-fields of the frame that helps the user in quick analysis.

Features

- Network Topology
- Time Stamping
- Multi-Channel Capture

Table of Contents

1. Quick References

1.1 Reference Documentation

For further details, refer to the following:

- *AT08550: ZigBee Attribute Reporting Application Note* (42334)
- *Atmel AT02597: ZigBee PRO Packet Analysis with Sniffer Application Note* (32210)
- *Atmel-ICE Programmers and Debuggers User Guide* (42330)
- *PRO Base Device Behavior Specification* (3.0.1)
- *ZigBee Alliance Cluster Library Specification Revision 8* (075123)
- *Matter Device Library Specification* (1.0)
- *Zigbee PRO Green Power feature specification Basic functionality set* (Version 1.1.1)
- *Zigbee Specification Revision 22 1.0* (05-3474-22)
- *ZigBit USB Stick User Guide* (42194)

1.2 Hardware Requirements

- 50-mil 10-pin IDC flat cable
- ATMEL-ICE (ATATMEL-ICE)
- ATXMEGA256A3U and AT86RF212B ZIGBIT USB Stick (ATZB-X-212B-USB)
- ATXMEGA256A3 and AT86RF233 ZIGBIT USB Stick (ATZB-X-233-USB)
- Micro-AB USB cable

1.3 Software Requirements

- Microchip Studio (7.0.2594)
- Windows 10
- Wireshark (3.6.8)
- Wireshark Sniffer Interface Tool (v3.0.0.10)

1.4 Acronyms and Abbreviations

Table 1-1. Acronyms and Abbreviations

Acronyms and Abbreviations	Description
API	Application Programming Interface
APL	Application Layer
APS	Application Support Sub-Layer
BDB	Base Device Behavior
GP	Green Power
GPC	Green Power Combo
GPD	Green Power Device
GPP	Green Power Proxy
GPS	Green Power Sink

.....continued

Acronyms and Abbreviations	Description
GUI	Graphical User Interface
NWK	Network
PAN	Personal Area Network
USB	Universal Serial Bus
ZCL	Zigbee® Cluster Library
ZDO	Zigbee Device Object
ZDP	Zigbee Device Profile

2. Wireshark Network Protocol Analyzer and Wireshark Sniffer Interface Tool Overview

This chapter provides an overview of the Wireshark Network Protocol Analyzer (Wireshark) and Wireshark Sniffer Interface Tool setup along with their respective components. By default, the Wireshark and Wireshark Sniffer Interface Tool installs the package in the *C:\Program Files*\ and *C:\Program Files (x86)*\, respectively.

Table 2-1. Wireshark Package Files

File Name	Description
`Wireshark-winXX-3.X.X.exe` file	Wireshark executable file

Table 2-2. Wireshark Sniffer Interface Tool Package Files

File Name/Folder Name		Description
`Wireshark Sniffer Interface Tool v3.0.0.10.exe`		Wireshark Sniffer Interface Tool executable file
C:\Program Files (x86)\Atmel\Atmel Wireshark Sniffer Interface Tool Folder	Atmel Wireshark Sniffer Firmware	—
	`Atmel_Wireshark_Sniffer_Interface.exe`	Wireshark Sniffer Interface tool executable file
	`Atmel_Wireshark_Sniffer_Interface.exe.config`	Atmel Wireshark Sniffer Interface framework configuration file
	`Release Notes.txt`	• Contains release and version information for the Wireshark Sniffer Interface Tool • To capture/sniff IEEE® 802.15.4 frames (2.4 GHz and Sub-GHz)
	`System.Xaml.dll`	—
C:\Program Files (x86)\Atmel\Atmel Wireshark Sniffer Interface Tool\Atmel Wireshark Sniffer Firmware Folder	`AWSI_at32uc3a3256s_rz600_at86rf212.hex`	Sniffer firmware for RZ600 USB stick. For more details, refer to the ATAVRRZ600.
	`AWSI_at32uc3a3256s_rz600_at86rf231.hex`	
	`AWSI_atxmega256a3u_rf212b_zigbit_usb.hex`	ZigBit USB stick firmware for Sub-GHz
	`AWSI_atxmega256a3u_rf233_zigbit_usb.hex`	ZigBit USB stick firmware for 2.4 GHz sniffers

2.1 Supported Sniffer Hardware Platforms

To start with capturing frames on an IEEE 802.15.4 channel, the user must have a sniffer hardware tool running a sniffer firmware plugged into the PC. The following are the supported sniffer hardware platforms:

- RF212B ZigBit USB stick – For sniffing IEEE 802.15.4 Sub-GHz channels
- RF233 ZigBit USB stick – For sniffing IEEE 802.15.4 2.4 GHz channels

Figure 2-1. Supported Sniffer Hardware Platforms – ZigBit USB Stick (RF212B/RF233 – Sub-GHz/2.4 GHz)

Use the Wireshark Sniffer Interface Tool to create capture sessions for IEEE 802.15.4 channels in 2.4 GHz and Sub-GHz range. The Wireshark Sniffer Interface Tool supports ATXMEGA256A3U_RF212B and ATXMEGA256A3U_RF233.

2.2 Getting Started with Wireshark Tool and Wireshark Sniffer Interface Tool

2.2.1 Wireshark Installation Procedure

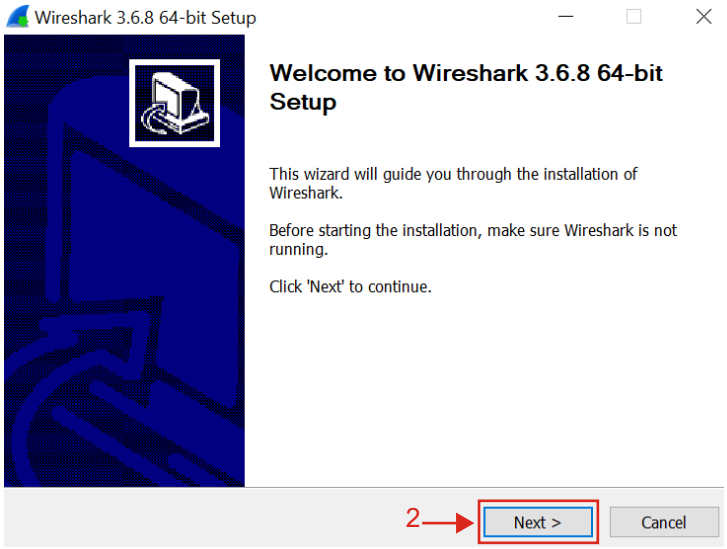
For downloading Wireshark, go to Wireshark. Fore more details about Wireshark, go to Wireshark.

Note: The stable release version of the Wireshark is version 3.6.6, or the user can also install the latest development release available from the official Wireshark website on the PC.

The following are the steps to install the Wireshark:

1. Double click the `Wireshark-winXX-3.X.X.exe` to start the installation procedure.
2. Click **Next** to continue.

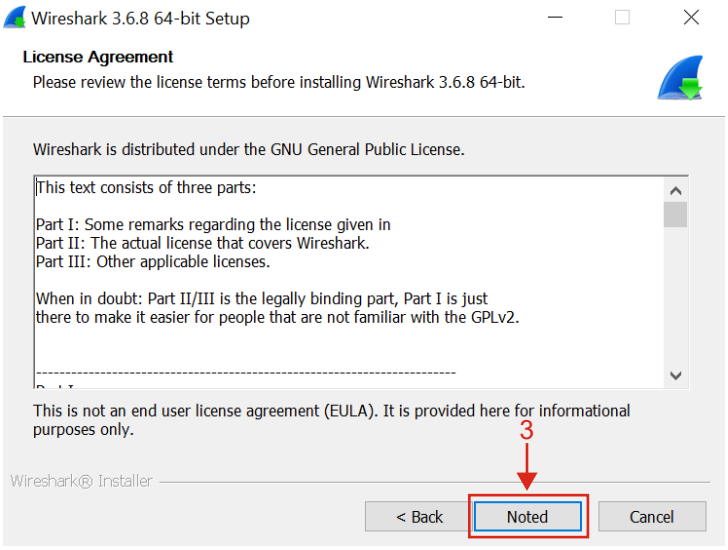
Figure 2-2. Wireshark Setup Window



Note: The user can use the latest version of Wireshark available.

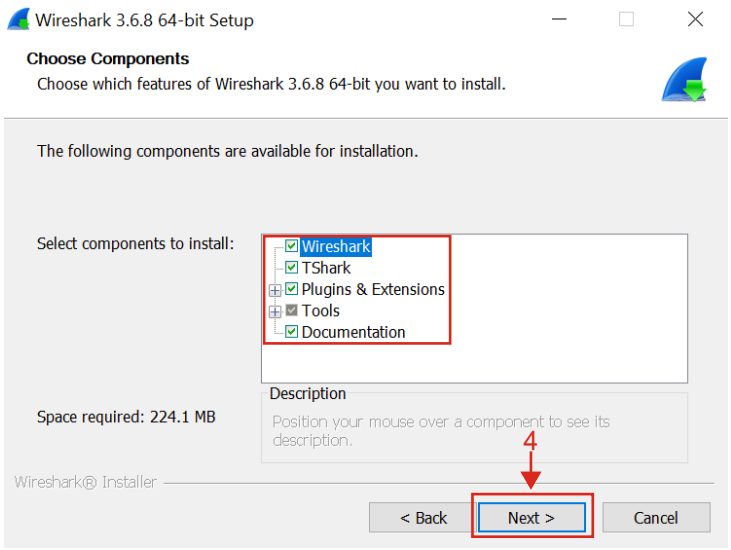
3. Click **Noted** to continue.

Figure 2-3. Wireshark – License Agreement



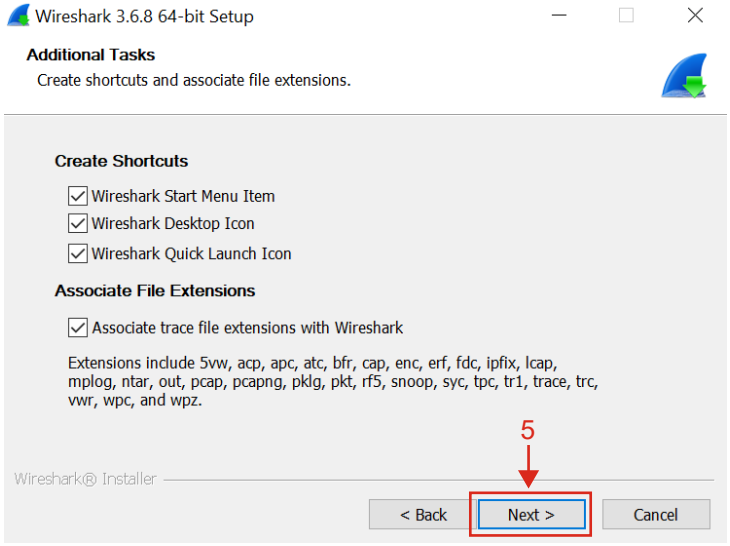
4. Under the “Select components to install:” field, check the respective components to install along with the tool. Click **Next** to continue.

Figure 2-4. Wireshark – Choose Components



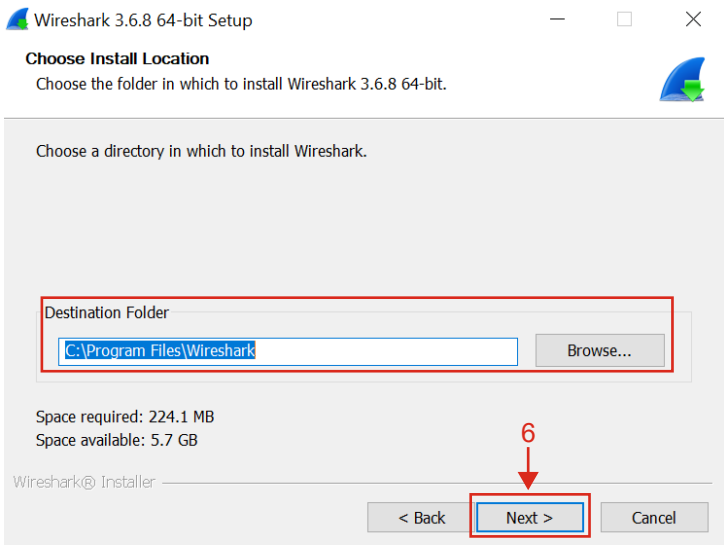
5. Under the “**Create Shortcuts**” field, check the required shortcuts, and under the “**Associate File Extensions**” field, check *Associate trace file extensions with Wireshark*. Click **Next** to continue.

Figure 2-5. Wireshark – Additional Tasks



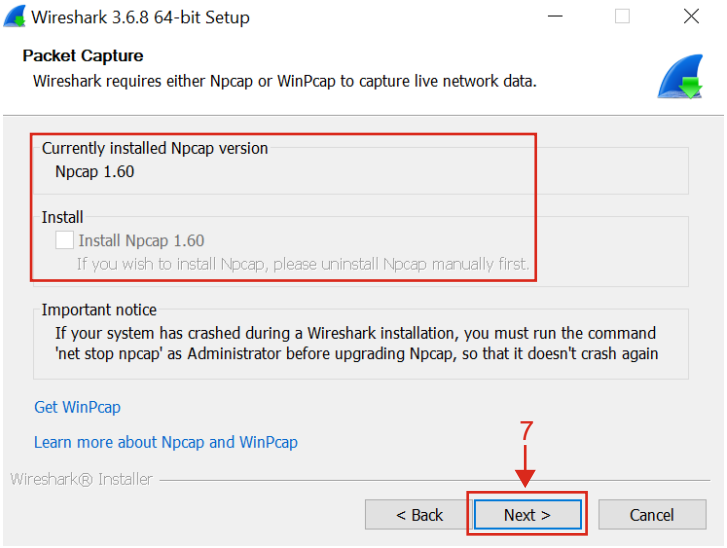
6. Install the Wireshark in the default location under the “Destination Folder” field: *C:\Program Files\Wireshark* . Click **Next** to continue.

Figure 2-6. Wireshark – Install Location



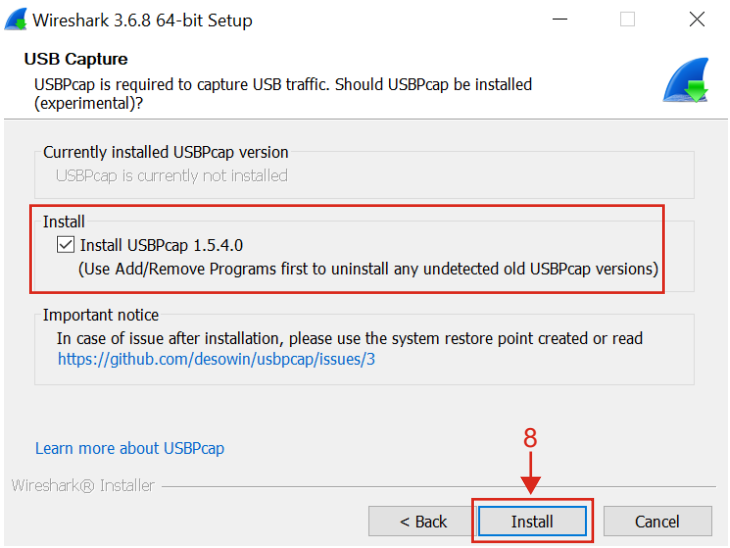
7. Install “Npcap” or “WinPcap” to capture live network data. In this scenario, it is *Npcap 1.60*. Click **Next** to continue. (Optional)

Figure 2-7. Wireshark – Packet Capture



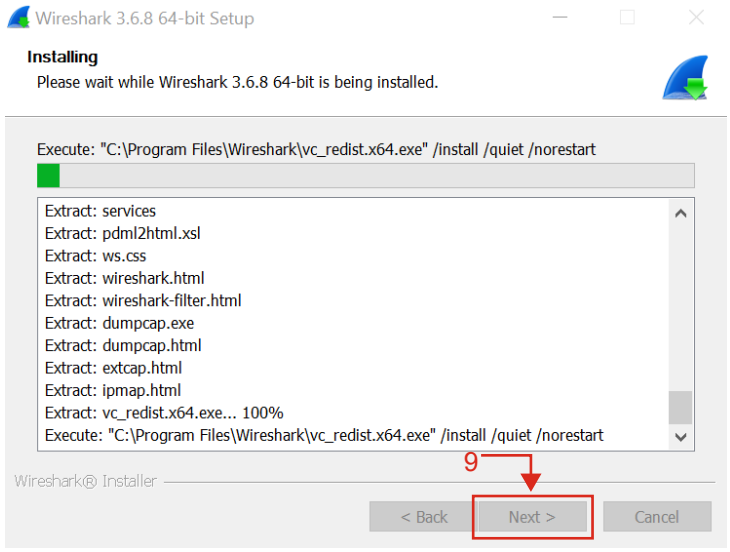
8. Install “USBPcap” to capture USB traffic. In this scenario, under the “Install” field, check *Install USBPcap 1.5.4.0*. Click **Install** to continue. (Optional)

Figure 2-8. Wireshark – USB Capture



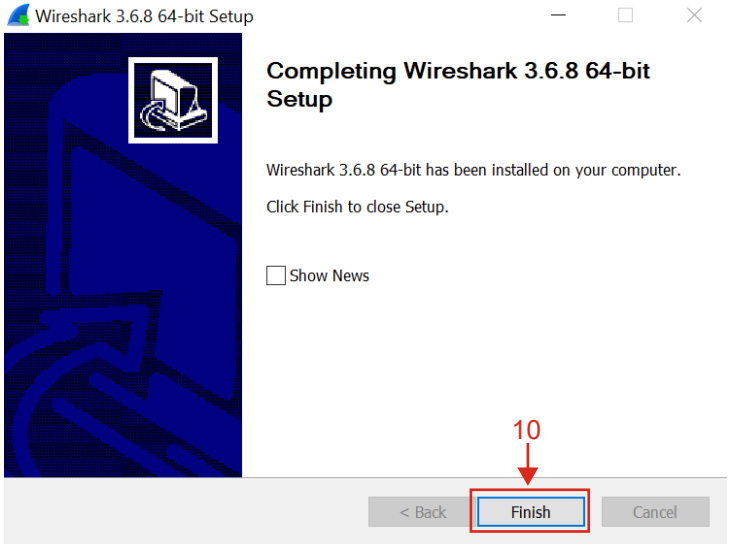
9. The user must wait for the completion of the Wireshark installation procedure. After the installation procedure completes, click **Next** to continue.

Figure 2-9. Wireshark – Installing



10. Click **Finish** to complete the Wirehsark setup.

Figure 2-10. Wirehark Completing Setup



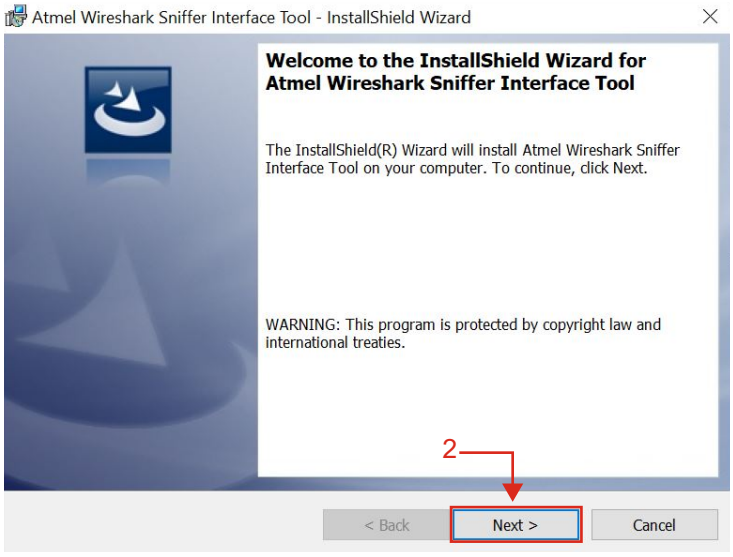
2.2.2 Wireshark Sniffer Interface Tool Installation

The user must install the Wireshark Sniffer Interface Tool to set up a capture session using Wireshark. For downloading the Wireshark Sniffer Interface Tool, go to Wireshark Sniffer Interface Tool v3.0.0.10.

The following are the steps to install the Wireshark Sniffer Interface Tool:

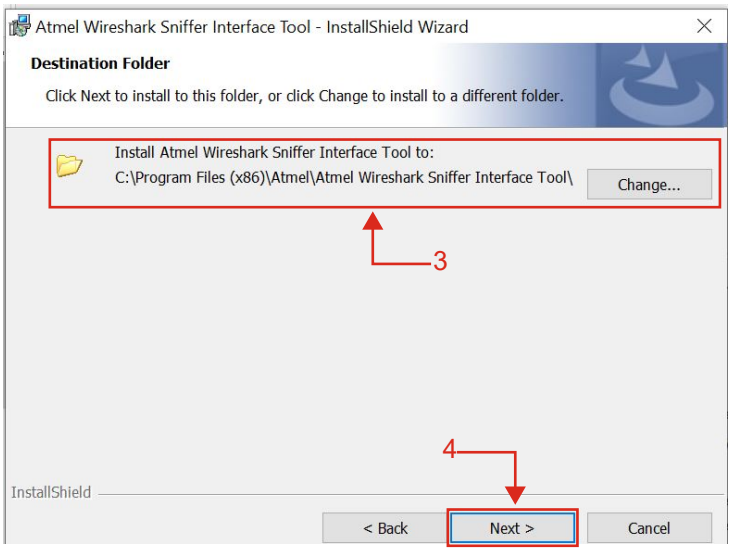
1. Double click the `Atmel Wireshark Sniffer Interface Tool.exe` to start the installation procedure.
2. Click **Next** to continue.

Figure 2-11. Wireshark Sniffer Interface Tool InstallShield Wizard



3. Install the Wireshark Sniffer Interface Tool in the default location, *C:\Program Files (x86)\Atmel\Atmel Wireshark Sniffer Interface Tool\Atmel Wireshark Sniffer Firmware*.
4. Click **Next** to complete the installation.

Figure 2-12. Default Location – Wireshark Sniffer Tool Installation



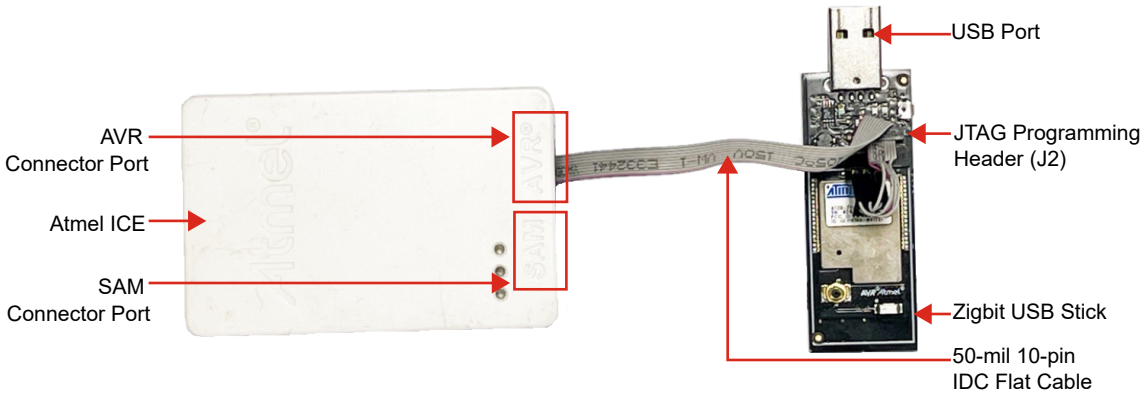
5. Follow the instructions in `Release Notes.txt` (available inside the package folder `Atmel Wireshark Sniffer Interface Tool`) to complete the setup procedure. For more details, refer to the *ZigBit USB Stick User Guide* (42194).
6. Flash the sniffer firmware on the respective Zigbit hardware platforms. For more details, refer to the 2.3. Flashing the Firmware into ZigBit USB Stick. The following are the available images in the package:
   - `AWSI_at32uc3a3256s_rz600_at86rf212.hex`
   - `AWSI_at32uc3a3256s_rz600_at86rf231.hex`
   - `AWSI_atxmega256a3u_rf212b_zigbit_usb.hex`
   - `AWSI_atxmega256a3u_rf233_zigbit_usb.hex`

2.3 Flashing the Firmware into ZigBit USB Stick

The following are the steps to flash the firmware into the ZigBit USB stick:

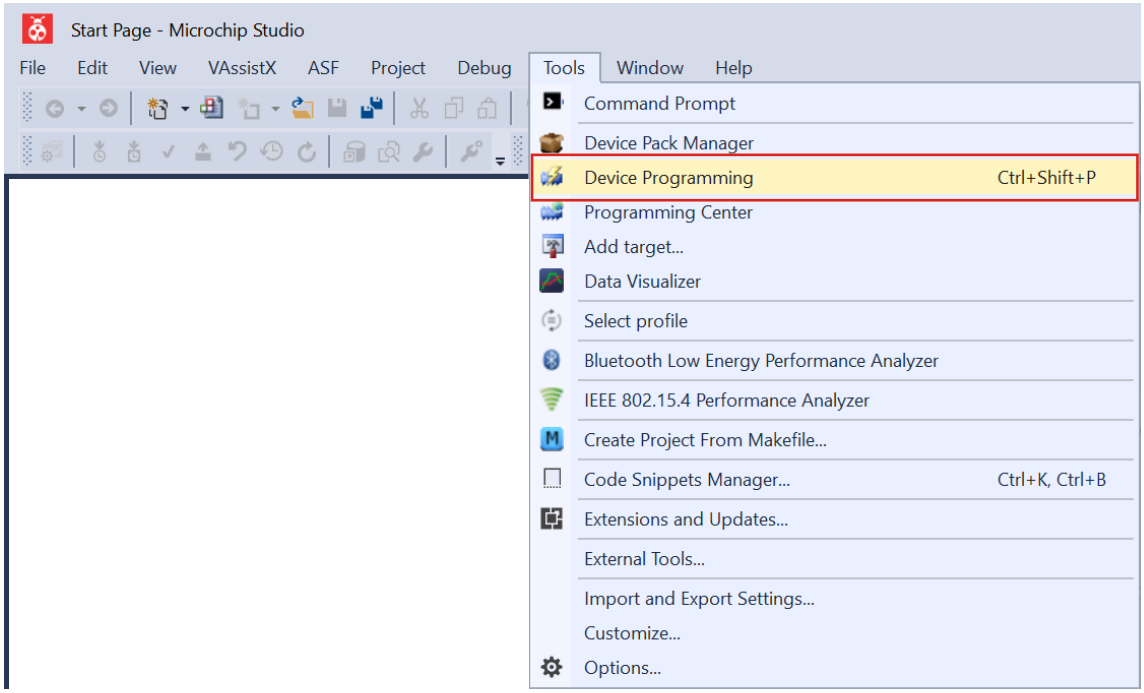
1. Connect the Atmel ICE JTAG cable from the AVR® connector port in Atmel ICE to JTAG programming header (J2). For more details, refer to the *ZigBit USB Stick User Guide* (42194).
2. Connect the Atmel ICE to one COM port of PC using the USB cable and ZigBit USB stick to another COM port of the PC. For more details, refer to the *Atmel-ICE Programmers and Debuggers User Guide* (42330).

Figure 2-13. Atmel ICE Zigbit Sniffer Connection



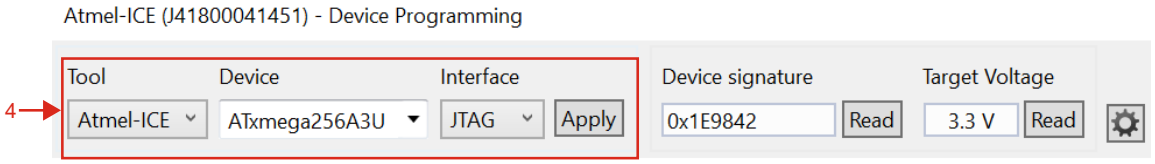
3. Open the Microchip Studio, go to *Tools>Device Programming* and select the corresponding tools, devices and interfaces.

Figure 2-14. Device Programming



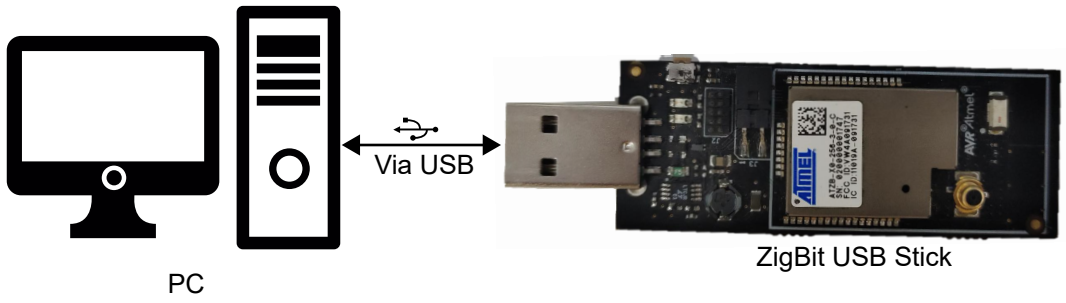
4. The user must select the following fields:
   - From the “Tool” drop-down list, select *Atmel-ICE*.
   - From the “Device” drop-down list, select *ATxmega256A3U*.
   - From the “Interface” drop-down list, select *JTAG*.

Figure 2-15. Device Programming Fields



5. The firmware images are available in the directory: *\Atmel Wireshark Sniffer Interface Tool\Atmel Wireshark Sniffer Firmware*. Load Wireshark sniffer firmware from the default location (step 3), and flash the firmware into the ZigBit sniffer.
6. Disconnect the Atmel ICE from the ZigBit USB stick.
7. Connect the ZigBit USB stick to the PC via USB, and open the Atmel Wireshark Sniffer Interface Tool.

Figure 2-16. Connect ZigBee USB Stick to PC



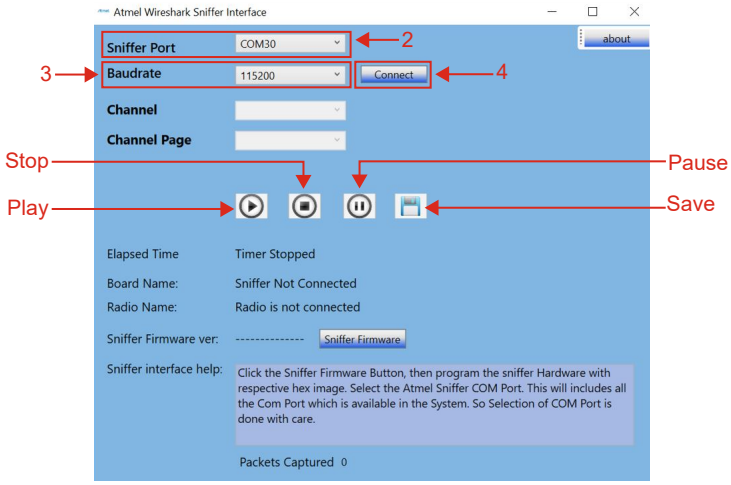
3. Sniffer Capture Session Setup

This chapter provides details about how to set up a sniffer capture session after successful installation of the Wireshark Sniffer Interface Tool in the PC.

3.1 Wireshark Packet Capture Procedure

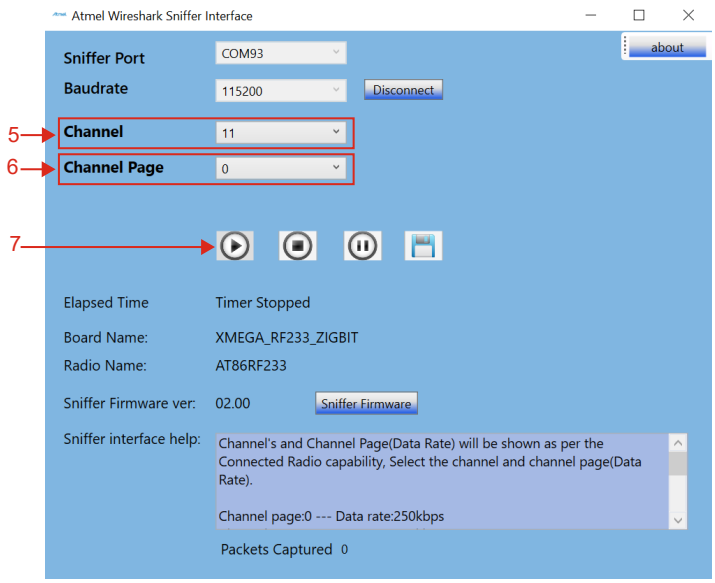
1. From the start menu, click *Atmel Wireshark Sniffer Interface Tool* to launch the Wireshark Sniffer Interface Tool.
2. From the “Sniffer Port” drop-down list, for example, select *COM30*.
3. From the “Baudrate” drop-down list, select *115200*.
4. Click **Connect** to continue.

Figure 3-1. Start-Up Window Atmel Wireshark Sniffer Interface Tool



5. From the “Channel” drop-down list, for example, select *11*.
   Notes: The user can select the “Channel” as per the requirement. The following are the values:
   - For Sub-GHz, the range is 0-10.
   - For 2.4 GHz, the range is 11-26.
6. From the “Channel Page” drop-down list, for example, select *0*. The range is 0-10.
   Note: The user can tune the “Channel Page” according to the custom data rate requirements.
7. Click **Play** to start capture.

Figure 3-2. Channel/Channel Page/Play Button

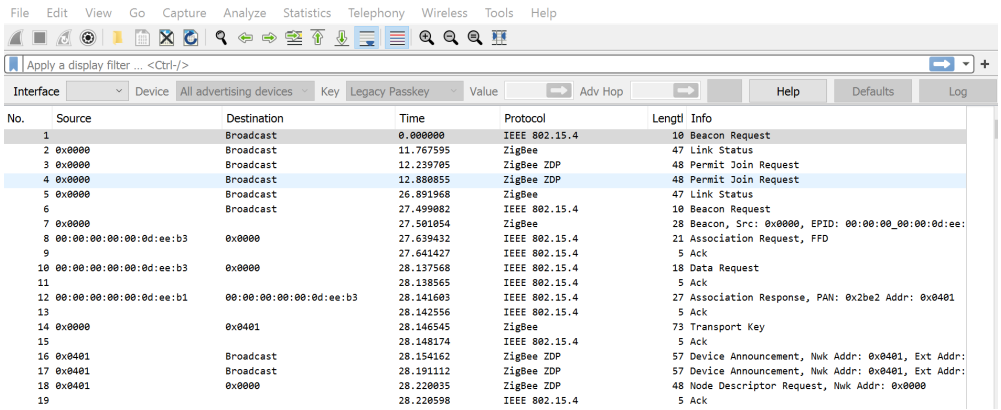


Notes:

- The AT86RF233 ZIGBIT USB stick (2.4 GHz) supports the following data rates:
  - Channel page 0 – Data rate is 250 kbps
  - Channel page 2 – Data rate is 500 kbps
  - Channel page16 – Data rate is 1 Mbps
  - Channel page 17 – Data rate is 2 Mbps
- The AT86RF212B ZIGBIT USB stick (Sub-GHz) supports the following data rates:
  - Channel page 0 – Data rate is 20 kbps (Channel 0), 40 kbps (Channel 1-10)
  - Channel page 2 – Data rate is 100 kbps (Channel 0), 250 kbps (Channel 1-10)
  - Channel page 5 – Data rate is 250 kbps
  - Channel page 16 – Data rate is 200 kbps (Channel 0), 500 kbps (Channel 1-10)
  - Channel page 17 – Data rate is 400 kbps (Channel 0),1 Mbps (Channel 1-10)
  - Channel page 18 – Data rate is 500 kbps
  - Channel page 19 – Data rate is 1 Mbps

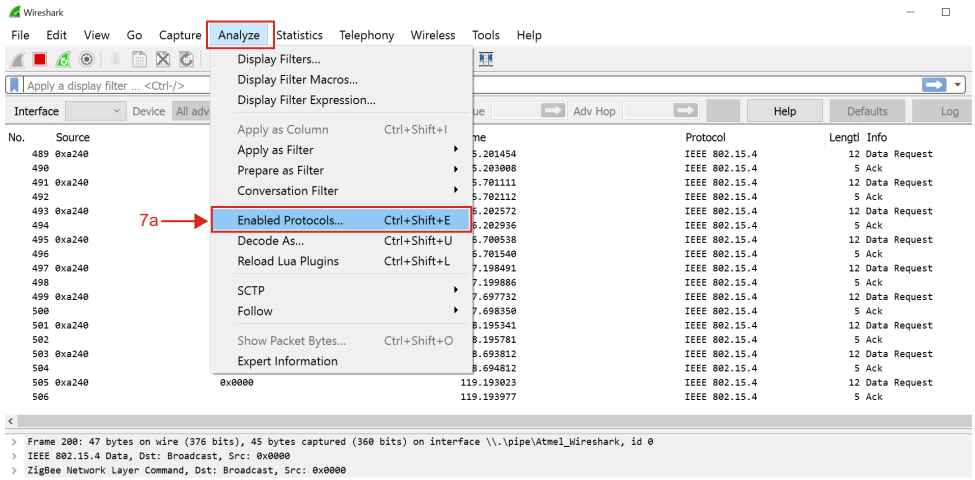
The following pop-up window appears showing the packets captured in the Wireshark.

Figure 3-3. Wireshark Start-up Window



a. Go to *Analyze>Enabled Protocols* to select the protocols.

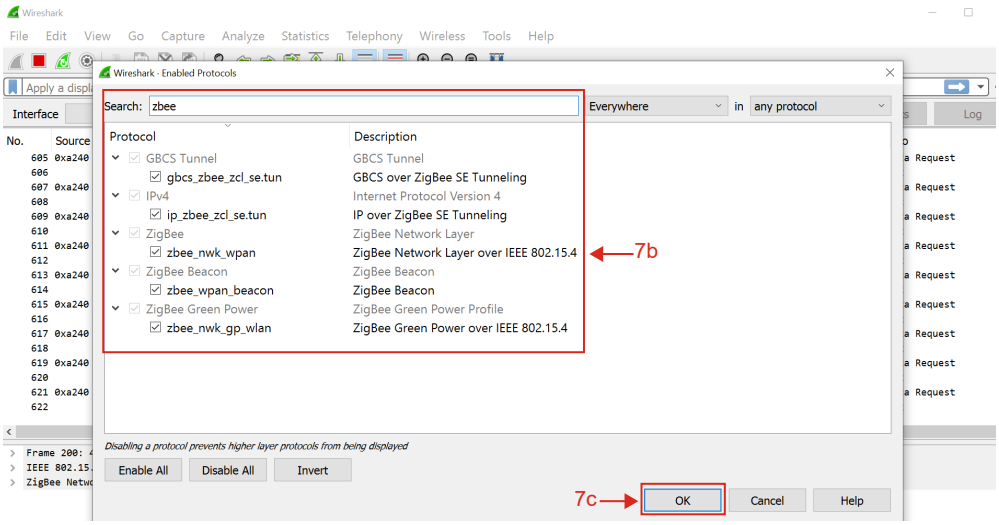
Figure 3-4. Enabled Protocols Selection



b. The user can select the protocols as per the requirement. For example, in this scenario, search for *zbee* to select **ZigBee** protocols.

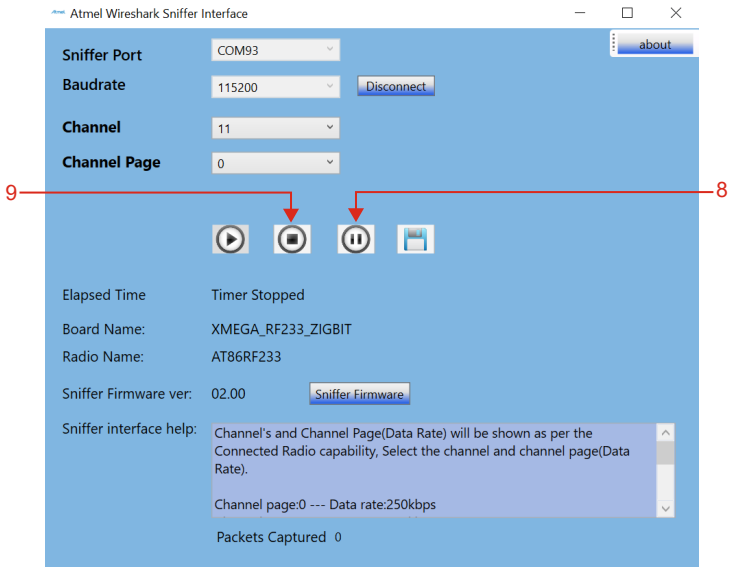
c. Click **OK** to continue.

Figure 3-5. Protocol Selection



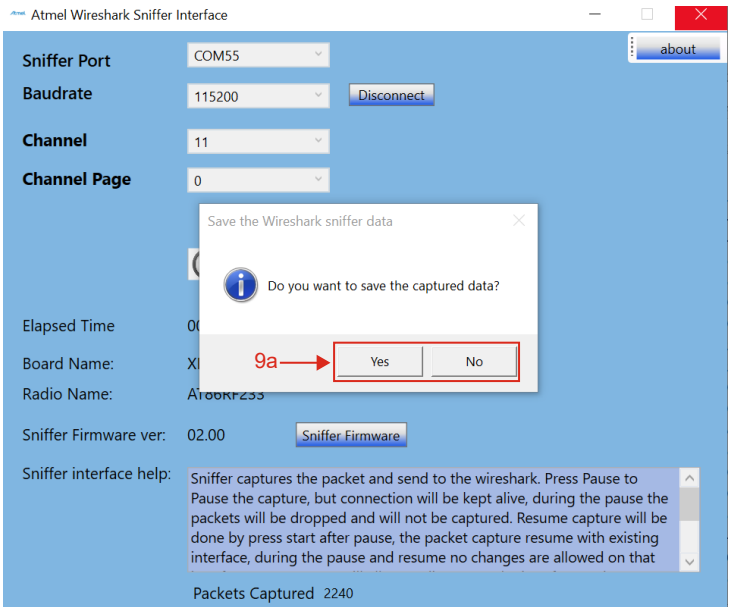
8. Click **Pause** to pause capturing of packets.
9. Click **Stop** to stop capturing of packets.

Figure 3-6. Pause/Stop Buttons



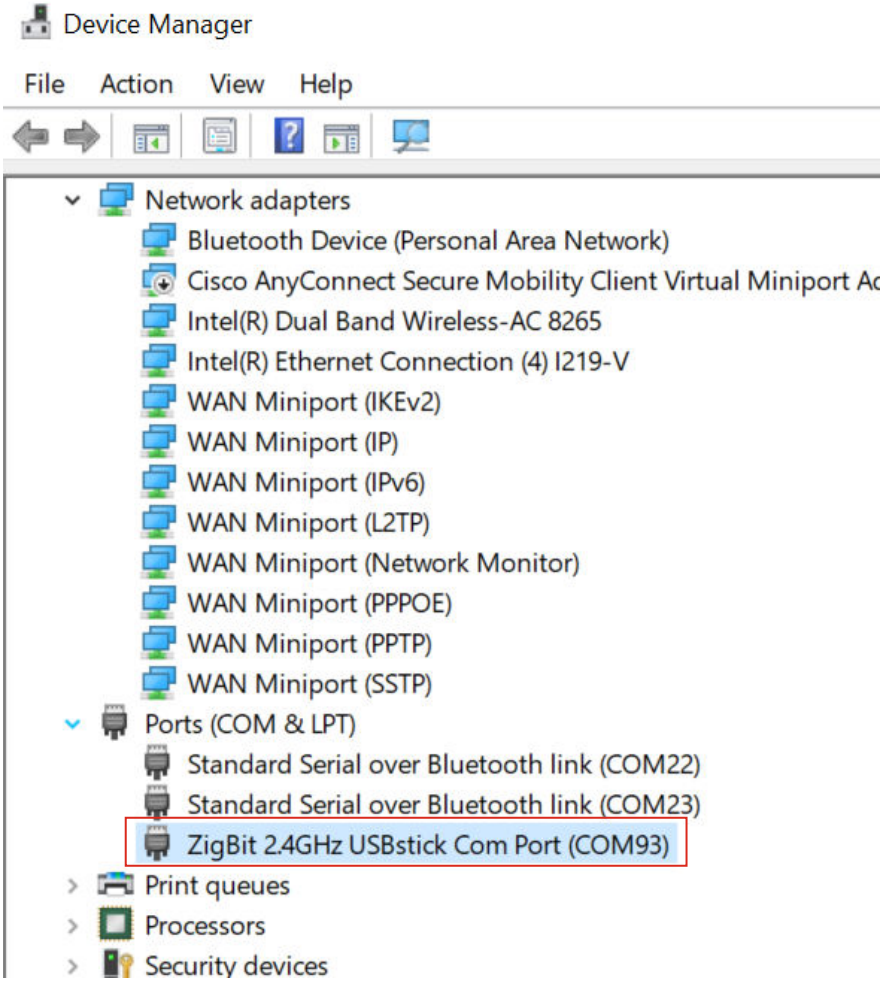
a. The following pop-up dialogue box appears, and the user must click **Yes** or **No** to save/delete the capture file (if there is any capture in the previous channel/instance).

Figure 3-7. Save the Wireshark Sniffer Data



10. The user can save the captured file for future reference or can continue without saving.
    Note: The user must ensure adding the path of `Wireshark-winXX-3.X.X.exe` in the system environment variables.
11. The following figure illustrates the ZigBit 2.4 GHz USB stick in the Device Manger of the PC.

Figure 3-8. ZigBit 2.4 GHz USB Stick Com Port (Sniffer) Listing in Windows Device Manager



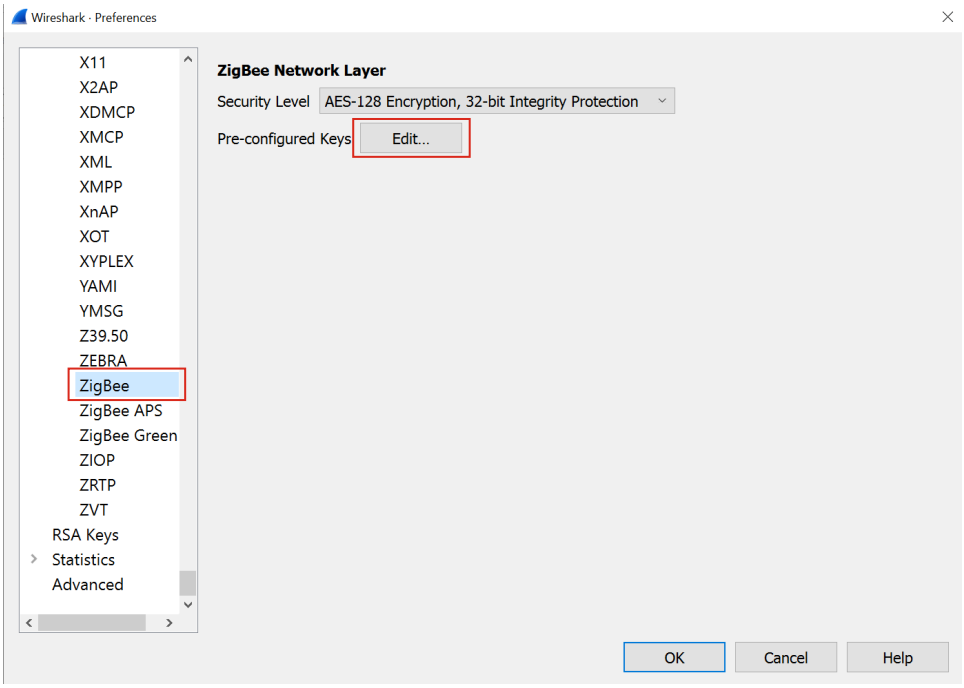
4. Configuring Sniffer Preferences

The Wireshark's GUI provides multiple filtering options for easy viewing and analysis. The user can obtain a complete outlook of the wireless network with the appropriate settings. This chapter provides information on configuring such preferences in the Wireshark's GUI.

4.1 Wireshark Capture Interface

- Protocols – Wireshark automatically identifies the protocol in use. All supported protocols are enabled by default, go to *Analyze>Enabled Protocols* to see all the menu options. The user can use this option to enable or disable protocols as per the requirement.
  Note: The user must ensure all the required protocols are enabled before capturing the packets.
- Security – It is possible to monitor encrypted ZigBee network data by entering the Network (NWK) security key used in the network. Go to *Edit>Preferences>Protocols>ZigBee*. The following figure illustrates the security key configuration options in Wireshark.
  – From the "Pre-configured Keys", click **Edit** to enter the security keys (see Figure 4-2).

Figure 4-1. Security Preferences in Wireshark



The security level can be set as per the *Zigbee Specification Revision 22 1.0*. The following table provides details about the security levels.

Table 4-1. Security Levels Available to the NWK, and Application Support Sub-Layer (APS)

Security Level Identifier	Security Level Subfield	Security Attributes	Data Encryption	Frame Integrity (Length M of MIC,in Number of Octets)
0x00	000	None	OFF	NO (M = 0)
0x01	001	MIC-32	OFF	YES (M = 4)
0x02	010	MIC-64	OFF	YES = (M = 8)

.....continued

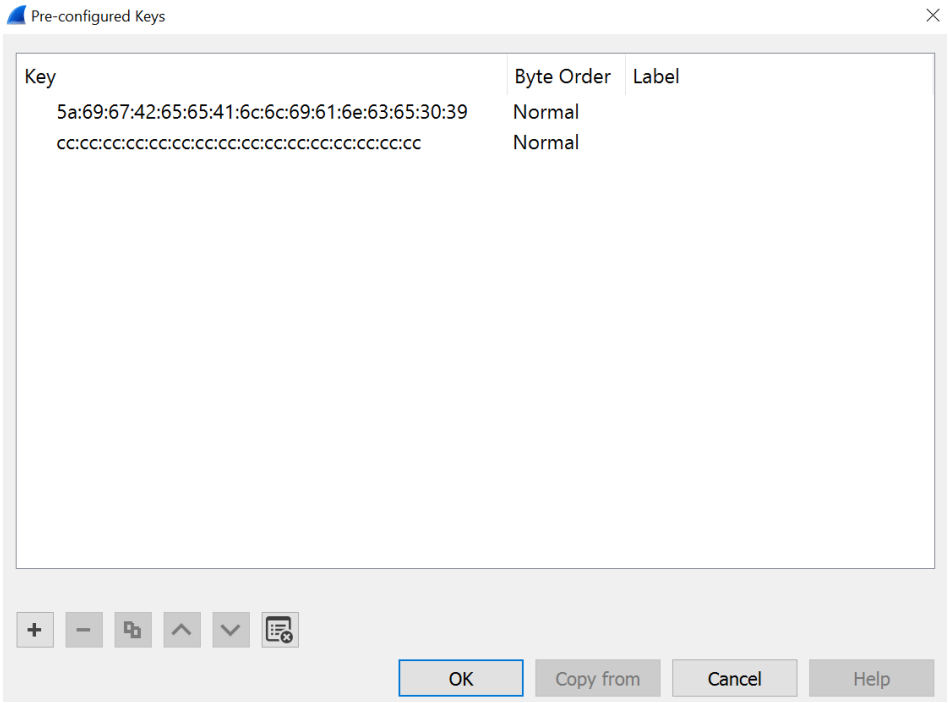
Security Level Identifier	Security Level Subfield	Security Attributes	Data Encryption	Frame Integrity (Length M of MIC,in Number of Octets)
0x03	011	MIC-128	OFF	YES (M = 16)
0x04	100	ENC	ON	NO (M = 0)
0x05	101	ENC-MIC-32	ON	YES (M = 4)
0x06	110	ENC-MIC-64	ON	YES = (M = 8)
0x07	111	ENC-MIC-128	ON	YES (M = 16)

Note: For more details on the security levels, refer to the *Table 4-30 Security Levels Available to the NWK, and APS Layers* in the *Zigbee Specification Revision 22 1.0* (05-3474-22).

It is possible to add multiple keys and edit or remove existing keys. The following figure illustrates the security key entries.

For example, for a Zigbee network that uses centralized security in the APS layer, a device joining the network establishes a link key with the trust center. To view all APS transactions happening in this link, such as the APS `Transport Key` command, add the Trust Center Link Key and network key under the preferences tab in Wireshark (see the following figure).

Figure 4-2. Security Key Entries

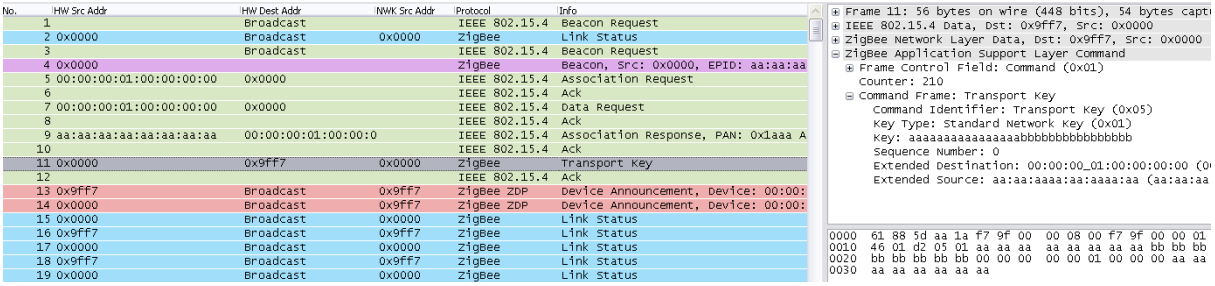


The user can customize the following viewing options in the Wireshark:

- For arranging the layout of the panels, go to *Edit>Preferences>Layout*.
- For adding columns to the packet display pane (for example, `HW Src Addr`), go to *Edit>Preferences>Columns*.
- To colorize frame formats (for example, NWK Link Status Frames), go to *View>Coloring Rules*. For more details, refer to the *Packet colorization* (11.3).
- Perform the following steps to apply filters to display frames based on chosen fields in a frame:
  a. Right-click the field

b. Select *Apply as Filter*

Figure 4-3. Wireshark Capture Screen Layout



5. Analyzing Data Traffic in Zigbee Pro Networks

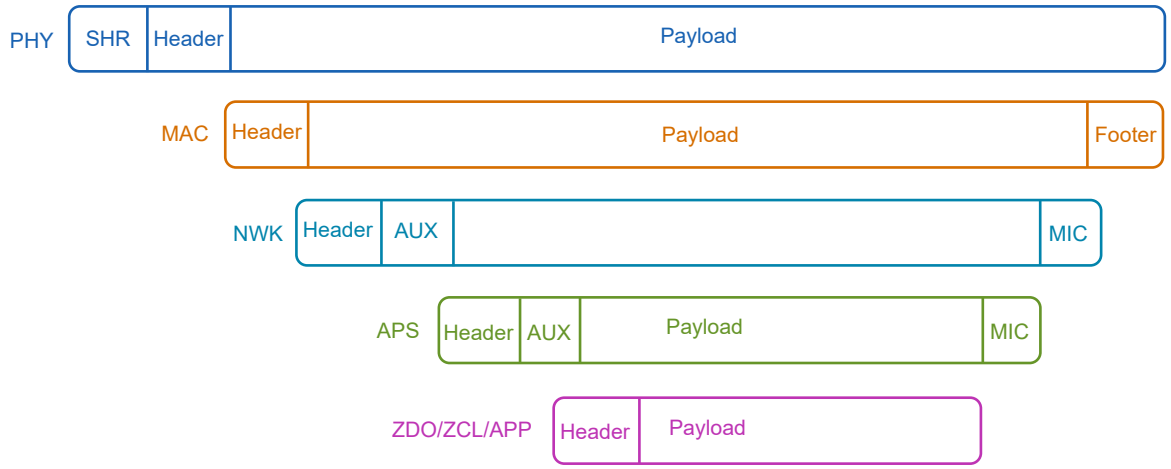
This chapter provides examples of common interaction in Zigbee Pro networks, helping the user to look closely into various fields of the frame.

Note: It does not cover all scenarios that fall under Zigbee specification.

5.1 Zigbee Frame Format Overview

The following figure illustrates the skeletal overview of the Zigbee frame format (APS and NWK layer security headers and footers). Zigbee uses a non-beacon enabled MAC format with no security in the MAC layer.

Figure 5-1. Zigbee Frame Format



5.2 MAC Association

Every node in a Zigbee network has its own unique 64-bit IEEE MAC address. When a node joins the network for the first time, the end device/router initiates the MAC association procedure, after which, it obtains a 16-bit network (short) address from the parent. For further communication in the network, this short address is used to reduce frame overhead. Use a configuration parameter, IEEE MAC address to set the value of the 64-bit MAC address of the node during compilation. The following conditions prevail:

- Setting `IEEE MAC address` for testing – Any random 64-bit value can be set at compile time in application configuration files or at run-time before calling the Application Programming Interface (API) to start the network request via ZDP.
- Setting `IEEE MAC address` during production – Commercial use of the Zigbee products requires the purchase of a block of IEEE/MAC addresses from IEEE. In this case, `IEEE MAC address` can be set to zero during compile-time.

When a node starts up, it does network discovery by performing an active scan over the channels specified in the list of channels configured at the Zigbee stack parameter configuration files or at application level. The node sends the `Beacon Request` (see packet `#1`) (see the following figure). After receiving a `Beacon Request` frame, the routers and coordinators already present in the network automatically respond with a beacon frame. The joining node filters a potential parent based on the settings in the beacon packet received.

Figure 5-2. Node Joins Network via MAC Association using IEEE Address 0xdeeb1ULL

No.	Source	Destination	Time	Protocol	Leng	Info
1		Broadcast	0.000000	IEEE 802.15.4	10	Beacon Request
2	0x0000	Broadcast	11.767595	ZigBee	47	Link Status
3	0x0000	Broadcast	12.239705	ZigBee ZDP	48	Permit Join Request
4	0x0000	Broadcast	12.880855	ZigBee ZDP	48	Permit Join Request
5	0x0000	Broadcast	26.891968	ZigBee	47	Link Status
6		Broadcast	27.499082	IEEE 802.15.4	10	Beacon Request
7	0x0000		27.501054	ZigBee	28	Beacon, Src: 0x0000, EPID: 00:00:00_00:00:0d:ee:b1
8	00:00:00:00:00:0d:ee:b3	0x0000	27.639432	IEEE 802.15.4	21	Association Request, FFD
9			27.641427	IEEE 802.15.4	5	Ack
10	00:00:00:00:00:0d:ee:b3	0x0000	28.137568	IEEE 802.15.4	18	Data Request
11			28.138565	IEEE 802.15.4	5	Ack
12	00:00:00:00:00:0d:ee:b1	00:00:00:00:00:0d:ee:b3	28.141603	IEEE 802.15.4	27	Association Response, PAN: 0x2be2 Addr: 0x0401
13			28.142556	IEEE 802.15.4	5	Ack

The beacon from the coordinator/router contains the `Association Permit` sub field. It is set to `True` if the device accepts the association to the Personal Area Network (PAN). A joining node cannot associate to the device if this sub-field is set to `False`. The `PERMIT DURATION` parameter in the Zigbee application controls the joining of devices into the network by setting a finite permit duration.

Figure 5-3. assocPermit Sub-Field in the Beacon Frame

```
IEEE 802.15.4 Beacon, Src: 0x0000
    <Frame Length: 26>
  > Frame Control Field: 0x8000, Frame Type: Beacon, Destination Addressing Mode: None, Frame Version: IEEE Std 802.15.4-2003, Source Addressing Mode: Short/16-bit
    Sequence Number: 249
    Source PAN: 0x1361
    Source: 0x0000
    <[Address: 0x0000]>
  v Superframe Specification: PAN Coordinator, Association Permit
        .... .... .... 1111 = Beacon Interval: 15
        .... .... 1111 .... = Superframe Interval: 15
        .... 1111 .... .... = Final CAP Slot: 15
        ...0 .... .... .... = Battery Extension: False
        .1.. .... .... .... = PAN Coordinator: True
        1... .... .... .... = Association Permit: True
  > GTS
    Pending Addresses: 0 Short and 0 Long
```

For example, the following figure illustrates the parsed beacon payload that contains information based on which joining node chooses a potential parent.

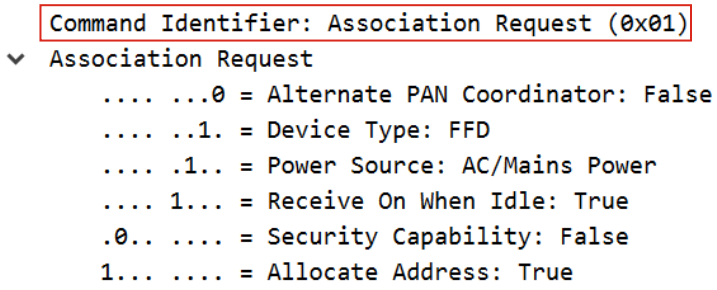
The beacon payload provides information on the Zigbee stack profile used in the network (`Stack Profile: ZigBee PRO = 0x2`), network protocol version (`nwkcProtocolVersion`, `0x02`). The `Router Capacity`, `End Device Capacity` and `Device Depth` limits the acceptance of children by a parent node. For more details, refer to the *Zigbee Specification Revision 22 1.0* (05-3474-22).

Figure 5-4. Beacon Payload

```
ZigBee Beacon, ZigBee PRO, EPID: 00:00:00_00:00:00:0b:ee
    Protocol ID: 0
  v Beacon: Stack Profile: ZigBee PRO, Router Capacity, End Device Capacity
        .... .... .... 0010 = Stack Profile: ZigBee PRO (0x2)
        .... .... 0010 .... = Protocol Version: 2
        .... .1.. .... .... = Router Capacity: True
        .000 0... .... .... = Device Depth: 0
        1... .... .... .... = End Device Capacity: True
    Extended PAN ID: 00:00:00_00:00:00:0b:ee (00:00:00:00:00:00:0b:ee)
    Tx Offset: 16777215
    Update ID: 0
```

The following figure illustrates a joining device indicating its capability information in the MAC `Association Request` it sends to its potential parent.

Figure 5-5. Capability Information in a MAC Association Request



5.3 Self-Leave and Parent-Induced Leave

The Zigbee Device Object (ZDO) layer manages the ZDP requests and uses them for various network control scenarios.

Use the ZDP requests to process the network leave when a device needs to leave the network on certain events. Network leave can be self-induced on a node or a node can order another remote node to leave the network.

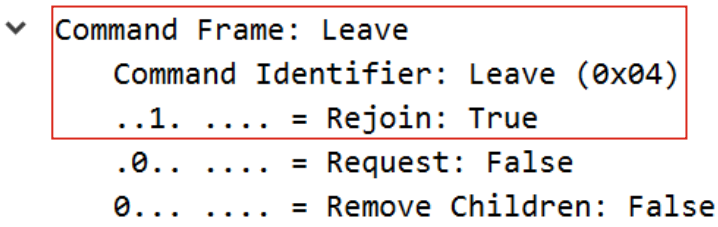
The following figure illustrates a node with a short address: `0x457a`. It leaves the network on its own (self-induced) and rejoins the network (see packet `#3300`) by sending a rejoin request.

Figure 5-6. Self-Leave of Node (End Device) with Short Address 0x457a and Extended Address `0xdeeb1ULL`

3292	0x457a	Broadcast	868.566706	ZigBee	47	Leave
3293			868.567703	IEEE 802.15.4	5	Ack
3294		Broadcast	868.712432	IEEE 802.15.4	10	Beacon Request
3295	0x0000		868.715116	ZigBee	28	Beacon, Src: 0x0000, EPID: 00:00:00_00:00:0d:ee:b1
3296		Broadcast	869.131080	IEEE 802.15.4	10	Beacon Request
3297	0x0000		869.135077	ZigBee	28	Beacon, Src: 0x0000, EPID: 00:00:00_00:00:0d:ee:b1
3298		Broadcast	870.272629	IEEE 802.15.4	10	Beacon Request
3299	0x0000		870.275375	ZigBee	28	Beacon, Src: 0x0000, EPID: 00:00:00_00:00:0d:ee:b1
3300	0x457a	0x0000	870.694727	ZigBee	47	Rejoin Request, Device: 0x457a
3301			870.696141	IEEE 802.15.4	5	Ack
3302	0x457a	0x0000	871.271862	IEEE 802.15.4	12	Data Request
3303			871.272858	IEEE 802.15.4	5	Ack
3304	0x0000	0x457a	871.276892	ZigBee	57	Rejoin Response, New Address: 0x457a
3305			871.278842	IEEE 802.15.4	5	Ack

The node rejoins because the `Rejoin` bit is set to `True` in the `Command Frame: Leave`. The following figure illustrates the leave packet rejoin bit setting.

Figure 5-7. Leave Packet Rejoin Bit Setting



The following figure illustrates a parent node requesting the child device with a short address, `0x6915`, to leave (`Leave Request`) (see packet `#99`). The child device sends a rejoin response (see packet `#101`). After a few seconds, the child device rejoins the network with the same short address. In this case, the child device rejoins a network with known network parameters, such as network PANID.

Figure 5-8. Parent Node 0x0000 Sends a ZDP Request Requesting Child 0x6915 to Leave

No.	Source	Destination	Time	Protocol	Leng	Info
99	0x0000	0x6915	344.198134	ZigBee ZDP	55	Leave Request, Device: 00:00:00_00:00:0d:ee:b3
100			344.199875	IEEE 802.15.4	5	Ack
101	0x6915	0x0000	344.200877	ZigBee ZDP	47	Leave Response, Status: Success
102			344.201874	IEEE 802.15.4	5	Ack
103	0x0000	0x6915	344.205547	ZigBee	45	APS: Ack, Dst Endpt: 0, Src Endpt: 0
104			344.207157	IEEE 802.15.4	5	Ack
105	0x6915	Broadcast	344.241797	ZigBee	47	Leave
106	0x6915	Broadcast	344.320390	ZigBee	47	Leave
107		Broadcast	344.324058	IEEE 802.15.4	10	Beacon Request
108	0x0000		344.327381	ZigBee	28	Beacon, Src: 0x0000, EPID: 00:00:00_00:00:0d:ee:b7
109	0x6915	0x0000	344.464216	ZigBee	47	Rejoin Request, Device: 0x6915
110			344.465868	IEEE 802.15.4	5	Ack
111	0x0000	0x6915	344.469117	ZigBee	57	Rejoin Response, New Address: 0x6915
112			344.469526	IEEE 802.15.4	5	Ack
113	0x6915	Broadcast	344.475399	ZigBee ZDP	57	Device Announcement, Nwk Addr: 0x6915, Ext Addr: 00:00:00_0
114	0x6915	Broadcast	344.550166	ZigBee ZDP	57	Device Announcement, Nwk Addr: 0x6915, Ext Addr: 00:00:00_0
115	0x6915	Broadcast	355.722891	ZigBee	50	Link Status
116	0x0000	Broadcast	356.902136	ZigBee	50	Link Status
117	0x6915	Broadcast	370.883667	ZigBee	50	Link Status

The following figure illustrates the parent request to the child device to leave the network with no rejoin.

Figure 5-9. Parent Induced Leave with No Rejoin

No.	Source	Destination	Time	Protocol	Leng	Info
143	0x0000	0x6915	509.080714	ZigBee ZDP	55	Leave Request, Device: 00:00:00_00:00:0d:ee:b3
144			509.082196	IEEE 802.15.4	5	Ack
145	0x6915	0x0000	509.082196	ZigBee ZDP	47	Leave Response, Status: Success
146			509.083197	IEEE 802.15.4	5	Ack
147	0x0000	0x6915	509.086683	ZigBee	45	APS: Ack, Dst Endpt: 0, Src Endpt: 0
148			509.087212	IEEE 802.15.4	5	Ack
149	0x6915	Broadcast	509.163823	ZigBee	47	Leave
150	0x6915	Broadcast	509.245952	ZigBee	47	Leave
151	0x0000	Broadcast	523.198620	ZigBee	47	Link Status
152	0x0000	Broadcast	538.359397	ZigBee	47	Link Status
153	0x0000	Broadcast	553.399912	ZigBee	47	Link Status
154	0x0000	Broadcast	568.443126	ZigBee	47	Link Status
155	0x0000	Broadcast	583.525611	ZigBee	47	Link Status
156	0x0000	Broadcast	598.565007	ZigBee	47	Link Status
157		Broadcast	608.471662	IEEE 802.15.4	10	Beacon Request
158	0x0000		608.472570	ZigBee	28	Beacon, Src: 0x0000, EPID: 00:00:00_00:00:0d:ee:b7
159	00:00:00:00:00:0d:ee:b3	0x0000	609.032644	IEEE 802.15.4	21	Association Request, FFD
160			609.033977	IEEE 802.15.4	5	Ack
161	00:00:00:00:00:0d:ee:b3	0x0000	609.529476	IEEE 802.15.4	18	Data Request
162			609.530474	IEEE 802.15.4	5	Ack
163	00:00:00:00:00:0d:ee:b7	00:00:00:00:00:0d:ee:b3	609.533465	IEEE 802.15.4	27	Association Response, PAN: 0x0daf Addr: 0x0a18

The difference between Figure 5-8 and the Figure 5-9 is the rejoin bit setting.

- If the Rejoin bit is set to `True`, the leaving device rejoins using rejoin request.
- If the Rejoin bit is set to `False`, the rejoining can happen using MAC association, in the case of invoking the child device to join the network.

It is possible to configure options, such as rejoin and removal of children in the leave request.

5.4 Network (NWK) Link Status Frame

The routers and coordinator send the Network (NWK) link status frames, so that neighboring nodes can maintain information on the link costs required for routing. The `Link Status` frames are periodically transmitted as one-hop broadcasts. The `Link Status` list contains the short address and link cost information of all neighboring nodes. The following figure illustrates the header information in a NWK link status frame.

Figure 5-10. NWK Link Status Command Frame

```
> Frame 144: 50 bytes on wire (400 bits), 48 bytes captured (384 bits) on interface \\.\pipe\Atmel_Wireshark, id 0
> IEEE 802.15.4 Data, Dst: Broadcast, Src: 0x0000
v ZigBee Network Layer Command, Dst: Broadcast, Src: 0x0000
  > Frame Control Field: 0x1209, Frame Type: Command, Discover Route: Suppress, Security, Extended Source Command
    Destination: 0xfffc
    <[Address: 0xfffc]>
    Source: 0x0000
    <[Address: 0x0000]>
    Radius: 1
    Sequence Number: 245
    Extended Source: 00:00:00_00:00:0d:ee:b1 (00:00:00:00:00:0d:ee:b1)
    <[Extended Address: 00:00:00_00:00:0d:ee:b1 (00:00:00:00:00:0d:ee:b1)]>
  > ZigBee Security Header
  v Command Frame: Link Status
      Command Identifier: Link Status (0x08)
      .1.. .... = Last Frame: True
      ..1. .... = First Frame: True
      ...0 0001 = Link Status Count: 1
    v Link 1
        Address: 0x0401
        .... .011 = Incoming Cost: 3
        .100 .... = Outgoing Cost: 4
```

5.5 Multicast

Broadcasting a message to a group of nodes involves creating a group table entry for a specified end-point and group ID. The following figure illustrates a multicast transmission from coordinator to group with group ID, `Group: 0x1111` and endpoint `0x20`. The network destination address is the group address.

Figure 5-11. Multicast Sub-field – NWK Header

```
v ZigBee Application Support Layer Data, Group: 0x1111, Src Endpt: 20
  v Frame Control Field: Data (0x0c)
        .... ..00 = Frame Type: Data (0x0)
        .... 11.. = Delivery Mode: Group (0x3)
        ..0. .... = Security: False
        .0.. .... = Acknowledgement Request: False
        0... .... = Extended Header: False
    Group: 0x1111
    Cluster: On/Off (0x0006)
    Profile: Home Automation (0x0104)
    Source Endpoint: 20
    Counter: 26
v ZigBee Cluster Library Frame
  v Frame Control Field: Cluster-specific (0x11)
        .... ..01 = Frame Type: Cluster-specific (0x1)
        .... .0.. = Manufacturer Specific: False
        .... 0... = Direction: Client to Server
        ...1 .... = Disable Default Response: True
    Sequence Number: 13
    Command: On (0x01)
```

5.6 Fragmentation

When the length of Application Layer (APL) data packets is greater than the maximum limit of the APL payload, the stack fragments the entire data into blocks.

Figure 5-12. Fragmentation – Relevant Header Information

```
> Frame 27: 57 bytes on wire (456 bits), 57 bytes captured (456 bits)
> IEEE 802.15.4 Data, Dst: 0xbc8f, Src: 0x0000
> ZigBee Network Layer Data, Dst: 0xbc8f, Src: 0x0000
v ZigBee Application Support Layer Data, Dst Endpt: 240, Src Endpt: 1
  v Frame Control Field: Data (0xc0)
      .... ..00 = Frame Type: Data (0x0)
      .... 00.. = Delivery Mode: Unicast (0x0)
      ..0. .... = Security: False
      .1.. .... = Acknowledgement Request: True
      1... .... = Extended Header: True
    Destination Endpoint: 240
    Cluster: Transmit Counted Packets (0x0001)
    Profile: Test Profile #2 (0x7f01)
    Source Endpoint: 1
    Counter: 160
  v Extended Frame Control Field (0x02)
      .... ..10 = Fragmentation: Middle Block (0x2)
      Block Number: 1
    Reassembled in: 43
v Data (30 bytes)
    Data: 1b 1c 1d 1e 1f 20 21 22 23 24 25 26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 32 …
    [Length: 30]
```

The following figure illustrates the fragmentation example.

Figure 5-13. Fragmentation – Example

No.	Source	Destination	Time	Protocol	Length	Info
20	0xbc8f	Broadcast	21.493704	ZigBee ZDP	37	Device Announcement, Nwk Addr: 0xbc8f, Ext Addr: 00:00:00_02:00:00:00:00
21	0xbc8f	Broadcast	22.133464	ZigBee ZDP	37	Device Announcement, Nwk Addr: 0xbc8f, Ext Addr: 00:00:00_02:00:00:00:00
22	0x0000	Broadcast	25.155616	ZigBee	33	Link Status
23	0xcbff	Broadcast	29.473672	ZigBee	33	Link Status
24	0xbc8f	Broadcast	30.607360	ZigBee	33	Link Status
25	0x0000	0xbc8f	36.144840	ZigBee T2	57	Transmit Counted Packets, Src Endpt: 1 (fragment 0)
26			36.147112	IEEE 802.15.4	3	Ack
27	0x0000	0xbc8f	36.243768	ZigBee T2	57	Transmit Counted Packets, Src Endpt: 1 (fragment 1)
28			36.246048	IEEE 802.15.4	3	Ack
29	0x0000	0xbc8f	36.343496	ZigBee T2	57	Transmit Counted Packets, Src Endpt: 1 (fragment 2)
30			36.345768	IEEE 802.15.4	3	Ack
31	0xbc8f	0x0000	36.350384	ZigBee	28	APS: Ack, Dst Endpt: 1, Src Endpt: 240
32			36.351728	IEEE 802.15.4	3	Ack
33	0x0000	0xbc8f	36.355896	ZigBee T2	57	Transmit Counted Packets, Src Endpt: 1 (fragment 3)
34			36.358168	IEEE 802.15.4	103	Data
35	0x0000	0xbc8f	36.452296	ZigBee T2	57	Transmit Counted Packets, Src Endpt: 1 (fragment 4)
36			36.454568	IEEE 802.15.4	3	Ack
37	0x0000	0xbc8f	36.552688	ZigBee T2	57	Transmit Counted Packets, Src Endpt: 1 (fragment 5)
38			36.554960	IEEE 802.15.4	3	Ack
39	0xbc8f	0x0000	36.558864	ZigBee	28	APS: Ack, Dst Endpt: 1, Src Endpt: 240
40			36.560208	IEEE 802.15.4	3	Ack
41	0x0000	0xbc8f	36.564128	ZigBee T2	57	Transmit Counted Packets, Src Endpt: 1 (fragment 6)
42			36.566400	IEEE 802.15.4	3	Ack
43	0x0000	0xbc8f	36.662472	ZigBee T2	57	Transmit Counted Packets, Src Endpt: 1
44			36.664744	IEEE 802.15.4	3	Ack
45	0xbc8f	0x0000	36.670120	ZigBee	28	APS: Ack, Dst Endpt: 1, Src Endpt: 240
46			36.671464	IEEE 802.15.4	3	Ack

The sender node sends the first fragment with the block number as the total number of blocks comprising the entire APL data. The subsequent fragments have block numbers starting from one going up to the maximum transmission window size. The receiving node sends an APS acknowledgment frame after receiving all blocks in the transmission window. For more details, refer to the *Zigbee Specification Revision 22 1.0* (05-3474-22).

5.7 Service Discovery

Service discovery is the process of collecting information on supported clusters on other devices in the network. Service discovery uses ZDP requests for every cluster ID supported. Service discovery requests can be unicast or broadcast, and so the response contains the network address of the responder along with the matched simple descriptor information. The response contains a match list with the end-points that support the cluster in the request. For more details on service discovery, refer to 6. Analyzing Data Traffic in Zigbee 3.0 Protocol.

5.8 Tunneling in Secure Networks

Consider a network wherein a node insecurely joins through a router parent, and the joining node does not know the network key prior to the join procedure. In this case, using the APS command to securely communicate the network key from the trust center to the newly joined router is called the APS tunnel command.

The end-device `0x0beeLL` joins router `0x3c08` from packet `#89`. The parent router sends an APS `Update Device` command (packet `#91`) to the trust center to inform it whether a node has joined or left the network. The following table and Figure 5-15 provide details about the update status of the device, from which the trust center takes necessary action, which is to send the network or remove the key and associated security counters for the device.

Figure 5-14. Tunneling

No.	Source	Destination	Time	Protocol	Length	Info
82		Broadcast	415.660689	IEEE 802.15.4	8	Beacon Request
83	0x0000		415.662913	ZigBee	26	Beacon, Src: 0x0000, EPID: 00:00:00_00:00:0d:ee:b7
84	0x3c08		415.670713	ZigBee	26	Beacon, Src: 0x3c08, EPID: 00:00:00_00:00:0d:ee:b7
85	00:00:00:00:00:00:0b:ee	0x3c08	416.221401	IEEE 802.15.4	19	Association Request, FFD
86			416.222473	IEEE 802.15.4	3	Ack
87	00:00:00:00:00:00:0b:ee	0x3c08	416.719889	IEEE 802.15.4	16	Data Request
88			416.720865	IEEE 802.15.4	3	Ack
89	00:00:00:00:00:0d:ee:b3	00:00:00:00:00:00:0b:ee	416.722841	IEEE 802.15.4	25	Association Response, PAN: 0x40a8 Addr: 0x54d7
90			416.724105	IEEE 802.15.4	3	Ack
91	0x3c08	0x0000	416.726233	ZigBee	66	Update Device
92			416.728801	IEEE 802.15.4	3	Ack
93	0x0000	0x3c08	416.732801	ZigBee	100	Transport Key
94			416.736465	IEEE 802.15.4	3	Ack
95	0x3c08	0x54d7	416.737593	ZigBee	71	Transport Key
96			416.740329	IEEE 802.15.4	3	Ack
97	0x54d7	Broadcast	416.745681	ZigBee ZDP	55	Device Announcement, Nwk Addr: 0x54d7, Ext Addr: 00:00:00_00:00
98	0x54d7	Broadcast	416.781937	ZigBee ZDP	55	Device Announcement, Nwk Addr: 0x54d7, Ext Addr: 00:00:00_00:00

The trust center sends the APS tunnel command frame in packet `#93`. The tunnel command frame contains the secured frame to be sent to the destination in its payload. Packet `#95` shows the APS `Transport Key` command frame sent from the router parent to the newly joined end-device. It includes the key sequence number and the active network key. In case the router joins with a pre-configured network key, the APS transport packet contains a key sequence number and the key values as all-zeros. The end-device receives the `Transport Key` command frame, sets and activates the network key and does a device announcement to the network (packets `#97` and `#98`).

Table 5-1. Status Field in APS Update Device Command

Parameter Name	Type	Valid Range	Description
Status	Integer	0x00-0x07	Indicates the updated status of the device given by the DeviceAddress parameter. • 0x00 = Standard device secured rejoin • 0x01 = Standard device unsecured join • 0x02 = Device left • 0x03 = Standard device unsecured rejoin • 0x04 = High security device secured rejoin • 0x05 = High security device unsecured join • 0x06 = Reserved • 0x07 = High security device unsecured rejoin
Note: For more details on the status field in APS update device command, refer to the *ZigBee Specification Revision 22 1.0* (05-3474-22).			

Figure 5-15. Update Device Status

```
Command Frame: Update Device
    Command Identifier: Update Device (0x06)
    Device Address: 00:00:00_00:00:00:0b:ee (00:00:00:00:00:00:0b:ee)
    Device Address: 0x54d7
    Device Status: Standard security, unsecured join (0x01)
```

6. Analyzing Data Traffic in Zigbee 3.0 Protocol

The Zigbee Alliance defines a set of standard device types. These device types specify the functionality of a device. This functionality is again dependent on independent functional entities called clusters. The cluster is a container of attributes, and can read/write through command/responses defined by Zigbee Device Profile (ZDP). The alliance also provides a Zigbee Cluster Library (ZCL) that acts as a repository for cluster functionality.

The packet capture was performed using the Wireshark/Zigbit sniffer for data transfer between various combinations of Zigbee device types. The following are the three Zigbee device types:

- Zigbee coordinator/Zigbee combined interface – Device capable of controlling and monitoring other devices. In general, it is a mains-powered device like a personal computer.
- Zigbee router/Zigbee lights – A lighting device that can be switched ON/OFF. Adjust the brightness and color of the light via the color commands.
- Zigbee end device/Zigbee multisensor

The following are the two Zigbee network architectures:

- Centralized network – Zigbee coordinator device can form a centralized network.
- Distributed network – Zigbee router device can form a distributed network.

The packet capture focuses on the following scenarios:

- Zigbee coordinator – Centralized network formation (Zigbee combined interface application). Commissioning and data exchange between Zigbee coordinator and Zigbee router (Zigbee extended color lights). For more details, refer to 6.2. Zigbee Coordinator.
- Zigbee router – Extended lights application is commissioned to the existing Zigbee network formed by the Zigbee coordinator (combined interface or Zigbee router is capable of creating a new Zigbee distributed network (if there is no nearby network)). Here it is commissioned to the existing Zigbee centralized network formed by Zigbee coordinator/combined interface. For more details, refer to 6.3. Zigbee Router.
- Zigbee end device – Joined to Zigbee coordinator (combined interface). After joining, the end device (multisensor/sensor device type) starts the ZCL attribute reporting of sensor data, such as temperature, occupancy, light and humidity after connecting to the network formed by the coordinator. For more details, refer to 6.4. Zigbee End Device.
- Touchlink commissioning – In this application note, the commissioning process happens between Zigbee extended lights (router) and the color scene controller (end device). For more details, refer to 6.5. Touchlink Commissioning.

6.1 General Description

6.1.1 Base Device Behavior (BDB)

BDB (Base Device Behavior) layer supports the initialization, commissioning and operating procedures of a base device operating on the Zigbee PRO stack to ensure profile interoperability. For more details, refer to the *PRO Base Device Behavior Specification* (3.0.1).

Commissioning

Commissioning is the process of initializing the devices to join a network and to work together. The Zigbee BDB specification specifies the execution order of the procedures for the following commissioning mechanisms:

1. Touchlink – A node can support the proximity-based commissioning mechanism. If touchlink commissioning is supported, the node supports touchlink as an initiator, a target or both.
   - Initiator – A member of an existing network or (if not) creates a new network
   - Target – Gets added to network by initiator
2. Network Steering – All nodes support network steering.
   - Node not on a network – Action of searching for and joining an open network
   - Node on a network – It is the action of opening the network to allow new nodes to join
3. Network Formation – Ability of a node to form a network with its network security model. It is dependent on the logical device type of the node.

  - Zigbee coordinator – Forms a centralized security network
  - Zigbee router – Forms a distributed security network
4. Finding and Binding – The following are the two procedures in finding and binding:
  - Initiator endpoint – Automatically searches and establishes application connections with target endpoint by using the identify cluster with matching cluster
  - Target endpoint – Handles finding and binding requests from initiator endpoint

6.1.2 Network Security Models

A Zigbee network can support a centralized security model (centralized security network) or a distributed security model (distributed security network). All devices except Zigbee coordinator are able to join a network supporting either model or adapt to the security conditions of the network they are joining. For more details, refer to the *Zigbee Specification Revision 22 1.0* (05-3474-22).

Centralized Security Network

A centralized security network is a Zigbee network formed by a Zigbee coordinator with the functionality of a trust center. The trust center authenticates each node that joins such a network before it can operate on the network. After creating the centralized network, the Zigbee coordinator device must not attempt to join another network.

Default Global Trust Center Link Key

A link key that is supported by all devices, and is used to join a centralized security network if there is no other specific link.
In a centralized network, use the following keys to allow the devices to join.

- Global Trust Center Link Key – Use this link key for joining centralized security networks. The value of the key is `0x5a 0x69 0x67 0x42 0x65 0x65 0x41 0x6c 0x6c 0x69 0x61 0x6e 0x63 0x65 0x30 0x39`.
- Install code link key – Is the link key derived from the install code from joining device to create unique Trust Center Link Key for joining.

Distributed Security Network

A distributed security network is a Zigbee network formed by a Zigbee router and does not have a trust center. The parent authenticates each node that joins such a network before it can operate on the network. A node designated as having a logical device type of a Zigbee router can attempt to join an existing centralized or distributed security network. However, a Zigbee router cannot form a centralized security network but can form a distributed security network. A node designated as having a logical device type of a Zigbee end device can attempt to join an existing centralized or distributed security network.

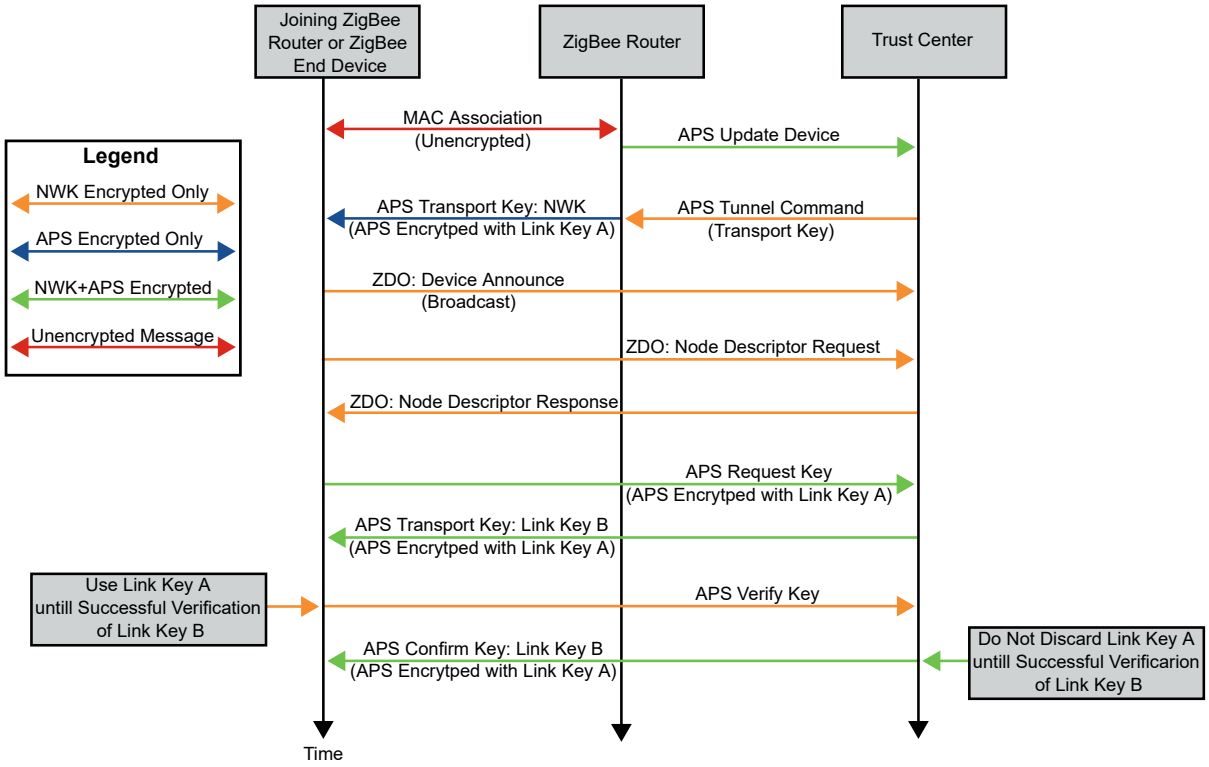
APL Layer Security

- `Transport Key` service – Supports secured means to transport a key to another device or other devices. The secured transport-key command provides a means to transport link or network key from a key source (for example, the trust center) to other devices.
- `Request Key` service – Supports a secure means for a device to request an end-to-end application link key or Trust Center Link Key from the trust center.
- `Verify Key` service – Supports a secure means for a device to verify that the device and the trust center agree on the current value of the device's link key.
- `Confirm Key` service – Supports a secure means for a trust center to confirm a previous request to verify a link key.

For more details, refer to the *PRO Base Device Behavior Specification* (3.0.1).

Trust Center Link Key Exchange Procedure

Figure 6-1. Trust Center Link Key Exchange Procedure Sequence Chart



For more details, refer to the *PRO Base Device Behavior Specification* (3.0.1).

6.1.3 Zigbee Device Profile (ZDP)

Device discovery

The device discovery mechanism provides the ability for a device to discover the identity of other devices on the PAN. The 64-bit IEEE address and the 16-bit network address both support device discovery.

- Device announcement – Enables the Zigbee devices on the network to notify other Zigbee devices that the device has joined or re-joined the network. It also helps in identifying the device's 64-bit IEEE address, new 16-bit NWK address and informing the remote devices of the capability of the Zigbee device. The destination addressing on this primitive is broadcast to all devices for which `macRxOnWhenIdle = True`. For more details, refer to the *Zigbee Specification Revision 22 1.0* (05-3474-22).

Service Discovery

The devices use the service discovery process to discover the capabilities of another device or to identify other devices that support similar services (clusters). After service discovery, the device becomes aware of the endpoints and addresses of devices supporting the same clusters. To accomplish this process, issue a query for each endpoint on a given device or by using a match service feature (either broadcast or unicast). The service discovery facility defines and utilizes various descriptors to outline the capabilities of a device. Service discovery is implemented within the Zigbee device object. For more details, refer to the *Zigbee Specification Revision 22 1.0* (05-3474-22).

Service discovery is the process by which a device in a Zigbee network identifies other devices that support similar services (clusters). After service discovery, the device becomes aware of the endpoints and addresses of devices supporting the same clusters.

- Node descriptor – Contains information about the capabilities of the Zigbee node. The local device generates the service discovery mechanism, which likes to get the node descriptor of a remote device. This packet can be unicast either to the remote device itself or to an alternative device that contains the discovery information of the remote device.

- Simple descriptor – Allows an inquiring device to get the cluster details for the supplied endpoint. This packet can be unicast.

6.1.4 Zigbee Cluster Library Specification (ZCL)

Attributes Reporting

Reporting a cluster's attribute signifies returning the value of a particular cluster attribute to the remote endpoint supporting this cluster using a specific ZCL attribute report command.

Attribute reporting starts after a device successfully joins a Zigbee network and completes service discovery. After service discovery, the device knows the endpoints and addresses of devices supporting the same clusters as it does. The device acting as a server cluster can send the periodic reports to clients supporting the same cluster.

For more details, refer to the *ZigBee Alliance Cluster Library Specification Revision 8* (075123).

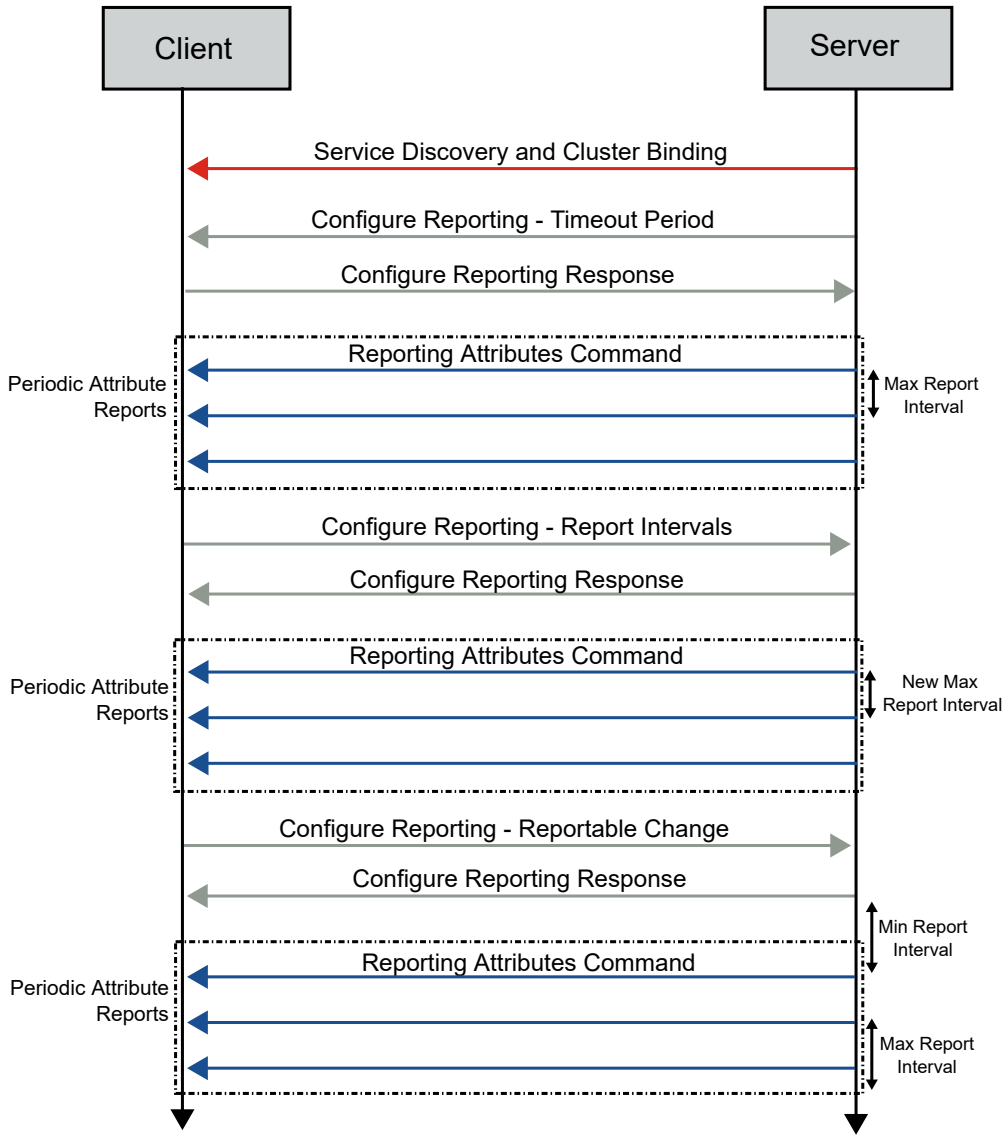
Periodic Reporting

The following are the types of periodic reporting:

- Automatic reporting – The user can configure the reporting intervals of the cluster from the application at compile-time or at run-time. The Zigbee stack application sends out the periodic reports once every maximum reporting interval period.
- Reporting on attribute value change.
- Manual reporting – The application can send out a report at any time using the attribute request via ZCL layer.

For more details, refer to the *AT08550: ZigBee Attribute Reporting Application Note* (42334).

Figure 6-2. Attribute Reporting – Typical Packet Exchange Sequence



6.2 Zigbee Coordinator

In the Zigbee centralized network, the Zigbee coordinator forms the network. Other routers and end-devices can enter after forming the network.

The following sections elaborate the association, commissioning, finding and binding, attribute reporting and security key exchange procedure of Zigbee coordinator device type (with Zigbee router).

6.2.1 Commissioning

6.2.1.1 Centralized Network Formation and Network Steering by Zigbee Coordinator/Combined Interface

The following figure illustrates the scenario of network formation by a Zigbee coordinator/combined interface device with network address `0x0000` and network steering to allow other devices to join the network.

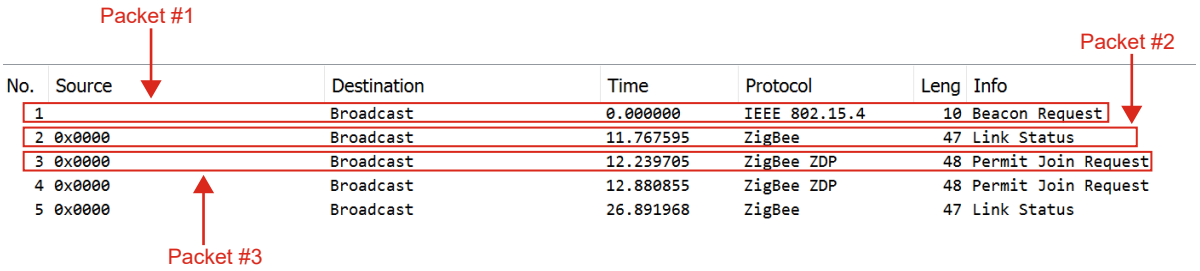
After invoking the network formation commissioning procedure, the coordinator sends a `Beacon Request` packet (see packet `#1`) followed by the MAC association process. For more details, refer to 5.2. MAC Association.

After the coordinator forms the network, it sends the `Link Status` (see packet `#2`). For more details, refer to 5.4. Network (NWK) Link Status Frame.

The permit-joining ZDP request is to provide a joining permit to the target node by MAC association. Send the request as a unicast command to just one node or as a broadcast command. Permit the joining via MAC association for a given interval in seconds or forbidden permanently, depending on the payload's `Permit Duration` field. This field specifies the duration of time, starting from the moment of the request's reception, when joining by association is permitted.

Packet `#3` shows the `Permit Join Request`, and the coordinator device sends the request as a broadcast packet. For more details, refer to 5.2. MAC Association.

Figure 6-3. Network Formation and Network Steering by ZigBee Coordinator



No.	Source	Destination	Time	Protocol	Leng	Info
1		Broadcast	0.000000	IEEE 802.15.4	10	Beacon Request
2	0x0000	Broadcast	11.767595	ZigBee	47	Link Status
3	0x0000	Broadcast	12.239705	ZigBee ZDP	48	Permit Join Request
4	0x0000	Broadcast	12.880855	ZigBee ZDP	48	Permit Join Request
5	0x0000	Broadcast	26.891968	ZigBee	47	Link Status

The permit duration of the permit duration packet is `Duration: 180`. The router/end device must join the network via MAC association within 180s.

To open the network after 180s and allow other devices to join, the user must input the following commands to the coordinator, before commissioning is initiated in another device:

- `setPermitJoin 180` – Opens the network for the next 180s
- `invokeCommissioning 8 0` – Opens the network for the finding and binding procedure

Figure 6-4. Permit Join Packet of Coordinator

```
> Frame 3: 48 bytes on wire (384 bits), 46 bytes captured (368 bits) on interface \\.\pipe\Atmel_Wireshark, id 0
> IEEE 802.15.4 Data, Dst: Broadcast, Src: 0x0000
> ZigBee Network Layer Data, Dst: Broadcast, Src: 0x0000
> ZigBee Application Support Layer Data, Dst Endpt: 0, Src Endpt: 0
v ZigBee Device Profile, Permit Join Request
    Sequence Number: 0
    Duration: 180
    Significance: 1
```

MAC Association – If any router/end device tries to join the network through `Beacon Request`, the coordinator sends the `Beacon` frame (see packet `#7`). Then, the device joins through MAC `Association Request`, and the coordinator responds with `Association Response` (see packet `#12`). The following figure illustrates the MAC association – coordinator. For more details, refer to 5.2. MAC Association.

Figure 6-5. MAC Association – Coordinator

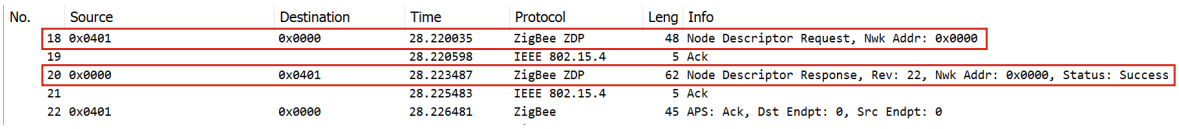
No.	Source	Destination	Time	Protocol	Leng	Info
3	0x0000	Broadcast	12.239705	ZigBee ZDP	48	Permit Join Request
4	0x0000	Broadcast	12.880855	ZigBee ZDP	48	Permit Join Request
5	0x0000	Broadcast	26.891968	ZigBee	47	Link Status
6		Broadcast	27.499082	IEEE 802.15.4	10	Beacon Request
7	0x0000		27.501054	ZigBee	28	Beacon, Src: 0x0000, EPID: 00:00:00_00:00:0d:ee:b1
8	00:00:00:00:00:0d:ee:b3	0x0000	27.639432	IEEE 802.15.4	21	Association Request, FFD
9			27.641427	IEEE 802.15.4	5	Ack
10	00:00:00:00:00:0d:ee:b3	0x0000	28.137568	IEEE 802.15.4	18	Data Request
11			28.138565	IEEE 802.15.4	5	Ack
12	00:00:00:00:00:0d:ee:b1	00:00:00:00:00:0d:ee:b3	28.141603	IEEE 802.15.4	27	Association Response, PAN: 0x2be2 Addr: 0x0401
13			28.142556	IEEE 802.15.4	5	Ack

6.2.2 Service Discovery

Node descriptor – Router/end device requests the node descriptor during the initialization procedure before finding and binding. For more details on the node descriptor, refer to 6.1.3. Zigbee Device Profile (ZDP).

- Packet #18 – Shows the Node Descriptor Request from router node
- Packet #20 – Shows the Node Descriptor Response from coordinator node

Figure 6-6. Node Descriptor

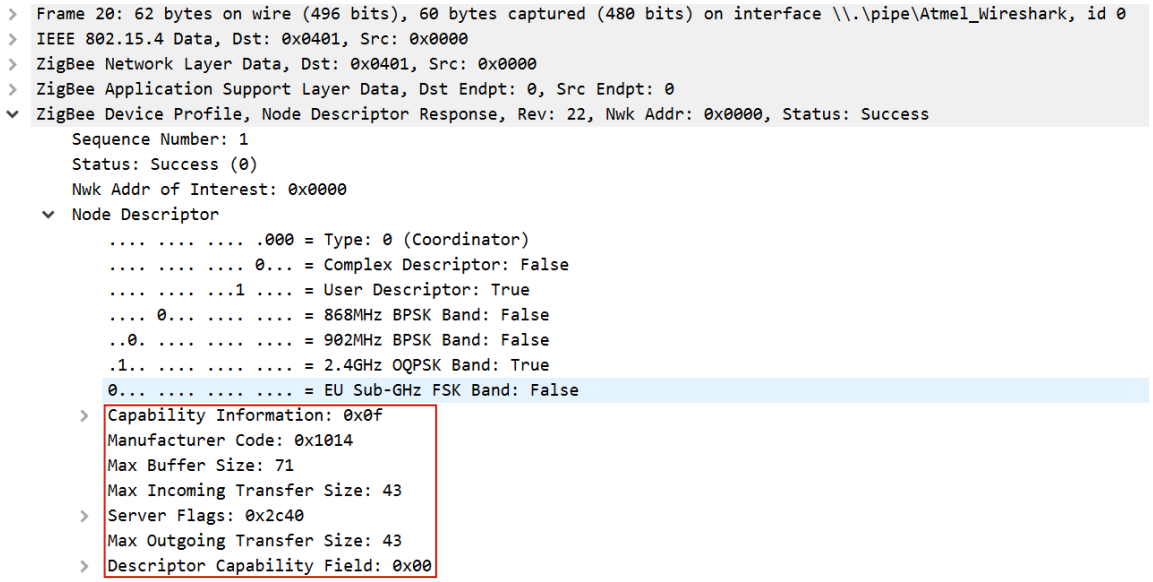


No.	Source	Destination	Time	Protocol	Leng	Info
18	0x0401	0x0000	28.220035	ZigBee ZDP	48	Node Descriptor Request, Nwk Addr: 0x0000
19			28.220598	IEEE 802.15.4	5	Ack
20	0x0000	0x0401	28.223487	ZigBee ZDP	62	Node Descriptor Response, Rev: 22, Nwk Addr: 0x0000, Status: Success
21			28.225483	IEEE 802.15.4	5	Ack
22	0x0401	0x0000	28.226481	ZigBee	45	APS: Ack, Dst Endpt: 0, Src Endpt: 0

The following figure illustrates the Node Descriptor Response from a coordinator device, where, under ZigBee Device Profile, the user can see the following information of the coordinator node:

- Capability Information
- Max Buffer Size
- Server Flags
- Descriptor Capability Field

Figure 6-7. Node Descriptor Response



```
> Frame 20: 62 bytes on wire (496 bits), 60 bytes captured (480 bits) on interface \\.\pipe\Atmel_Wireshark, id 0
> IEEE 802.15.4 Data, Dst: 0x0401, Src: 0x0000
> ZigBee Network Layer Data, Dst: 0x0401, Src: 0x0000
> ZigBee Application Support Layer Data, Dst Endpt: 0, Src Endpt: 0
v ZigBee Device Profile, Node Descriptor Response, Rev: 22, Nwk Addr: 0x0000, Status: Success
    Sequence Number: 1
    Status: Success (0)
    Nwk Addr of Interest: 0x0000
  v Node Descriptor
      .... .... .... .000 = Type: 0 (Coordinator)
      .... .... .... 0... = Complex Descriptor: False
      .... .... ...1 .... = User Descriptor: True
      .... 0... .... .... = 868MHz BPSK Band: False
      ..0. .... .... .... = 902MHz BPSK Band: False
      .1.. .... .... .... = 2.4GHz OQPSK Band: True
      0... .... .... .... = EU Sub-GHz FSK Band: False
    > Capability Information: 0x0f
      Manufacturer Code: 0x1014
      Max Buffer Size: 71
      Max Incoming Transfer Size: 43
    > Server Flags: 0x2c40
      Max Outgoing Transfer Size: 43
    > Descriptor Capability Field: 0x00
```

Simple Descriptor – After receiving the Identify Query Response from the coordinator, the router identifies the target endpoint and sends a simple descriptor request to the target endpoint (coordinator). Packets #53 and #55 are Simple Descriptor Request and Simple Descriptor Response from router and coordinator devices, respectively. For more details, refer to the *Zigbee Specification Revision 22 1.0* (05-3474-22).

Figure 6-8. Simple Descriptor – Coordinator and Router

No.	Source	Destination	Time	Protocol	Length	Info
47	0x3c08	Broadcast	270.393782	ZigBee HA	46	ZCL Identify: Identify Query, Seq: 0
48	0x0000	0x3c08	270.399198	ZigBee HA	48	ZCL Identify: Identify Query Response, Seq: 0
49			270.401198	IEEE 802.15.4	3	Ack
50	0x3c08	0x0000	270.403622	ZigBee	43	APS: Ack, Dst Endpt: 20, Src Endpt: 35
51			270.405470	IEEE 802.15.4	3	Ack
52	0x3c08	Broadcast	270.436398	ZigBee HA	46	ZCL Identify: Identify Query, Seq: 0
53	0x3c08	0x0000	270.442286	ZigBee ZDP	47	Simple Descriptor Request, Nwk Addr: 0x0000, Endpoint: 20
54			270.444253	IEEE 802.15.4	3	Ack
55	0x0000	0x3c08	270.446654	ZigBee ZDP	102	Simple Descriptor Response, Nwk Addr: 0x0000, Status: Success
56			270.450381	IEEE 802.15.4	3	Ack
57	0x3c08	0x0000	270.453318	ZigBee	43	APS: Ack, Dst Endpt: 0, Src Endpt: 0
58			270.455157	IEEE 802.15.4	3	Ack

The following figure illustrates the `Simple Descriptor Response` with the list of supported input and output clusters of the coordinator.

Figure 6-9. Simple Descriptor Response

```
ZigBee Device Profile, Simple Descriptor Response, Nwk Addr: 0x0000, Status: Success
    Sequence Number: 3
    Status: Success (0)
    Nwk Addr of Interest: 0x0000
    Simple Descriptor Length: 54
  Simple Descriptor
      Endpoint: 20
      Profile: Home Automation (0x0104)
      Application Device: Unknown (0x0007)
      Application Version: 0x0001
      Input Cluster Count: 6
    Input Cluster List
      Output Cluster Count: 17
    Output Cluster List
```

6.2.3 Finding and Binding

The following are configured as the target endpoint and initiator endpoint:

- Target endpoint – Zigbee coordinator/combined interface
- Initiator endpoint – Zigbee router/extended lights

Coordinator as a target endpoint receives `Identify Query` request from router, for which the coordinator sends `Identify Query Response` to the initiator endpoint (router).

The following figure illustrates packets `#57` and `# 58` are `Identify Query` request and `Identify Query Response` from router and coordinator devices respectively.

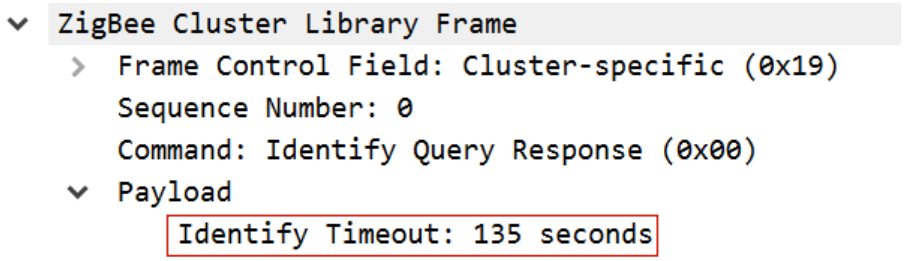
The target endpoint identifies itself for a finite duration, then handles subsequent finding and binding requests from an initiator endpoint.

Figure 6-10. Identify Query

57	0x0401	Broadcast	214.172086	ZigBee HA	48 ZCL Identify: Identify Query, Seq: 0
58	0x0000	0x0401	214.176690	ZigBee HA	50 ZCL Identify: Identify Query Response, Seq: 0
59			214.177686	IEEE 802.15.4	5 Ack
60	0x0401	0x0000	214.179998	ZigBee	45 APS: Ack, Dst Endpt: 20, Src Endpt: 35
61			214.179998	IEEE 802.15.4	5 Ack

The following figure illustrates the `Identify Timeout: 135 seconds` for target endpoint.

Figure 6-11. Identify Timeout



When the decrementing `Identify Timeout` attribute reaches zero, the target device terminates the finding and binding procedure for a target endpoint.

6.2.4 Reporting

As a client, the device is capable of making device discovery, service discovery, binding or network management requests. As a server, the device services these requests and responds to them. The client and server roles are non-exclusive, and a given device can act as both client and server.

The device profile describes devices in one of two configurations:

- Client – Issues requests to the server via device profile messages
- Server – Issues responses to the client that initiated the device profile message

The following table provides details about the client/server clusters available for the combined interface device type in the Microchip Zigbee stack. For more details, refer to the *ZigBee Alliance Cluster Library Specification Revision 8* (075123). For more details regarding mandatory or optional clusters for a specific device type, refer to the *Matter Device Library Specification* (1.0).

Note: The combined interface device type is only supported in the Microchip Zigbee stack.

Table 6-1. Supported Clusters – Combined Interface

Device Type	Server Cluster ID	Server Clusters	Client Cluster ID	Client Clusters
Combined interface	0x0000	Basic	0x0000	Basic
	0x0003	Identify	0x0003	Identify
	0x0004	Groups	0x0004	Groups
	0x000A	Time	0x0005	Scenes
	0x0501	IAS ACE	0x0006	On/Off
	0x0008	Level control	0x0009	Alarms
	0x0300	Color control	0x0201	Thermostat
	—	—	0x0202	Fan control
	—	—	0x0406	Occupancy sensing
	—	—	0x0400	Illuminance measurement
	—	—	0x0402	Temperature measurement
	—	—	0x0204	Thermostat UI
	—	—	0x0405	Water content measurement
	—	—	0x0500	IAS zone

In this scenario, the coordinator/combined interface is configured as a target endpoint; therefore, the coordinator device does not report any attributes. It monitors the attributes reported by routers/end devices.

6.2.5 Security

The Zigbee coordinator/combined interface device with address `0x0000` acts as a trust center, and the device with address `0x0401` acts as a Zigbee router (see the following figure). For details on the centralized security mechanism, refer to 6.1.2. Network Security Models.

As per Figure 6-1, MAC association packets were unencrypted. After completion of the association process:

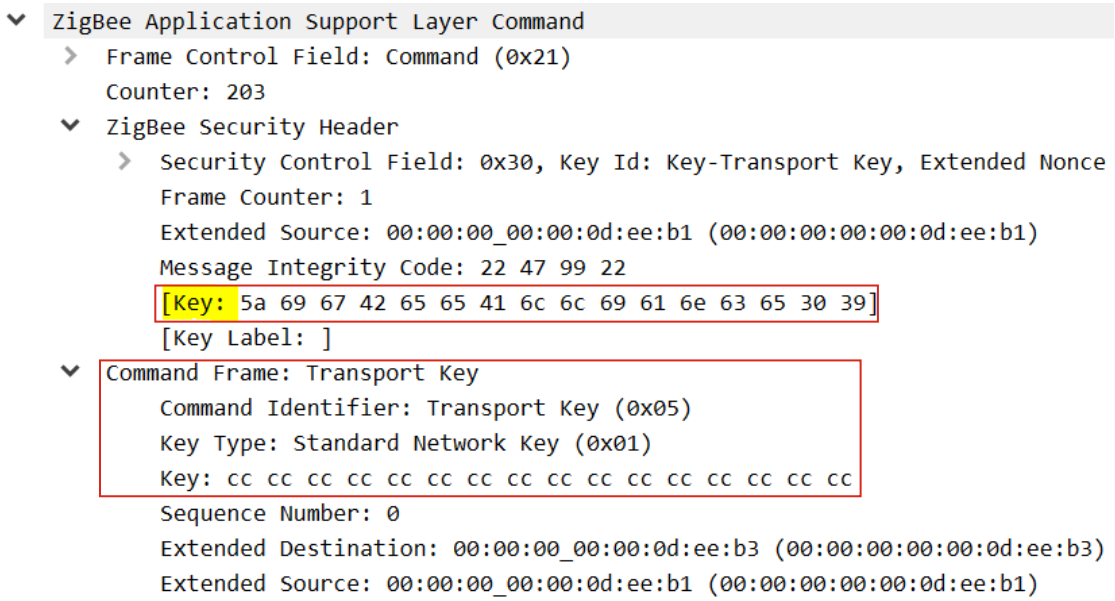
1. The trust center sends the `Transport Key` (coordinator with address `0x0000`) from which the joining device receives the link key (router-`0x0401`) (see packet `#14`). The `APS` frame carrying the transport key is encrypted with Link Key A.
2. The joined device (router) performs the device announcement (see packets `#16` and `#17`).
3. Node descriptor exchange happens between coordinator and router as part of the initialization procedure (see packets `#18` to `#22`).
4. Packet `#23` shows the router sending the request key to the trust center as a request for link `Key` B. Link `Key` A secures the APS frame carrying this request key.
5. The trust center transports (packet `#25`) the requested key via `Transport Key` with `APS` encryption by Link `Key` A.
6. Packet `#27` shows `Verify Key`, which ensures that the trust center and joined device agree on the same key.
7. Packet `#29` shows the `Confirm Key`, which permits the trust center to confirm a previous request to verify a link key.

Figure 6-12. Trust Center Key Exchange Centralized Network

	No.	Source	Destination	Time	Protocol	Leng	Info
	6		Broadcast	27.499082	IEEE 802.15.4	10	Beacon Request
	7	0x0000		27.501054	ZigBee	28	Beacon, Src: 0x0000, EPID: 00:00:00_00:00:0d:ee:b1
	8	00:00:00:00:00:0d:ee:b3	0x0000	27.639432	IEEE 802.15.4	21	Association Request, FFD
	9			27.641427	IEEE 802.15.4	5	Ack
	10	00:00:00:00:00:0d:ee:b3	0x0000	28.137568	IEEE 802.15.4	18	Data Request
	11			28.138565	IEEE 802.15.4	5	Ack
	12	00:00:00:00:00:0d:ee:b1	00:00:00:00:00:0d:…	28.141603	IEEE 802.15.4	27	Association Response, PAN: 0x2be2 Addr: 0x0401
	13			28.142556	IEEE 802.15.4	5	Ack
1	14	0x0000	0x0401	28.146545	ZigBee	73	Transport Key
	15			28.148174	IEEE 802.15.4	5	Ack
2	16	0x0401	Broadcast	28.154162	ZigBee ZDP	57	Device Announcement, Nwk Addr: 0x0401, Ext Addr: 00:00:00_00:00:0d:ee:b3
	17	0x0401	Broadcast	28.191112	ZigBee ZDP	57	Device Announcement, Nwk Addr: 0x0401, Ext Addr: 00:00:00_00:00:0d:ee:b3
	18	0x0401	0x0000	28.220035	ZigBee ZDP	48	Node Descriptor Request, Nwk Addr: 0x0000
	19			28.220598	IEEE 802.15.4	5	Ack
3	20	0x0000	0x0401	28.223487	ZigBee ZDP	62	Node Descriptor Response, Rev: 22, Nwk Addr: 0x0000, Status: Success
	21			28.225483	IEEE 802.15.4	5	Ack
	22	0x0401	0x0000	28.226481	ZigBee	45	APS: Ack, Dst Endpt: 0, Src Endpt: 0
4	23	0x0401	0x0000	28.230471	ZigBee	66	Request Key
	24			28.232511	IEEE 802.15.4	5	Ack
5	25	0x0000	0x0401	28.237494	ZigBee	90	Transport Key
	26			28.239448	IEEE 802.15.4	5	Ack
6	27	0x0401	0x0000	28.242440	ZigBee	65	Verify Key
	28			28.244433	IEEE 802.15.4	5	Ack
7	29	0x0000	0x0401	28.247426	ZigBee	67	Confirm Key, SUCCESS
	30			28.249421	IEEE 802.15.4	5	Ack

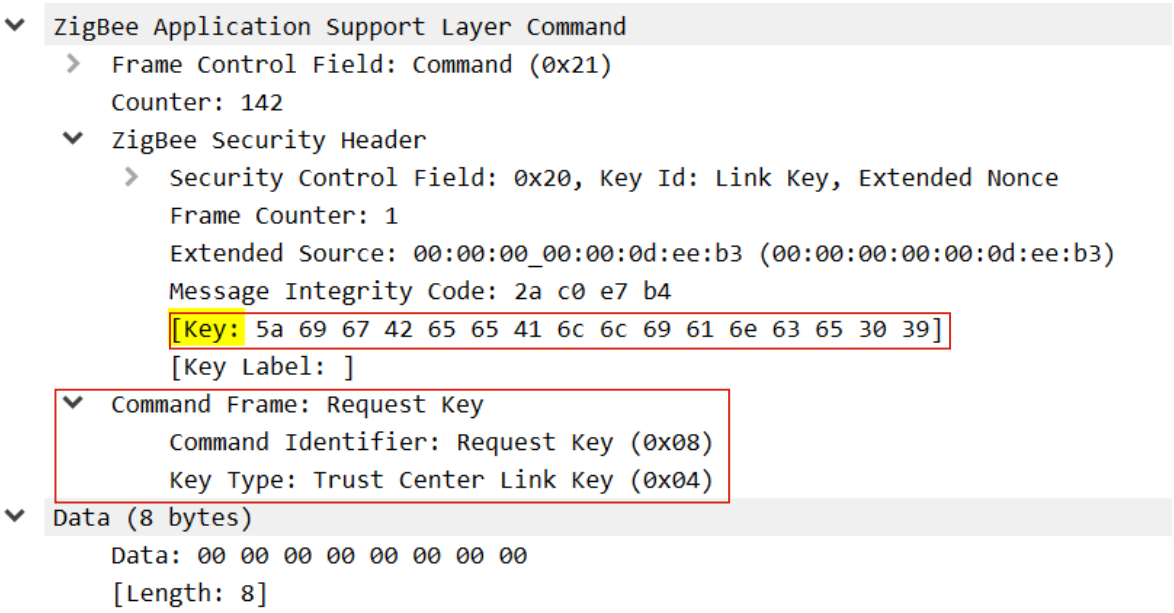
The following figure illustrates the `Transport Key`, where Link `Key` A (5a 69 67 42 65 65 41 6c 6c 69 61 6e 63 65 30 39) is highlighted, which encrypts the `APS` layer. By default, the network key is used for cluster commands. The following figure highlights the network key, `Key: cc cc cc cc cc cc cc cc cc cc cc cc cc cc cc cc`.

Figure 6-13. APS Tunnel Transport Key



The following figure illustrates the `Request Key`, where Link `Key` A (5a 69 67 42 65 65 41 6c 6c 69 61 6e 63 65 30 39) is highlighted, that encrypts the APS layer.

Figure 6-14. Request Key



The following figure illustrates the `Transport Key`, where Link `Key` A (5a 69 67 42 65 65 41 6c 6c 69 61 6e 63 65 30 39) is highlighted, that encrypts the APS layer. The following figure illustrates the Link `Key` B (fb 40 45 17 7a 0a bc 68 e3 35 ce 4b 93 12 63 0a), which is being transported from the trust center to the router.

Figure 6-15. Transport Key

```
ZigBee Application Support Layer Command
  Frame Control Field: Command (0x21)
  Counter: 206
  ZigBee Security Header
    Security Control Field: 0x38, Key Id: Key-Load Key, Extended Nonce
    Frame Counter: 2
    Extended Source: 00:00:00_00:00:0d:ee:b1 (00:00:00:00:00:0d:ee:b1)
    Message Integrity Code: 88 f7 fc 08
    [Key: 5a 69 67 42 65 65 41 6c 6c 69 61 6e 63 65 30 39]
    [Key Label: ]
  Command Frame: Transport Key
    Command Identifier: Transport Key (0x05)
    Key Type: Trust Center Link Key (0x04)
    Key: fb 40 45 17 7a 0a bc 68 e3 35 ce 4b 93 12 63 0a
    Extended Destination: 00:00:00_00:00:0d:ee:b3 (00:00:00:00:00:0d:ee:b3)
    Extended Source: 00:00:00_00:00:0d:ee:b1 (00:00:00:00:00:0d:ee:b1)
```

6.3 Zigbee Router

The Zigbee router can form a distributed network. If it does not form a distributed network, it can join a centralized network formed by the Zigbee coordinator.

The following section elaborates the association, commissioning, finding and binding, attribute reporting and security key exchange procedure of the Zigbee router device type (with the Zigbee coordinator).

6.3.1 Commissioning

6.3.1.1 Network Steering by Zigbee Router/Extended Light

In the following figure, the packet `#6` shows the network steering procedure starts with the router device broadcasting the `Beacon Request`. After receiving the beacon frames from the coordinator, the router device joins through MAC association.

MAC Association – The device tries to join a network through MAC association (as joining unknown network for the first time). Packet `#8` to `#12` shows the MAC association procedure. For more details on MAC Association, refer to 5.2. MAC Association.

Figure 6-16. Network Steering – Router

No.	Source	Destination	Time	Protocol	Leng	Info
6		Broadcast	27.499082	IEEE 802.15.4	10	Beacon Request
7	0x0000		27.501054	ZigBee	28	Beacon, Src: 0x0000, EPID: 00:00:00_00:00:0d:ee:b1
8	00:00:00:00:00:0d:ee:b3	0x0000	27.639432	IEEE 802.15.4	21	Association Request, FFD
9			27.641427	IEEE 802.15.4	5	Ack
10	00:00:00:00:00:0d:ee:b3	0x0000	28.137568	IEEE 802.15.4	18	Data Request
11			28.138565	IEEE 802.15.4	5	Ack
12	00:00:00:00:00:0d:ee:b1	00:00:00:00:00:0d:…	28.141603	IEEE 802.15.4	27	Association Response, PAN: 0x2be2 Addr: 0x0401
13			28.142556	IEEE 802.15.4	5	Ack

6.3.1.2 Distributed Network Formation by Zigbee Router/Extended Light

Zigbee routers can form distributed networks. After successfully forming the distributed network, broadcast the device announcement packets and permit join requests to notify other routers/end devices that are searching for a network to join. After sending the end device/router a beacon request, the parent router responds with beacon frame, which leads to MAC association. After successful MAC association, the end device/router joins the network. The link key is communicated between the parent router and the end device/router through transport key for security purpose. For more details, refer to 6.1.2. Network Security Models.

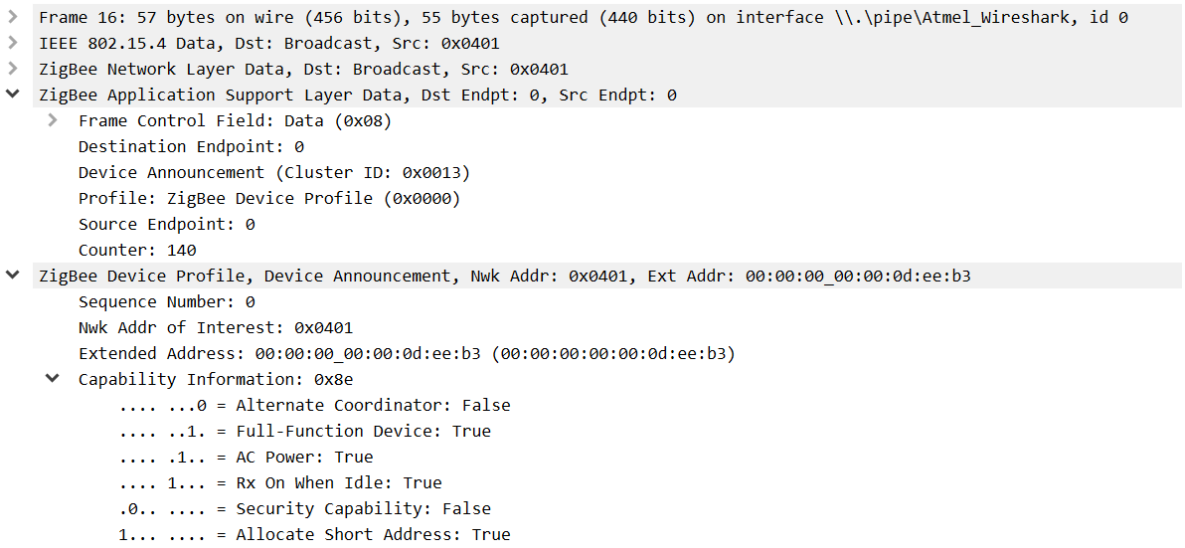
6.3.2 Device Discovery

Device Announcement – The Zigbee router device with NWK address `0x0401` broadcasts the packets `#16` and `#17` that show the `Device Announcement`. For more details, refer to 6.1.3. Zigbee Device Profile (ZDP).

Figure 6-17. Device Announcement

No.	Source	Destination	Time	Protocol	Leng	Info
16	0x0401	Broadcast	28.154162	ZigBee ZDP	57	Device Announcement, Nwk Addr: 0x0401, Ext Addr: 00:00:00_00:00:0d:ee:b3
17	0x0401	Broadcast	28.191112	ZigBee ZDP	57	Device Announcement, Nwk Addr: 0x0401, Ext Addr: 00:00:00_00:00:0d:ee:b3

Figure 6-18. Device Announcement



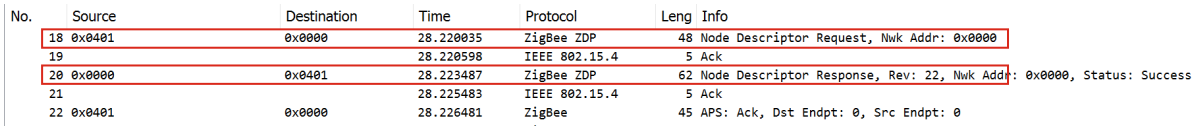
```
Frame 16: 57 bytes on wire (456 bits), 55 bytes captured (440 bits) on interface \\.\pipe\Atmel_Wireshark, id 0
IEEE 802.15.4 Data, Dst: Broadcast, Src: 0x0401
ZigBee Network Layer Data, Dst: Broadcast, Src: 0x0401
ZigBee Application Support Layer Data, Dst Endpt: 0, Src Endpt: 0
    Frame Control Field: Data (0x08)
    Destination Endpoint: 0
    Device Announcement (Cluster ID: 0x0013)
    Profile: ZigBee Device Profile (0x0000)
    Source Endpoint: 0
    Counter: 140
ZigBee Device Profile, Device Announcement, Nwk Addr: 0x0401, Ext Addr: 00:00:00_00:00:0d:ee:b3
    Sequence Number: 0
    Nwk Addr of Interest: 0x0401
    Extended Address: 00:00:00_00:00:0d:ee:b3 (00:00:00:00:00:0d:ee:b3)
    Capability Information: 0x8e
        .... ...0 = Alternate Coordinator: False
        .... ..1. = Full-Function Device: True
        .... .1.. = AC Power: True
        .... 1... = Rx On When Idle: True
        .0.. .... = Security Capability: False
        1... .... = Allocate Short Address: True
```

6.3.3 Service Discovery

Node Descriptor – The router/end device requests the node descriptor during the initialization procedure before finding and binding to discover the capability information and other information of the coordinator device in the network. For more details, refer to the *Zigbee Specification Revision 22 1.0* (05-3474-22).

The following figure illustrates packets `#18` and `#20` as the `Node Descriptor Request` and `Node Descriptor Response` from the router and coordinator nodes, respectively.

Figure 6-19. Node Descriptor



No.	Source	Destination	Time	Protocol	Leng	Info
18	0x0401	0x0000	28.220035	ZigBee ZDP	48	Node Descriptor Request, Nwk Addr: 0x0000
19			28.220598	IEEE 802.15.4	5	Ack
20	0x0000	0x0401	28.223487	ZigBee ZDP	62	Node Descriptor Response, Rev: 22, Nwk Addr: 0x0000, Status: Success
21			28.225483	IEEE 802.15.4	5	Ack
22	0x0401	0x0000	28.226481	ZigBee	45	APS: Ack, Dst Endpt: 0, Src Endpt: 0

The following figure illustrates the `Node Descriptor Response` from a coordinator device. The user can see the following under `ZigBee Device Profile`:

- `Capability Information` of the coordinator node
- `Max Buffer Size`
- `Server Flags`
- `Descriptor Capability Field`

Figure 6-20. Node Descriptor Response

```
> Frame 20: 62 bytes on wire (496 bits), 60 bytes captured (480 bits) on interface \\.\pipe\Atmel_Wireshark, id 0
> IEEE 802.15.4 Data, Dst: 0x0401, Src: 0x0000
> ZigBee Network Layer Data, Dst: 0x0401, Src: 0x0000
> ZigBee Application Support Layer Data, Dst Endpt: 0, Src Endpt: 0
v ZigBee Device Profile, Node Descriptor Response, Rev: 22, Nwk Addr: 0x0000, Status: Success
    Sequence Number: 1
    Status: Success (0)
    Nwk Addr of Interest: 0x0000
  v Node Descriptor
        .... .... .... .000 = Type: 0 (Coordinator)
        .... .... .... 0... = Complex Descriptor: False
        .... .... ...1 .... = User Descriptor: True
        .... 0... .... .... = 868MHz BPSK Band: False
        ..0. .... .... .... = 902MHz BPSK Band: False
        .1.. .... .... .... = 2.4GHz OQPSK Band: True
        0... .... .... .... = EU Sub-GHz FSK Band: False
      > Capability Information: 0x0f
        Manufacturer Code: 0x1014
        Max Buffer Size: 71
        Max Incoming Transfer Size: 43
      > Server Flags: 0x2c40
        Max Outgoing Transfer Size: 43
      > Descriptor Capability Field: 0x00
```

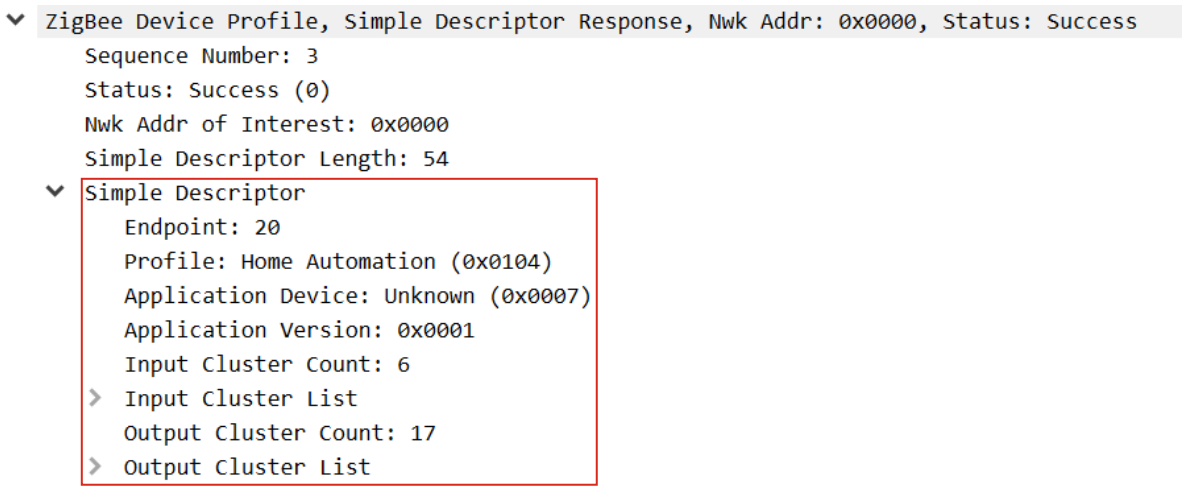
Simple Descriptor – After receiving the `Identify Query Response` from the coordinator, the router identifies the target endpoint and sends a simple descriptor request to the target endpoint (coordiantor). Packets `#53` and `#55` are `Simple Descriptor Request` and `Simple Descriptor Response` from router and coordinator devices, respectively. For more details, refer to the *Zigbee Specification Revision 22 1.0* (05-3474-22).

Figure 6-21. Simple Descriptor – Coordinator and Router

No.	Source	Destination	Time	Protocol	Length	Info
47	0x3c08	Broadcast	270.393782	ZigBee HA	46	ZCL Identify: Identify Query, Seq: 0
48	0x0000	0x3c08	270.399198	ZigBee HA	48	ZCL Identify: Identify Query Response, Seq: 0
49			270.401198	IEEE 802.15.4	3	Ack
50	0x3c08	0x0000	270.403622	ZigBee	43	APS: Ack, Dst Endpt: 20, Src Endpt: 35
51			270.405470	IEEE 802.15.4	3	Ack
52	0x3c08	Broadcast	270.436398	ZigBee HA	46	ZCL Identify: Identify Query, Seq: 0
53	0x3c08	0x0000	270.442286	ZigBee ZDP	47	Simple Descriptor Request, Nwk Addr: 0x0000, Endpoint: 20
54			270.444253	IEEE 802.15.4	3	Ack
55	0x0000	0x3c08	270.446654	ZigBee ZDP	102	Simple Descriptor Response, Nwk Addr: 0x0000, Status: Success
56			270.450381	IEEE 802.15.4	3	Ack
57	0x3c08	0x0000	270.453318	ZigBee	43	APS: Ack, Dst Endpt: 0, Src Endpt: 0
58			270.455157	IEEE 802.15.4	3	Ack

The following figure illustrates the `Simple Descriptor Response` with the list of supported input and output clusters of the coordinator.

Figure 6-22. Simple Descriptor Response



6.3.4 Finding and Binding

The user can configure the target endpoint/initiator endpoint as the following:

- Zigbee router/extended lights – As the initiator endpoint
- Zigbee coordinator/combined interface – As the target endpoint

The following figure illustrates packets `#47` and `#48` as the `Identify Query Request` and `Identify Query Response` from router and coordinator devices, respectively.

The router, as an initiator, broadcasts `Identify Query` for identifying target endpoints. After receiving the `Identify Query Response` from a target endpoint, the initiator unicasts the `Simple Descriptor Request` to the target device. The initiator endpoint, then, searches for any matching clusters between itself and the target endpoint; then, for each match found, it creates a corresponding entry in its binding table. If there is a request for group binding, the initiator endpoint configures group membership of the target endpoint.

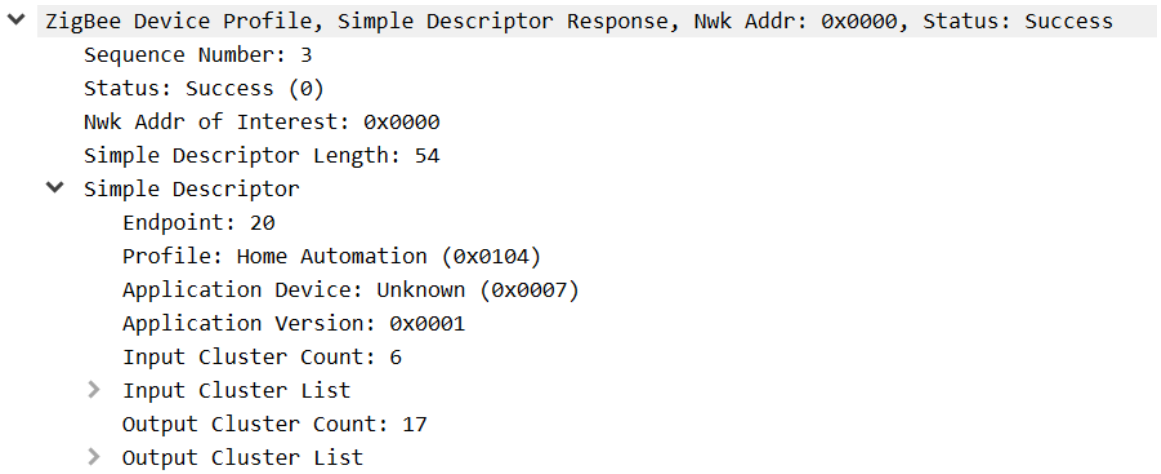
After receiving the `Identify Query Response`, the router identifies the target endpoint and requests the `Simple Descriptor`. Packets `#53` and `#55` are `Simple Descriptor Request` and `Simple Descriptor Response` from the router and coordinator devices, respectively.

Figure 6-23. Finding and Binding – Router

No.	Source	Destination	Time	Protocol	Length	Info
47	0x3c08	Broadcast	270.393782	ZigBee HA	46	ZCL Identify: Identify Query, Seq: 0
48	0x0000	0x3c08	270.399198	ZigBee HA	48	ZCL Identify: Identify Query Response, Seq: 0
49			270.401198	IEEE 802.15.4	3	Ack
50	0x3c08	0x0000	270.403622	ZigBee	43	APS: Ack, Dst Endpt: 20, Src Endpt: 35
51			270.405470	IEEE 802.15.4	3	Ack
52	0x3c08	Broadcast	270.436398	ZigBee HA	46	ZCL Identify: Identify Query, Seq: 0
53	0x3c08	0x0000	270.442286	ZigBee ZDP	47	Simple Descriptor Request, Nwk Addr: 0x0000, Endpoint: 20
54			270.444253	IEEE 802.15.4	3	Ack
55	0x0000	0x3c08	270.446654	ZigBee ZDP	102	Simple Descriptor Response, Nwk Addr: 0x0000, Status: Success
56			270.450381	IEEE 802.15.4	3	Ack
57	0x3c08	0x0000	270.453318	ZigBee	43	APS: Ack, Dst Endpt: 0, Src Endpt: 0
58			270.455157	IEEE 802.15.4	3	Ack

The following figure illustrates the `Simple Descriptor Response`, providing the details about the list of supported input and output clusters.

Figure 6-24. Simple Descriptor Response



6.3.5 Reporting

The following table provides details about the client/server clusters available for extended color light device types in the Zigbee stack. For more details, refer to the *ZigBee Alliance Cluster Library Specification Revision 8* (075123). For more details regarding mandatory or optional clusters for specific device type, refer to the *Matter Device Library Specification* (1.0).

Table 6-2. Supported Clusters – Extended Color Light

Device Type	Cluster ID	Server Clusters	Client Clusters	Attribute Identifier	Attribute Name
Extended color light	0x0000	Basic	Basic	—	—
	0x0003	Identify	Identify	—	—
	0x0004	Groups	Groups	—	—
	0x0005	Scenes	—	—	—
	0x0006	On/Off(1)	—	0x0000(1)	On/Off(1)
	0x0008	Level control(1)	—	0x0000(1)	Current level(1)
	0x0300	Color control	—	—	—

Note:

1. In this scenario, the router/extended light device reports the On/Off (0x0000) attribute of the On/Off (0x0006) cluster and the current level (0x0000) attribute of the level control (0x0008) cluster to the coordinator/ combined interface.

The extended color light is a lighting device that can be switched ON or OFF. The user can adjust the intensity of light, and the bound controller device (color controller) adjusts the color. The device supports the adjustment of color via hue/saturation, enhanced hue, color looping, XY coordinates and color temperature. In addition, the user can switch ON/OFF via a bound occupancy sensor.

Reporting Attributes – The device uses the `Report Attributes` command to report the values of one or more of its attributes to another device. Individual clusters define which attributes are to be reported and at what interval.

Figure 6-25. Report Attributes – Router

No.	Source	Destination	Time	Protocol	Leng	Info
78	0x0401	0x0000	294.270097	ZigBee HA	52	ZCL: Report Attributes, Seq: 1
79			294.271856	IEEE 802.15.4	5	Ack
80	0x0000	Broadcast	296.359236	ZigBee	50	Link Status
81	0x0401	Broadcast	296.590545	ZigBee	50	Link Status
82	0x0000	Broadcast	311.478669	ZigBee	50	Link Status
83	0x0401	Broadcast	311.633395	ZigBee	50	Link Status
84	0x0000	Broadcast	326.602218	ZigBee	50	Link Status
85	0x0401	Broadcast	326.673345	ZigBee	50	Link Status
86	0x0401	0x0000	334.276770	ZigBee HA	52	ZCL: Report Attributes, Seq: 2
87			334.277767	IEEE 802.15.4	5	Ack
88	0x0401	Broadcast	341.796805	ZigBee	50	Link Status
89	0x0000	Broadcast	356.805413	ZigBee	50	Link Status
90	0x0401	Broadcast	356.876190	ZigBee	50	Link Status
91	0x0000	Broadcast	371.928857	ZigBee	50	Link Status
92	0x0401	Broadcast	371.957362	ZigBee	50	Link Status
93	0x0401	0x0000	374.278997	ZigBee HA	52	ZCL: Report Attributes, Seq: 3
94			374.282998	IEEE 802.15.4	5	Ack
95	0x0000	Broadcast	387.090623	ZigBee	50	Link Status
96	0x0401	Broadcast	387.120543	ZigBee	50	Link Status
97	0x0000	Broadcast	402.253476	ZigBee	50	Link Status
98	0x0401	Broadcast	402.282435	ZigBee	50	Link Status

The following figure illustrates the `On/Off` attribute of `On/Off` cluster.

Figure 6-26. On/Off Cluster – On/Off Attribute

```
ZigBee Application Support Layer Data, Dst Endpt: 20, Src Endpt: 35
  Frame Control Field: Data (0x00)
      .... ..00 = Frame Type: Data (0x0)
      .... 00.. = Delivery Mode: Unicast (0x0)
      ..0. .... = Security: False
      .0.. .... = Acknowledgement Request: False
      0... .... = Extended Header: False
  Destination Endpoint: 20
  Cluster: On/Off (0x0006)
  Profile: Home Automation (0x0104)
  Source Endpoint: 35
  Counter: 148
ZigBee Cluster Library Frame, Command: Report Attributes, Seq: 2
  Frame Control Field: Profile-wide (0x18)
  Sequence Number: 2
  Command: Report Attributes (0x0a)
  Attribute Field
      Attribute: OnOff (0x0000)
      Data Type: Boolean (0x10)
      On/off Control: Off (0x00)
```

The following figure illustrates the `Current Level` attribute of `Cluster: Level Control`.

Figure 6-27. Level Control Cluster – Current Level Attribute

```
ZigBee Application Support Layer Data, Dst Endpt: 20, Src Endpt: 35
  Frame Control Field: Data (0x00)
      .... ..00 = Frame Type: Data (0x0)
      .... 00.. = Delivery Mode: Unicast (0x0)
      ..0. .... = Security: False
      .0.. .... = Acknowledgement Request: False
      0... .... = Extended Header: False
  Destination Endpoint: 20
  Cluster: Level Control (0x0008)
  Profile: Home Automation (0x0104)
  Source Endpoint: 35
  Counter: 147
ZigBee Cluster Library Frame, Command: Report Attributes, Seq: 1
  Frame Control Field: Profile-wide (0x18)
  Sequence Number: 1
  Command: Report Attributes (0x0a)
  Attribute Field
     Attribute: Current Level (0x0000)
     Data Type: 8-Bit Unsigned Integer (0x20)
     Current Level: 127
```

In this scenario, the router device and coordinator device are configured as the following:

- Router device – Reports the On/Off attribute of the On/Off cluster
- Coordinator device – The current level attribute of the level control cluster

6.3.6 Security

For security key exchange in a centralized network between trust center and router, refer to 6.2.5. Security.

6.4 Zigbee End Device

The Zigbee end device joins one of the following:

- Coordinator – Forms centralized network
- Router – Forms distributed network

After joining, the multisensor device starts the ZCL attribute reporting of sensor data, such as temperature, occupancy, light and humidity after connecting to the network.

The following section elaborates the association, commissioning, finding and binding, attribute reporting and security key exchange procedure of the Zigbee end device type (with Zigbee coordinator).

6.4.1 Commissioning

6.4.1.1 Network Steering by Zigbee End Device/Multisensor

MAC Association – The end device tries to join a network through the MAC association procedure (as joining unknown network for the first time). End device/router broadcasts `Beacon Request`, and the coordinators/routers in the network sends `Beacon Response`. The MAC association process is carried out between the end device/router and the coordinator/router. For more details, refer to 5.2. MAC Association.

Figure 6-28. Network Steering – End Device

No.	Source	Destination	Time	Protocol	Lengt	Info
6		Broadcast	36.992308	IEEE 802.15.4	10	Beacon Request
7	0x0000		36.993083	ZigBee	28	Beacon, Src: 0x0000, EPID: 00:00:00_00:00:0d:ee:b1
8	00:00:00:00:00:0d:ee:b4	0x0000	37.131442	IEEE 802.15.4	21	Association Request, RFD
9			37.132444	IEEE 802.15.4	5	Ack
10	00:00:00:00:00:0d:ee:b4	0x0000	37.631124	IEEE 802.15.4	18	Data Request
11			37.631124	IEEE 802.15.4	5	Ack
12	00:00:00:00:00:0d:ee:b1	00:00:00:00:00:0d:ee:b4	37.635136	IEEE 802.15.4	27	Association Response, PAN: 0x0733 Addr: 0x017d
13			37.635136	IEEE 802.15.4	5	Ack

6.4.2 Device Discovery

Device Announcement – Packet `#20` illustrates the Zigbee end device broadcasting the `Device Announcement` with NWK address `0x017d`. For more details, refer to 6.1.3. Zigbee Device Profile (ZDP).

Figure 6-29. Device Announcement – End Device

No.	Source	Destination	Time	Protocol	Length	Info
20	0x017d	Broadcast	37.684574	ZigBee ZDP	57	Device Announcement, Nwk Addr: 0x017d, Ext Addr: 00:00:00_00:00:0d:ee:b4

Figure 6-30. Device Announcement Packet

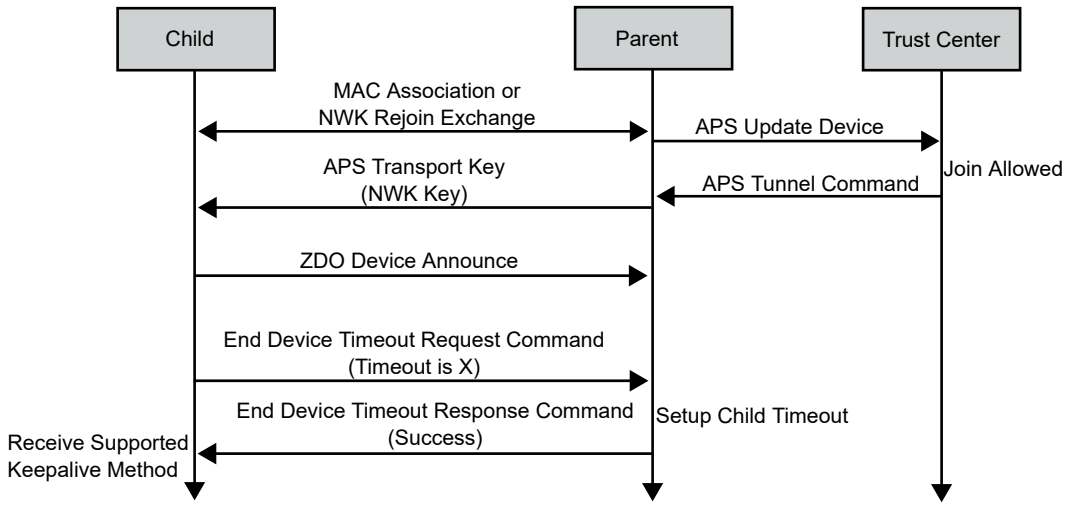
```
> Frame 20: 57 bytes on wire (456 bits), 55 bytes captured (440 bits) on interface \\.\pipe\Atmel_Wireshark, id 0
> IEEE 802.15.4 Data, Dst: 0x0000, Src: 0x017d
> ZigBee Network Layer Data, Dst: Broadcast, Src: 0x017d
> ZigBee Application Support Layer Data, Dst Endpt: 0, Src Endpt: 0
v ZigBee Device Profile, Device Announcement, Nwk Addr: 0x017d, Ext Addr: 00:00:00_00:00:0d:ee:b4
    Sequence Number: 0
    Nwk Addr of Interest: 0x017d
    Extended Address: 00:00:00_00:00:0d:ee:b4 (00:00:00:00:00:0d:ee:b4)
  v Capability Information: 0x80
      .... ...0 = Alternate Coordinator: False
      .... ..0. = Full-Function Device: False
      .... .0.. = AC Power: False
      .... 0... = Rx On When Idle: False
      .0.. .... = Security Capability: False
      1... .... = Allocate Short Address: True
```

6.4.3 End Device Timeout

After joining or rejoining the network, the end device sends an `End Device Timeout Request` command to its parent with the desired timeout value. The parent updates the timeout in its neighbor table for the corresponding end device entry. The parent generates an `End Device Timeout Response` command with a status as `Success` and with the `Parent Information` field set to `MAC Data Poll Keepalive` method.

The end device sends the `End Device Timeout Request` command to inform its parent about the timeout requirements. This provides the parent with the ability to delete the child entry from the neighbor table if the child does not communicate with the parent in the specified amount of time. For more details, refer to the *Zigbee Specification Revision 22 1.0* (05-3474-22).

Figure 6-31. End Device Timeout Sequence



The following figure illustrates the complete scenario of the `End Device Timeout` setup. Packets `#24` and `#32` illustrate the `End Device Timeout Request` and `End Device Timeout Response` from the end device and coordinator devices, respectively.

Figure 6-32. End Device Timeout Sniffer Log

No.	Source	Destination	Time	Protocol	Leng	Info
6		Broadcast	36.992308	IEEE 802.15.4	10	Beacon Request
7	0x0000		36.993083	ZigBee	28	Beacon, Src: 0x0000, EPID: 00:00:00_00:00:0d:ee:b1
8	00:00:00:00:00:0d:ee:b4	0x0000	37.131442	IEEE 802.15.4	21	Association Request, RFD
9			37.132444	IEEE 802.15.4	5	Ack
10	00:00:00:00:00:0d:ee:b4	0x0000	37.631124	IEEE 802.15.4	18	Data Request
11			37.631124	IEEE 802.15.4	5	Ack
12	00:00:00:00:00:0d:ee:b1	00:00:00:00:00:0d:ee:b4	37.635136	IEEE 802.15.4	27	Association Response, PAN: 0x0733 Addr: 0x017d
13			37.635136	IEEE 802.15.4	5	Ack
14	0x017d	0x0000	37.636115	IEEE 802.15.4	12	Data Request
15			37.637112	IEEE 802.15.4	5	Ack
16	0x0000	0x017d	37.640103	ZigBee	73	Transport Key
17			37.641101	IEEE 802.15.4	5	Ack
18	0x017d	0x0000	37.642098	IEEE 802.15.4	12	Data Request
19			37.643096	IEEE 802.15.4	5	Ack
20	0x017d	Broadcast	37.684574	ZigBee ZDP	57	Device Announcement, Nwk Addr: 0x017d, Ext Addr: 00:00:00_00:00:0d:ee:b4
21			37.685572	IEEE 802.15.4	5	Ack
22	0x017d	0x0000	37.689054	IEEE 802.15.4	12	Data Request
23			37.690053	IEEE 802.15.4	5	Ack
24	0x017d	0x0000	37.723371	ZigBee	56	End Device Timeout Request
25			37.725366	IEEE 802.15.4	5	Ack
26	0x017d	0x0000	37.726364	ZigBee ZDP	48	Node Descriptor Request, Nwk Addr: 0x0000
27	0x017d	Broadcast	37.730383	ZigBee ZDP	57	Device Announcement, Nwk Addr: 0x017d, Ext Addr: 00:00:00_00:00:0d:ee:b4
28	0x017d	0x0000	37.736369	ZigBee ZDP	48	Node Descriptor Request, Nwk Addr: 0x0000
29			37.737367	IEEE 802.15.4	5	Ack
30	0x017d	0x0000	38.223383	IEEE 802.15.4	12	Data Request
31			38.224385	IEEE 802.15.4	5	Ack
32	0x0000	0x017d	38.226428	ZigBee	56	End Device Timeout Response, Success
33			38.228657	IEEE 802.15.4	5	Ack

The following figure illustrates the `End Device Timeout Response` with `Parent Information` field set to `MAC Data Poll Keepalive` and the `MAC Data Poll Keepalive` field set to `True`.

Figure 6-33. End Device Timeout Response

```
Frame 32: 56 bytes on wire (448 bits), 54 bytes captured (432 bits) on interface \\.\pipe\Atmel_Wireshark, id 0
IEEE 802.15.4 Data, Dst: 0x017d, Src: 0x0000
ZigBee Network Layer Command, Dst: 0x017d, Src: 0x0000
    Frame Control Field: 0x1a09, Frame Type: Command, Discover Route: Suppress, Security, Destination, Extended Source Command
    Destination: 0x017d
    <[Address: 0x017d]>
    Source: 0x0000
    <[Address: 0x0000]>
    Radius: 1
    Sequence Number: 159
    Destination: 00:00:00_00:00:0d:ee:b4 (00:00:00:00:00:0d:ee:b4)
    <[Extended Address: 00:00:00_00:00:0d:ee:b4 (00:00:00:00:00:0d:ee:b4)]>
    Extended Source: 00:00:00_00:00:0d:ee:b1 (00:00:00:00:00:0d:ee:b1)
    <[Extended Address: 00:00:00_00:00:0d:ee:b1 (00:00:00:00:00:0d:ee:b1)]>
    ZigBee Security Header
    Command Frame: End Device Timeout Response, Success
        Command Identifier: End Device Timeout Response (0x0c)
        Status: Success (0)
        Parent Information: 0x01, MAC Data Poll Keepalive
            .... ...1 = MAC Data Poll Keepalive: True
            .... ..0. = End Device Timeout Request Keepalive: False
            .... .0.. = Power Negotiation Supported: False
```

6.4.4 Service Discovery

The router/end device requests `Node Descriptor` during the initialization procedure before finding and binding to discover the `Capability Information` and other details of the `Coordinator` device in the network. For more details, refer to 6.1.3. Zigbee Device Profile (ZDP).

Packet `#28` and `#36` show the `Node Descriptor Request` and `Node Descriptor Response` from the router and coordinator devices, respectively. Packet `#40` shows the `APS: ACK` by end device for `Node Descriptor Response` from `Coordinator`.

Figure 6-34. Node Descriptor – End Device

No.	Source	Destination	Time	Protocol	Length	Info
28	0x017d	0x0000	37.736369	ZigBee ZDP	48	Node Descriptor Request, Nwk Addr: 0x0000
29			37.737367	IEEE 802.15.4	5	Ack
30	0x017d	0x0000	38.223383	IEEE 802.15.4	12	Data Request
31			38.224385	IEEE 802.15.4	5	Ack
32	0x0000	0x017d	38.226428	ZigBee	56	End Device Timeout Response, Success
33			38.228657	IEEE 802.15.4	5	Ack
34	0x017d	0x0000	38.228657	IEEE 802.15.4	12	Data Request
35			38.229655	IEEE 802.15.4	5	Ack
36	0x0000	0x017d	38.233083	ZigBee ZDP	62	Node Descriptor Response, Rev: 22, Nwk Addr: 0x0000, Status: Success
37			38.234151	IEEE 802.15.4	5	Ack
38	0x017d	0x0000	38.234151	IEEE 802.15.4	12	Data Request
39			38.234151	IEEE 802.15.4	5	Ack
40	0x017d	0x0000	38.274348	ZigBee	45	APS: Ack, Dst Endpt: 0, Src Endpt: 0
41			38.276347	IEEE 802.15.4	5	Ack

The following figure illustrates the `Node Descriptor Response` from a `Coordinator` device. Under `ZigBee Device Profile` field, the user can see the following:

- `Capability Information` of the coordinator node
- `Max Buffer Size`
- `Server Flags`
- `Descriptor Capability Field`

Figure 6-35. Node Descriptor Response – End Device

```
ZigBee Device Profile, Node Descriptor Response, Rev: 22, Nwk Addr: 0x0000, Status: Success
    Sequence Number: 1
    Status: Success (0)
    Nwk Addr of Interest: 0x0000
    Node Descriptor
        .... .... .... .000 = Type: 0 (Coordinator)
        .... .... .... 0... = Complex Descriptor: False
        .... .... ...1 .... = User Descriptor: True
        .... 0... .... .... = 868MHz BPSK Band: False
        ..0. .... .... .... = 902MHz BPSK Band: False
        .1.. .... .... .... = 2.4GHz OQPSK Band: True
        0... .... .... .... = EU Sub-GHz FSK Band: False
        Capability Information: 0x0f
            .... ...1 = Alternate Coordinator: True
            .... ..1. = Full-Function Device: True
            .... .1.. = AC Power: True
            .... 1... = Rx On When Idle: True
            .0.. .... = Security Capability: False
            0... .... = Allocate Short Address: False
        Manufacturer Code: 0x1014
        Max Buffer Size: 71
        Max Incoming Transfer Size: 43
        Server Flags: 0x2c40
        Max Outgoing Transfer Size: 43
        Descriptor Capability Field: 0x00
```

6.4.5 Finding and Binding

The user can configure the target endpoint/initiator endpoint as the following:

- Zigbee end device/multisensor – As the initiator endpoint
- Zigbee coordinator/combined interface – As the target endpoint

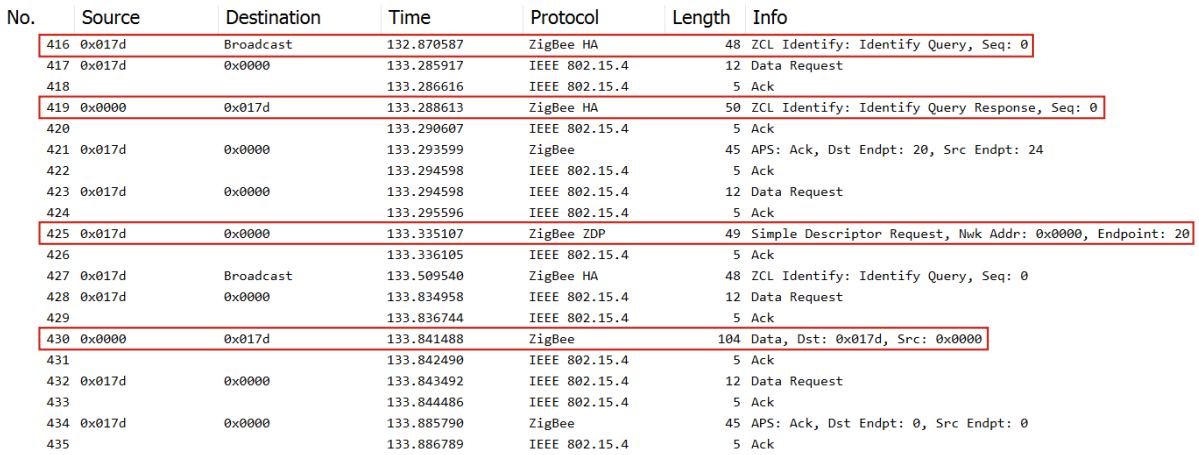
The following figure illustrates packets `#416` and `#419` as `Identify Query Request` and `Identify Query Response` from the end device and coordinator devices, respectively.

The end device as an initiator broadcasts `Identify Query` for identifying target endpoints. After receiving `Identify Query Response` from a target endpoint, the initiator unicasts the `Simple Descriptor Request` to the target device. The initiator endpoint, then, searches for any matching clusters between itself and the target endpoint and for each match found. It creates a corresponding entry in its binding table. If there is a request for group binding, the initiator endpoint configures group membership of the target endpoint.

After receiving `Identify Query Response`, that is, identifying the target endpoint, the target endpoint requests the `Simple Descriptor`.

Packets `#425` and `#430` are `Simple Descriptor Request` and `Simple Descriptor Response` from end device and coordinator devices, respectively.

Figure 6-36. Finding and Binding – End Device



No.	Source	Destination	Time	Protocol	Length	Info
416	0x017d	Broadcast	132.870587	ZigBee HA	48	ZCL Identify: Identify Query, Seq: 0
417	0x017d	0x0000	133.285917	IEEE 802.15.4	12	Data Request
418			133.286616	IEEE 802.15.4	5	Ack
419	0x0000	0x017d	133.288613	ZigBee HA	50	ZCL Identify: Identify Query Response, Seq: 0
420			133.290607	IEEE 802.15.4	5	Ack
421	0x017d	0x0000	133.293599	ZigBee	45	APS: Ack, Dst Endpt: 20, Src Endpt: 24
422			133.294598	IEEE 802.15.4	5	Ack
423	0x017d	0x0000	133.294598	IEEE 802.15.4	12	Data Request
424			133.295596	IEEE 802.15.4	5	Ack
425	0x017d	0x0000	133.335107	ZigBee ZDP	49	Simple Descriptor Request, Nwk Addr: 0x0000, Endpoint: 20
426			133.336105	IEEE 802.15.4	5	Ack
427	0x017d	Broadcast	133.509540	ZigBee HA	48	ZCL Identify: Identify Query, Seq: 0
428	0x017d	0x0000	133.834958	IEEE 802.15.4	12	Data Request
429			133.836744	IEEE 802.15.4	5	Ack
430	0x0000	0x017d	133.841488	ZigBee	104	Data, Dst: 0x017d, Src: 0x0000
431			133.842490	IEEE 802.15.4	5	Ack
432	0x017d	0x0000	133.843492	IEEE 802.15.4	12	Data Request
433			133.844486	IEEE 802.15.4	5	Ack
434	0x017d	0x0000	133.885790	ZigBee	45	APS: Ack, Dst Endpt: 0, Src Endpt: 0
435			133.886789	IEEE 802.15.4	5	Ack

6.4.6 Reporting

The following table provides details about the client/server clusters available for the multi-sensor device type in the Microchip Zigbee stack. For more details, refer to the *ZigBee Alliance Cluster Library Specification Revision 8* (075123). For more details regarding mandatory or optional clusters for a specific device type, refer to the *Matter Device Library Specification* (1.0).

Table 6-3. Supported Clusters – Multi-Sensor/Sensor Device Type

Device Type	Cluster ID	Server Clusters	Client Clusters	Attribute Identifier	Attribute Name
Multi sensor	0x0000	Basic	Basic	—	—
	0x0003	Identify	Identify	—	—
	0x0004	Groups	Groups	—	—
	0x0406	Occupancy sensing(1)	—	0x0000(1)	Occupancy(1)
	0x0400	Illuminance measurement(1)	—	0x0000(1)	Measured value(1)
	0x0402	Temperature measurement	—	—	—
	0x0405	Water content measurement	—	—	—
	0x0B05	Diagnostics	—	—	—

Note:

1. In this scenario, the end device/multi-sensor reports the occupancy (0x0000) attribute of the occupancy sensing (0x0406) cluster and the measured value (0x0000) attribute of the illuminance measurement (0x0400) cluster to the router/extended light device.

- Configure Reporting – Use the `Configure Reporting` command to configure the reporting mechanism for one or more of the attributes of a cluster.
  The following figure illustrates the packet `#436` that indicates `Configure Reporting Request` to coordinator by end device.

Figure 6-37. Configure Reporting – End Device

No.	Source	Destination	Time	Protocol	Leng	Info
436	0x017d	0x0000	133.891775	ZigBee HA	53	ZCL: Configure Reporting, Seq: 2
437			133.892784	IEEE 802.15.4	5	Ack
438	0x017d	0x0000	134.385302	IEEE 802.15.4	12	Data Request
439			134.385302	IEEE 802.15.4	5	Ack
440	0x0000	0x017d	134.388299	ZigBee HA	52	ZCL: Configure Reporting Response, Seq: 2
441			134.390292	IEEE 802.15.4	5	Ack
442	0x017d	0x0000	134.390292	IEEE 802.15.4	12	Data Request
443			134.390292	IEEE 802.15.4	5	Ack
444	0x017d	0x0000	134.425562	ZigBee	45	APS: Ack, Dst Endpt: 20, Src Endpt: 24
445			134.427559	IEEE 802.15.4	5	Ack

The `Direction` field specifies whether to report values of the attribute or whether to receive reports of the attribute.

The following figure illustrates the `Direction` field under `Reporting Configuration Record` in `ZCL`, which is set to `Received`, indicating that the coordinator device must receive the attribute values. It also indicates that the sender (end device) can configure its reporting mechanism to transmit/report the required/ desired attributes to the receiver (coordinator). Based on the current state of the sender's bindings, the sender sends reports to the receiver.

In the preceding scenario (see Figure 6-37), the user must configure the end device using the `Configure Reporting` command to report `Occupancy Sensing` and `Illumination Measurement` (of light sensor) attributes to the coordinator device.

The occupancy sensor is a measurement and sensing device that can measure and report the occupancy state within some area.

The light sensor is a measurement and sensing device that measures and reports the intensity of the emitting light by a light source.

Figure 6-37 illustrates the `Configure Reporting` for `Cluster: Illuminance Measurement`, where the end device reports the `Attribute: Measured Value` to the coordinator.

Figure 6-38. Configure Reporting – Illuminance Measurement

```
> Frame 436: 53 bytes on wire (424 bits), 51 bytes captured (408 bits) on interface \\.\pipe\Atmel_Wireshark, id 0
> IEEE 802.15.4 Data, Dst: 0x0000, Src: 0x017d
> ZigBee Network Layer Data, Dst: 0x0000, Src: 0x017d
v ZigBee Application Support Layer Data, Dst Endpt: 20, Src Endpt: 24
    > Frame Control Field: Data (0x00)
      Destination Endpoint: 20
      Cluster: Illuminance Measurement (0x0400)
      Profile: Home Automation (0x0104)
      Source Endpoint: 24
      Counter: 90
v ZigBee Cluster Library Frame, Command: Configure Reporting, Seq: 2
    > Frame Control Field: Profile-wide (0x18)
      Sequence Number: 2
      Command: Configure Reporting (0x06)
    v Reporting Configuration Record
          Direction: Received (0x01)
          Attribute: Measured Value (0x0000)
      Timeout: 90
```

The following figure illustrates the `Configure Reporting` for `Cluster: Occupancy Sensing`, where the end device reports the `Attribute: Occupancy` value to the coordinator.

Figure 6-39. Configure Reporting – Occupancy Sensing

```
> Frame 757: 53 bytes on wire (424 bits), 51 bytes captured (408 bits) on interface \\.\pipe\Atmel_Wireshark, id 0
> IEEE 802.15.4 Data, Dst: 0x0000, Src: 0x017d
> ZigBee Network Layer Data, Dst: 0x0000, Src: 0x017d
v ZigBee Application Support Layer Data, Dst Endpt: 20, Src Endpt: 19
    > Frame Control Field: Data (0x00)
      Destination Endpoint: 20
      Cluster: Occupancy Sensing (0x0406)
      Profile: Home Automation (0x0104)
      Source Endpoint: 19
      Counter: 93
v ZigBee Cluster Library Frame, Command: Configure Reporting, Seq: 5
    > Frame Control Field: Profile-wide (0x18)
      Sequence Number: 5
      Command: Configure Reporting (0x06)
    v Reporting Configuration Record
          Direction: Received (0x01)
          Attribute: Occupancy (0x0000)
      Timeout: 80
```

- Reporting Attributes – A device uses the `Report Attributes` command to report the values of one or more of its attributes to another device. Individual clusters define about reporting which attributes and at what interval.

Figure 6-40. Report Attributes – End Device

No.	Source	Destination	Time	Protocol	Leng	Info
1068	0x017d	0x0000	294.929010	ZigBee HA	52	ZCL: Report Attributes, Seq: 7
1069			294.930010	IEEE 802.15.4	5	Ack

The following figures illustrate details about the end device reporting the `Illuminance Measurement` cluster's `Measured Value` attribute and the `Occupancy Sensing` cluster's `Occupancy` attribute respectively to the coordinator.

Figure 6-41. Report Attributes – Illuminance Measurement

```
> Frame 4195: 53 bytes on wire (424 bits), 51 bytes captured (408 bits) on interface \\.\pipe\Atmel_Wireshark, id 0
> IEEE 802.15.4 Data, Dst: 0x0000, Src: 0x017d
> ZigBee Network Layer Data, Dst: 0x0000, Src: 0x017d
v ZigBee Application Support Layer Data, Dst Endpt: 20, Src Endpt: 24
    > Frame Control Field: Data (0x00)
      Destination Endpoint: 20
      Cluster: Illuminance Measurement (0x0400)
      Profile: Home Automation (0x0104)
      Source Endpoint: 24
      Counter: 120
v ZigBee Cluster Library Frame, Command: Report Attributes, Seq: 32
    > Frame Control Field: Profile-wide (0x18)
      Sequence Number: 32
      Command: Report Attributes (0x0a)
    v Attribute Field
          Attribute: Measured Value (0x0000)
          Data Type: 16-Bit Unsigned Integer (0x21)
          Measured Value: 255 (=0.060474 [lx])
```

Figure 6-42. Report Attributes – Occupancy Sensing

```
> Frame 4118: 52 bytes on wire (416 bits), 50 bytes captured (400 bits) on interface \\.\pipe\Atmel_Wireshark, id 0
> IEEE 802.15.4 Data, Dst: 0x0000, Src: 0x017d
> ZigBee Network Layer Data, Dst: 0x0000, Src: 0x017d
v ZigBee Application Support Layer Data, Dst Endpt: 20, Src Endpt: 19
    > Frame Control Field: Data (0x00)
      Destination Endpoint: 20
      Cluster: Occupancy Sensing (0x0406)
      Profile: Home Automation (0x0104)
      Source Endpoint: 19
      Counter: 119
v ZigBee Cluster Library Frame, Command: Report Attributes, Seq: 31
    > Frame Control Field: Profile-wide (0x18)
      Sequence Number: 31
      Command: Report Attributes (0x0a)
    v Attribute Field
        Attribute: Occupancy (0x0000)
        Data Type: 8-Bit Bitmap (0x18)
      v Occupancy: 0x00
            .... ...0 = Occupied: False
```

6.4.7 Security

For more details on the security key exchange in a centralized network between the trust center and end device, refer to 6.2.5. Security.

6.5 Touchlink Commissioning

The Zigbee protocol provides special commissioning called `Touchlink`, which is an easy-to-use proximity mechanism for commissioning a device to a network. The `Touchlink` commissioning cluster provides commands to support `Touchlink` commissioning. The `Touchlink` commissioning command set has command identifiers in the range 0x00-0x3f and is transmitted using the inter-PAN transmission service. This process works by the `Touchlink` initiator determining the proximity of the target device (to be commissioned) and negotiating/transferring network parameters. The `Touchlink` commissioning process can be used to form a new network or to join a node to an existing network. `Touchlink` is initiated on a node called the initiator. The ZCL provides the `Touchlink` as a cluster. The initiator must support the `Touchlink` cluster as a client, and the target node must support the cluster as a server. If it is required on a node, enable `Touchlink` commissioning via the Zigbee base device attribute `bdbCommissioningMode`. For more details on `Touchlink` commissioning, refer to the *ZigBee Alliance Cluster Library Specification Revision 8* (075123).

For example, a `ColorSceneController`, which is an end device type, brings the light into the network by requesting the light to form the distributed network via `Touchlink`. To enable `Touchlink` commissioning, bring a color scene controller close to a target (light) device around like 20-30 cms range.

Figure 6-43. Touchlink Commissioning

Color Scene Controller - ZigBee End Device (Initiator) →
Extended Lights- ZigBee Router (Target) →

No.	Source	Destination	Time	Protocol	Length	Info
4	00:00:00:00:00:0d:ee:b5	Broadcast	20.051326	ZigBee	35	ZCL Touchlink: Scan Request, Seq: 0
5	00:00:00:00:00:0d:ee:b3	00:00:00:00:00:0d:ee:b5	20.102189	ZigBee	71	ZCL Touchlink: Scan Response, Seq: 0
6			20.104184	IEEE 802.15.4	5	Ack
7	00:00:00:00:00:0d:ee:b5	Broadcast	20.301004	ZigBee	35	ZCL Touchlink: Scan Request, Seq: 0
8	00:00:00:00:00:0d:ee:b5	Broadcast	20.550030	ZigBee	35	ZCL Touchlink: Scan Request, Seq: 0
9	00:00:00:00:00:0d:ee:b5	Broadcast	20.799899	ZigBee	35	ZCL Touchlink: Scan Request, Seq: 0
10	00:00:00:00:00:0d:ee:b5	Broadcast	21.050908	ZigBee	35	ZCL Touchlink: Scan Request, Seq: 0
11	00:00:00:00:00:0d:ee:b5	00:00:00:00:00:0d:ee:b3	22.051030	ZigBee	41	ZCL Touchlink: Identify Request, Seq: 0
12			22.052980	IEEE 802.15.4	5	Ack
13	00:00:00:00:00:0d:ee:b5	00:00:00:00:00:0d:ee:b3	24.154246	ZigBee	91	ZCL Touchlink: Network Start Request, Seq: 0
14			24.156202	IEEE 802.15.4	5	Ack
15		Broadcast	24.156202	IEEE 802.15.4	10	Beacon Request
16	00:00:00:00:00:0d:ee:b3	00:00:00:00:00:0d:ee:b5	24.352929	ZigBee	52	ZCL Touchlink: Network Start Response, Seq: 0
17			24.353925	IEEE 802.15.4	5	Ack

7. Example Application Scenarios

7.1 Personal Area Network (PAN) Same Channel Co-Existence

It is possible to have multiple Zigbee networks on the same channel. The following figure illustrates that it is possible to start a second Personal Area Network (PAN) in the presence of an existing PAN.

The following figure illustrates packets `#19` and `#20` and provides details about the beacon frame transmission by coordinator 1 with extended PAN ID `0xdeeb1` and coordinator 2 with EPID `0xdeeb7` for router sending the `Beacon Request` as packet `#18` in the same channel.

Figure 7-1. PAN Channel Co-Existence

No.	Source	Destination	Time	Protocol	Leng	Info
18		Broadcast	81.052115	IEEE 802.15.4	10	Beacon Request
19	0x0000		81.053228	ZigBee	28	Beacon, Src: 0x0000, EPID: 00:00:00_00:00:0d:ee:b1
20	0x0000		81.054230	ZigBee	28	Beacon, Src: 0x0000, EPID: 00:00:00_00:00:0d:ee:b7

7.2 End-to-End Establishment of Application Link Key

In a secure network, when two devices need to communicate on a secure link with each other, the devices must request a link key from the trust center.

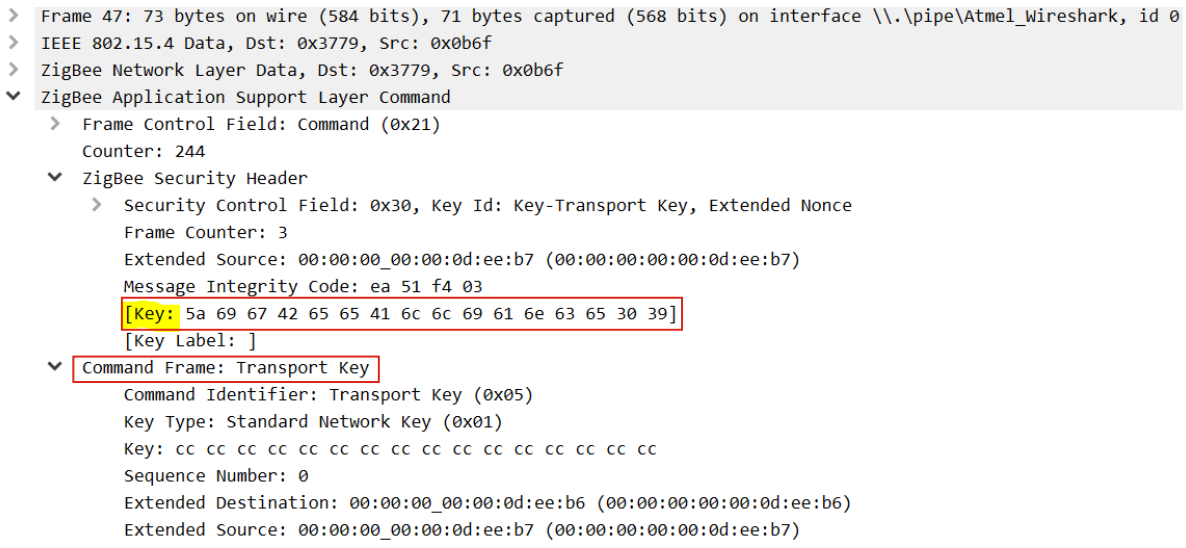
The following figure illustrates link key establishment between two routers. The router `0x0b6f` requests a link key from the trust center (see packet `#24`) to communicate with the router `0x3779`.

Figure 7-2. Link Key Establishment Between Two Routers

No.	Source	Destination	Time	Protocol	Leng	Info
24	0x0b6f	0x0000	19.053669	ZigBee	66	Request Key
25			19.055663	IEEE 802.15.4	5	Ack
26	0x0000	0x0b6f	19.061647	ZigBee	90	Transport Key
27			19.063673	IEEE 802.15.4	5	Ack
28	0x0b6f	0x0000	19.067631	ZigBee	65	Verify Key
29			19.069626	IEEE 802.15.4	5	Ack
30	0x0000	0x0b6f	19.071621	ZigBee	67	Confirm Key, SUCCESS
31			19.072618	IEEE 802.15.4	5	Ack
32	0x0b6f	Broadcast	19.076607	ZigBee ZDP	48	Permit Join Request
33	0x0b6f	Broadcast	19.115503	ZigBee ZDP	48	Permit Join Request
34		Broadcast	19.789421	IEEE 802.15.4	10	Beacon Request
35	0x0b6f		19.792413	ZigBee	28	Beacon, Src: 0x0b6f, EPID: 00:00:00_00:00:0d:ee:b7
36	0x0000		19.798862	ZigBee	28	Beacon, Src: 0x0000, EPID: 00:00:00_00:00:0d:ee:b7
37	00:00:00:00:00:0d:ee:b6	0x0b6f	19.930298	IEEE 802.15.4	21	Association Request, FFD
38			19.930874	IEEE 802.15.4	5	Ack
39	00:00:00:00:00:0d:ee:b6	0x0b6f	20.427924	IEEE 802.15.4	18	Data Request
40			20.428921	IEEE 802.15.4	5	Ack
41	00:00:00:00:00:0d:ee:b5	00:00:00:00:00:0d:ee…	20.431913	IEEE 802.15.4	27	Association Response, PAN: 0x3c3b Addr: 0x3779
42			20.432910	IEEE 802.15.4	5	Ack
43	0x0b6f	0x0000	20.435948	ZigBee	68	Update Device
44			20.436940	IEEE 802.15.4	5	Ack
45	0x0000	0x0b6f	20.441886	ZigBee	102	Data, Dst: 0x0b6f, Src: 0x0000
46			20.443880	IEEE 802.15.4	5	Ack
47	0x0b6f	0x3779	20.447870	ZigBee	73	Transport Key
48			20.448868	IEEE 802.15.4	5	Ack
49	0x3779	Broadcast	20.454894	ZigBee ZDP	57	Device Announcement, Nwk Addr: 0x3779, Ext Addr: 00:00:00_00:00:0d:ee:b6
50	0x3779	Broadcast	20.490799	ZigBee ZDP	57	Device Announcement, Nwk Addr: 0x3779, Ext Addr: 00:00:00_00:00:0d:ee:b6
51	0x3779	Broadcast	20.494745	ZigBee ZDP	57	Device Announcement, Nwk Addr: 0x3779, Ext Addr: 00:00:00_00:00:0d:ee:b6

The trust center uses the `Transport Key` command to send the link key to both the routers. The link key is applied in further data exchange between the routers. Packet `#47` uses this link key. The following figure illustrates the link key.

Figure 7-3. Link Key Establishment between Two Routers



8. Zigbee Green Power

The following are the Green Power (GP) infrastructure device types:

- Green Power Proxy (GPP) device or Proxy device
- Green Power Sink (GPS) device or Sink device
- Green Power Combo (GPC) device or Combo device

The Green Power Device (GPD) can get commissioned directly with the sink device if the device is in the vicinity. The following are the two ways of commissioning:

- Unidirectional Commissioning
- Bidirectional Commissioning

8.1 Unidirectional Commissioning

- For unidirectional commissioning, the sink device is put in Commissioning mode, and GPD sends a `Commissioning` command with `RxAfterTx-0` (see packet `#74`) and all device details, including the device type, security level, security key type, security key and more (see the following figure).
- The sink device verifies the device details, security key and accepts the commissioning. The sink device makes a new entry for this device in its sink table, broadcasts the GP pairing command (see packets `#76` and `#78`), and the device announces in a broadcast the commissioning of this new GPD (see packets `#77` and `#79`) (see the following figure). For more details, refer to the *Zigbee PRO Green Power feature specification Basic functionality set* (Version 1.1.1).

Figure 8-1. Green Power Commissioning – Unidirectional Commissioning

No.	Source	Destination	Time	Protocol	Length	Info
73	0x5ffd	Broadcast	55.120520	IEEE 802.15.4	48	Data, Dst: Broadcast, Src: 0x5ffd
74		Broadcast	64.787024	IEEE 802.15.4	40	Data, Dst: Broadcast
75	0xbeef	Broadcast	64.792848	IEEE 802.15.4	55	Data, Dst: Broadcast, Src: 0xbeef
76	0xbeef	Broadcast	64.796144	IEEE 802.15.4	76	Data, Dst: Broadcast, Src: 0xbeef
77	0x5ffd	Broadcast	64.832152	IEEE 802.15.4	55	Data, Dst: Broadcast, Src: 0x5ffd
78	0x5ffd	Broadcast	64.837176	IEEE 802.15.4	76	Data, Dst: Broadcast, Src: 0x5ffd
79	0xbeef	Broadcast	65.431176	IEEE 802.15.4	55	Data, Dst: Broadcast, Src: 0xbeef
80	0xbeef	Broadcast	69.629432	IEEE 802.15.4	48	Data, Dst: Broadcast, Src: 0xbeef
81	0x5ffd	0xbeef	69.800312	IEEE 802.15.4	48	Data, Dst: 0xbeef, Src: 0x5ffd
82			69.802312	IEEE 802.15.4	3	Ack
83	0xbeef	0x5ffd	69.804584	IEEE 802.15.4	81	Data, Dst: 0x5ffd, Src: 0xbeef
84			69.807640	IEEE 802.15.4	3	Ack
85	0x5ffd	0xbeef	69.809208	IEEE 802.15.4	43	Data, Dst: 0xbeef, Src: 0x5ffd
86			69.811048	IEEE 802.15.4	3	Ack
87	0x5ffd	Broadcast	70.281856	IEEE 802.15.4	48	Data, Dst: Broadcast, Src: 0x5ffd

8.2 Bidirectional Commissioning

- For bidirectional commissioning, the sink device is put in `Commissioning` mode, and GPD sends a `Commissioning` command with `RxAfterTx-1` (see packet `#145`) and all device details including the device type, security level, security key type, security key and more.
- The sink device verifies the details and responds with `Commissioning Reply` (see packets `#146` and `#147`). `Commissioning Reply` can include a new security key and PAN ID if the same are requested in the commissioning packet.
- When GPD receives and processes this `Commissioning Reply`, it sends a `Success` command with a new key and PAN ID (see packet `#150`). On successfully decrypting packet `#150` (`Success`) from GPD, the sink device adds a new entry in its sink table, broadcasts GP pairing and device announce for this device. For more details, refer to the *Zigbee PRO Green Power feature specification Basic functionality set* (Version 1.1.1).

Figure 8-2. Bidirectional Commissioning

No.	Source	Destination	Time	Protocol	Length	Info
133	0xbeef	Broadcast	106.345496	ZigBee	48	Link Status
134		Broadcast	108.996288	ZigBee Green Power	10	Channel Request
135	0xbeef	Broadcast	108.998000	ZigBee GP	57	ZCL Green Power: GP Response, Seq: 2
136	0xbeef	Broadcast	109.078672	ZigBee GP	57	ZCL Green Power: GP Response, Seq: 2
137		Broadcast	109.995840	ZigBee Green Power	10	Channel Request
138		Broadcast	110.016624	ZigBee Green Power	10	Channel Configuration
139	0xbeef	Broadcast	110.020040	ZigBee GP	57	ZCL Green Power: GP Response, Seq: 3
140	0xbeef	Broadcast	110.060984	ZigBee GP	57	ZCL Green Power: GP Response, Seq: 3
141	0x12345678	Broadcast	110.542648	ZigBee Green Power	41	Commissioning
142	0xbeef	Broadcast	110.546208	ZigBee GP	81	ZCL Green Power: GP Response, Seq: 4
143	0xbeef	Broadcast	110.629832	ZigBee GP	81	ZCL Green Power: GP Response, Seq: 4
144	0xbeef	Broadcast	110.654704	ZigBee GP	57	ZCL Green Power: GP Response, Seq: 3
145	0x12345678	Broadcast	111.536240	ZigBee Green Power	41	Commissioning
146	0x12345678	Broadcast	111.557800	ZigBee Green Power	39	Commissioning Reply
147	0x12345678	Broadcast	111.559592	ZigBee Green Power	39	Commissioning Reply
148	0xbeef	Broadcast	111.563320	ZigBee GP	81	ZCL Green Power: GP Response, Seq: 5
149	0xbeef	Broadcast	111.599312	ZigBee GP	81	ZCL Green Power: GP Response, Seq: 5
150	0x12345678	Broadcast	111.657168	ZigBee Green Power	22	Success
151	0x5678	Broadcast	111.663944	ZigBee ZDP	55	Device Announcement, Nwk Addr: 0x5678, Ext Addr: ff:ff:ff:ff:ff:ff:ff:ff
152	0xbeef	Broadcast	111.667568	ZigBee GP	76	ZCL Green Power: GP Pairing, Seq: 6
153	0x5678	Broadcast	111.739280	ZigBee ZDP	55	Device Announcement, Nwk Addr: 0x5678, Ext Addr: ff:ff:ff:ff:ff:ff:ff:ff
154	0xbeef	Broadcast	111.743656	ZigBee GP	76	ZCL Green Power: GP Pairing, Seq: 6
155	0x12345678	Broadcast	111.756856	ZigBee Green Power	22	Success
156	0x5678	Broadcast	112.306512	ZigBee ZDP	55	Device Announcement, Nwk Addr: 0x5678, Ext Addr: ff:ff:ff:ff:ff:ff:ff:ff
157	0x289c	0xbeef	116.776896	ZigBee GP	48	ZCL: Read Attributes, Seq: 20
158			116.778896	IEEE 802.15.4	3	Ack
159	0xbeef	0x289c	116.781160	ZigBee GP	81	ZCL: Read Attributes Response, Seq: 20
160			116.784216	IEEE 802.15.4	3	Ack
161	0x289c	0xbeef	116.787384	ZigBee	43	APS: Ack, Dst Endpt: 242, Src Endpt: 242
162			116.789224	IEEE 802.15.4	3	Ack

8.3 Basic Commissioning (Channel Configuration)

- If the operational channel of the sink device is not known to the device, GPD gets the same by performing the channel configuration procedure. To get the operational channel, GPD sends the channel request to the sink device (see packets `#134` and `#137`), and it responds with the operational channel to GPD by channel configuration command. The following figure illustrates packets `#134`, `#137` and `#138`.
- At first, the GPD does not know about the operational channel and sends the channel requests in multiple channels, which are enabled in channel mask.
- GPD indicates its availability for reception in Frame Control Field (FCF) of the `Channel Request` command. If the `Auto Commissioning` of the FCF field is set to '`0`', `RxAfterTx` is enabled and vice versa.
- After sending a channel request from the same device with `Auto Commissioning = 0`, the GPD receives the channel configuration.
- Send packet `#134` channel request packet with `Auto Commissioning = 1` and `RxAfterTx` is disabled, hence the GPD does not receive the channel configuration. Whereas, for the channel request packet `#137`, the `Auto Commissioning = 0` and `RxAfterTx` is enabled, the GPD receives the channel configuration (see packet `#138`).
- GPD after receiving this command changes its operational channel to the sink device's operational channel.
- When the operational channel of the sink device is not same as GPD's RX channels, the sink device changes its channel for a short duration (5s) to deliver the channel configuration packet in GPD's RX channel. When the operational channel and RX channels are the same, the sink device need not change its channel. After receiving the channel configuration, the GPD need not send any more channel requests and can continue with commissioning. For more details, refer to the *Zigbee PRO Green Power feature specification Basic functionality set* (Version 1.1.1).

Figure 8-3. Basic Commissioning (Channel Configuration)

No.	Source	Destination	Time	Protocol	Length	Info
133	0xbeef	Broadcast	106.345496	ZigBee	48	Link Status
134		Broadcast	108.996288	ZigBee Green Power	10	Channel Request
135	0xbeef	Broadcast	108.998000	ZigBee GP	57	ZCL Green Power: GP Response, Seq: 2
136	0xbeef	Broadcast	109.078672	ZigBee GP	57	ZCL Green Power: GP Response, Seq: 2
137		Broadcast	109.995840	ZigBee Green Power	10	Channel Request
138		Broadcast	110.016624	ZigBee Green Power	10	Channel Configuration
139	0xbeef	Broadcast	110.020040	ZigBee GP	57	ZCL Green Power: GP Response, Seq: 3
140	0xbeef	Broadcast	110.060984	ZigBee GP	57	ZCL Green Power: GP Response, Seq: 3
141	0x12345678	Broadcast	110.542648	ZigBee Green Power	41	Commissioning
142	0xbeef	Broadcast	110.546208	ZigBee GP	81	ZCL Green Power: GP Response, Seq: 4

8.4 Data Transmission

GPD transmits the data packet to the sink device via proxy, if the proxy device is present in the network. In such cases, the proxy device sends the GP notification on behalf of the GPD device. The following figure illustrates the data transmission from the GPD via proxy.

Figure 8-4. Data Transmission from GPD via Proxy

No.	Source	Destination	Time	Protocol	Length	Info
1	0xbeef	Broadcast	0.000000	ZigBee	51	Link Status
2	0x55d1	Broadcast	2.423008	ZigBee GP	76	ZCL Green Power: GP Pairing, Seq: 1
3	0x55d1	Broadcast	2.499440	ZigBee GP	76	ZCL Green Power: GP Pairing, Seq: 1
4	0x55d1	Broadcast	2.502912	ZigBee GP	76	ZCL Green Power: GP Pairing, Seq: 1
5	0x55d1	Broadcast	3.142104	ZigBee GP	76	ZCL Green Power: GP Pairing, Seq: 1
6	0x70da	Broadcast	11.412616	ZigBee	51	Link Status
7	0x12345678	Broadcast	12.436600	ZigBee Green Power	22	Data, GPD Src ID: 0x12345678
8	0x5678	Broadcast	12.442520	ZigBee GP	62	ZCL Green Power: GP Notification, Seq: 1
9	0x5678	Broadcast	12.480344	ZigBee GP	62	ZCL Green Power: GP Notification, Seq: 1
10	0x5678	Broadcast	12.541600	ZigBee GP	62	ZCL Green Power: GP Notification, Seq: 2
11	0x55d1	Broadcast	12.920024	ZigBee	51	Link Status
12	0x5678	Broadcast	13.081848	ZigBee GP	62	ZCL Green Power: GP Notification, Seq: 1
13	0x5678	Broadcast	13.182536	ZigBee GP	62	ZCL Green Power: GP Notification, Seq: 2

If the proxy device is not present in the network, the GPD sends the data to the sink device. The following figure illustrates the GPD data TX without proxy.

Figure 8-5. GPD Data TX Without Proxy

No.	Source	Destination	Time	Protocol	Length	Info
1	0x12345678	Broadcast	0.000000	ZigBee Green Power	14	Toggle

9. Document Revision History

Table 9-1. Document Revision History

Revision	Date	Section	Description
A	11/2022	Document	Initial Revision

Microchip Information

The Microchip Website

Microchip provides online support via our website at www.microchip.com/. This website is used to make files and information easily available to customers. Some of the content available includes:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQs), technical support requests, online discussion groups, Microchip design partner program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

Product Change Notification Service

Microchip's product change notification service helps keep customers current on Microchip products. Subscribers will receive email notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, go to www.microchip.com/pcn and follow the registration instructions.

Customer Support

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Embedded Solutions Engineer (ESE)
- Technical Support

Customers should contact their distributor, representative or ESE for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in this document.

Technical support is available through the website at: www.microchip.com/support

Microchip Devices Code Protection Feature

Note the following details of the code protection feature on Microchip products:

- Microchip products meet the specifications contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is secure when used in the intended manner, within operating specifications, and under normal conditions.
- Microchip values and aggressively protects its intellectual property rights. Attempts to breach the code protection features of Microchip product is strictly prohibited and may violate the Digital Millennium Copyright Act.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of its code. Code protection does not mean that we are guaranteeing the product is "unbreakable". Code protection is constantly evolving. Microchip is committed to continuously improving the code protection features of our products.

Legal Notice

This publication and the information herein may be used only with Microchip products, including to design, test, and integrate Microchip products with your application. Use of this information in any other manner violates these terms. Information regarding device applications is provided only for your convenience and may be superseded

by updates. It is your responsibility to ensure that your application meets with your specifications. Contact your local Microchip sales office for additional support or, obtain additional support at www.microchip.com/en-us/support/design-help/client-support-services.

THIS INFORMATION IS PROVIDED BY MICROCHIP "AS IS". MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE, OR WARRANTIES RELATED TO ITS CONDITION, QUALITY, OR PERFORMANCE.

IN NO EVENT WILL MICROCHIP BE LIABLE FOR ANY INDIRECT, SPECIAL, PUNITIVE, INCIDENTAL, OR CONSEQUENTIAL LOSS, DAMAGE, COST, OR EXPENSE OF ANY KIND WHATSOEVER RELATED TO THE INFORMATION OR ITS USE, HOWEVER CAUSED, EVEN IF MICROCHIP HAS BEEN ADVISED OF THE POSSIBILITY OR THE DAMAGES ARE FORESEEABLE. TO THE FULLEST EXTENT ALLOWED BY LAW, MICROCHIP'S TOTAL LIABILITY ON ALL CLAIMS IN ANY WAY RELATED TO THE INFORMATION OR ITS USE WILL NOT EXCEED THE AMOUNT OF FEES, IF ANY, THAT YOU HAVE PAID DIRECTLY TO MICROCHIP FOR THE INFORMATION.

Use of Microchip devices in life support and/or safety applications is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights unless otherwise stated.

Trademarks

The Microchip name and logo, the Microchip logo, Adaptec, AVR, AVR logo, AVR Freaks, BesTime, BitCloud, CryptoMemory, CryptoRF, dsPIC, flexPWR, HELDO, IGLOO, JukeBlox, KeeLoq, Kleer, LANCheck, LinkMD, maXStylus, maXTouch, MediaLB, megaAVR, Microsemi, Microsemi logo, MOST, MOST logo, MPLAB, OptoLyzer, PIC, picoPower, PICSTART, PIC32 logo, PolarFire, Prochip Designer, QTouch, SAM-BA, SenGenuity, SpyNIC, SST, SST Logo, SuperFlash, Symmetricom, SyncServer, Tachyon, TimeSource, tinyAVR, UNI/O, Vectron, and XMEGA are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

AgileSwitch, APT, ClockWorks, The Embedded Control Solutions Company, EtherSynch, Flashtec, Hyper Speed Control, HyperLight Load, Libero, motorBench, mTouch, Powermite 3, Precision Edge, ProASIC, ProASIC Plus, ProASIC Plus logo, Quiet- Wire, SmartFusion, SyncWorld, Temux, TimeCesium, TimeHub, TimePictra, TimeProvider, TrueTime, and ZL are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Adjacent Key Suppression, AKS, Analog-for-the-Digital Age, Any Capacitor, AnyIn, AnyOut, Augmented Switching, BlueSky, BodyCom, Clockstudio, CodeGuard, CryptoAuthentication, CryptoAutomotive, CryptoCompanion, CryptoController, dsPICDEM, dsPICDEM.net, Dynamic Average Matching, DAM, ECAN, Espresso T1S, EtherGREEN, GridTime, IdealBridge, In-Circuit Serial Programming, ICSP, INICnet, Intelligent Paralleling, IntelliMOS, Inter-Chip Connectivity, JitterBlocker, Knob-on-Display, KoD, maxCrypto, maxView, memBrain, Mindi, MiWi, MPASM, MPF, MPLAB Certified logo, MPLIB, MPLINK, MultiTRAK, NetDetach, Omniscient Code Generation, PICDEM, PICDEM.net, PICkit, PICtail, PowerSmart, PureSilicon, QMatrix, REAL ICE, Ripple Blocker, RTAX, RTG4, SAM-ICE, Serial Quad I/O, simpleMAP, SimpliPHY, SmartBuffer, SmartHLS, SMART-I.S., storClad, SQI, SuperSwitcher, SuperSwitcher II, Switchtec, SynchroPHY, Total Endurance, Trusted Time, TSHARC, USBCheck, VariSense, VectorBlox, VeriPHY, ViewSpan, WiperLock, XpressConnect, and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

The Adaptec logo, Frequency on Demand, Silicon Storage Technology, and Symmcom are registered trademarks of Microchip Technology Inc. in other countries.

GestIC is a registered trademark of Microchip Technology Germany II GmbH & Co. KG, a subsidiary of Microchip Technology Inc., in other countries.

All other trademarks mentioned herein are property of their respective companies.

ISBN: 978-1-6683-1511-8

Quality Management System

For information regarding Microchip's Quality Management Systems, please visit www.microchip.com/quality.

Worldwide Sales and Service

AMERICAS	ASIA/PACIFIC	ASIA/PACIFIC	EUROPE
Corporate Office 2355 West Chandler Blvd. Chandler, AZ 85224-6199 Tel: 480-792-7200 Fax: 480-792-7277 Technical Support: www.microchip.com/support Web Address: www.microchip.com **Atlanta** Duluth, GA Tel: 678-957-9614 Fax: 678-957-1455 **Austin, TX** Tel: 512-257-3370 **Boston** Westborough, MA Tel: 774-760-0087 Fax: 774-760-0088 **Chicago** Itasca, IL Tel: 630-285-0071 Fax: 630-285-0075 **Dallas** Addison, TX Tel: 972-818-7423 Fax: 972-818-2924 **Detroit** Novi, MI Tel: 248-848-4000 **Houston, TX** Tel: 281-894-5983 **Indianapolis** Noblesville, IN Tel: 317-773-8323 Fax: 317-773-5453 Tel: 317-536-2380 **Los Angeles** Mission Viejo, CA Tel: 949-462-9523 Fax: 949-462-9608 Tel: 951-273-7800 **Raleigh, NC** Tel: 919-844-7510 **New York, NY** Tel: 631-435-6000 **San Jose, CA** Tel: 408-735-9110 Tel: 408-436-4270 **Canada - Toronto** Tel: 905-695-1980 Fax: 905-695-2078	**Australia - Sydney** Tel: 61-2-9868-6733 **China - Beijing** Tel: 86-10-8569-7000 **China - Chengdu** Tel: 86-28-8665-5511 **China - Chongqing** Tel: 86-23-8980-9588 **China - Dongguan** Tel: 86-769-8702-9880 **China - Guangzhou** Tel: 86-20-8755-8029 **China - Hangzhou** Tel: 86-571-8792-8115 **China - Hong Kong SAR** Tel: 852-2943-5100 **China - Nanjing** Tel: 86-25-8473-2460 **China - Qingdao** Tel: 86-532-8502-7355 **China - Shanghai** Tel: 86-21-3326-8000 **China - Shenyang** Tel: 86-24-2334-2829 **China - Shenzhen** Tel: 86-755-8864-2200 **China - Suzhou** Tel: 86-186-6233-1526 **China - Wuhan** Tel: 86-27-5980-5300 **China - Xian** Tel: 86-29-8833-7252 **China - Xiamen** Tel: 86-592-2388138 **China - Zhuhai** Tel: 86-756-3210040	**India - Bangalore** Tel: 91-80-3090-4444 **India - New Delhi** Tel: 91-11-4160-8631 **India - Pune** Tel: 91-20-4121-0141 **Japan - Osaka** Tel: 81-6-6152-7160 **Japan - Tokyo** Tel: 81-3-6880- 3770 **Korea - Daegu** Tel: 82-53-744-4301 **Korea - Seoul** Tel: 82-2-554-7200 **Malaysia - Kuala Lumpur** Tel: 60-3-7651-7906 **Malaysia - Penang** Tel: 60-4-227-8870 **Philippines - Manila** Tel: 63-2-634-9065 **Singapore** Tel: 65-6334-8870 **Taiwan - Hsin Chu** Tel: 886-3-577-8366 **Taiwan - Kaohsiung** Tel: 886-7-213-7830 **Taiwan - Taipei** Tel: 886-2-2508-8600 **Thailand - Bangkok** Tel: 66-2-694-1351 **Vietnam - Ho Chi Minh** Tel: 84-28-5448-2100	**Austria - Wels** Tel: 43-7242-2244-39 Fax: 43-7242-2244-393 **Denmark - Copenhagen** Tel: 45-4485-5910 Fax: 45-4485-2829 **Finland - Espoo** Tel: 358-9-4520-820 **France - Paris** Tel: 33-1-69-53-63-20 Fax: 33-1-69-30-90-79 **Germany - Garching** Tel: 49-8931-9700 **Germany - Haan** Tel: 49-2129-3766400 **Germany - Heilbronn** Tel: 49-7131-72400 **Germany - Karlsruhe** Tel: 49-721-625370 **Germany - Munich** Tel: 49-89-627-144-0 Fax: 49-89-627-144-44 **Germany - Rosenheim** Tel: 49-8031-354-560 **Israel - Ra'anana** Tel: 972-9-744-7705 **Italy - Milan** Tel: 39-0331-742611 Fax: 39-0331-466781 **Italy - Padova** Tel: 39-049-7625286 **Netherlands - Drunen** Tel: 31-416-690399 Fax: 31-416-690340 **Norway - Trondheim** Tel: 47-72884388 **Poland - Warsaw** Tel: 48-22-3325737 **Romania - Bucharest** Tel: 40-21-407-87-50 **Spain - Madrid** Tel: 34-91-708-08-90 Fax: 34-91-708-08-91 **Sweden - Gothenberg** Tel: 46-31-704-60-40 **Sweden - Stockholm** Tel: 46-8-5090-4654 **UK - Wokingham** Tel: 44-118-921-5800 Fax: 44-118-921-5820