AN1364: Using the Wi-SUN Network Measurement Application

This version of AN1364 has been deprecated. For the latest version, see docs.silabs.com.

****

The Wi-SUN Network Measurement Application provides a simple tool for testing Silicon Labs Wi-SUN FAN (Wireless Smart Ubiquitous Network, Field Area Network) stack performance. Using the IPv6 ping capability, the application can trigger round-trip latency tests within a Wi-SUN mesh network. It provides information about the packet error rate, Routing Protocol for Low-Power and Lossy Networks (RPL) routing metrics, average ping latencies with different neighbors, and more. To do so, the application exposes two interfaces: a Command Line Interface (CLI) and the WSTK mainboard LCD screen.

The Wi-SUN FAN stack is supported on the EFR32FG12 and EFR32MG12 devices. The radio boards used must support one of the sub-GHz bands.

KEY POINTS

- Presents the Wi-SUN Network Measurement application capabilities.
- Introduces the LCD screen interface.
- Introduces the CLI interface.
- Provides guidelines to improve the ping latency in a Wi-SUN network.

1 Introduction

Silicon Labs recommends that you review *QSG181: Silicon Labs Wi-SUN SDK QuickStart Guide* to familiarize yourself with the Silicon Labs Wi-SUN stack and Simplicity Studio 5 before reading this document.

The following features are provided as part of the Wi-SUN Network Measurement application:

- Transmits pings and receives pong replies.
- Computes ping latencies.
- Configures the number of pings to be sent during a test.
- Configures the ping payload size.
- Configures the devices to target with the ping packets.
- Provides the minimum, maximum, and average ping latencies on a given test.
- Provides the ping error rate, the MAC-level packet error rate, and RPL metrics.
- Enables access to the previous features through an LCD screen or a CLI. In addition to these features the Wi-SUN Network Measurement application behaves like a standard Wi-SUN router once connected to a Wi-SUN network. During the connection steps, it automatically selects a suitable parent to connect to the network. When connected, it can be selected by another Wi-SUN device to be its parent.

The Wi-SUN Network Measurement application requires you to have at least two WSTK mainboards and two preferably from the following list of matching radio boards:

- brd4163a
- brd4164a
- brd4170a
- brd4172a
- brd4173a
- brd4253a
- brd4254a

2 Theory of Operation

This chapter discusses the operation of the Wi-SUN Network Measurement application independently from the interface. You can execute these either through the CLI or the LCD screen interface. Specific interface information is shared in the following sections.

For the first step you need to start a Wi-SUN network using one of these solutions:

- The Wi-SUN SoC Border Router demo available in the Gecko SDK through Simplicity Studio 5 (for more information, see *QSG181: Silicon Labs Wi-SUN SDK QuickStart Guide*).
- The Wi-SUN Linux Border Router solution (for more information, see *AN1332: Silicon Labs Wi-SUN Network Setup and Configuration*).

Follow the steps to get your border router up and running and move to the next chapter.

2.1 Flash the Application

In the Simplicity Studio 5 Launcher perspective, select the EFR32 evaluation kit you want to use to run the Wi-SUN Network Measurement application. In the **EXAMPLE PROJECTS & DEMOS** panel, find the **Wi-SUN - SoC Network Measurement** project and click **CREATE** to add it to your workspace.

Finally, build the wisun_soc_network_measurement project and flash it on the selected EFR32. Refer to *QSG181: Silicon Labs Wi-SUN SDK QuickStart Guide* for the detailed steps.

2.2 Connect to a Wi-SUN Network

By default, the Wi-SUN Network Measurement application tries to connect to the **Wi-SUN Network**. If a network with this name and a matching Wi-SUN PHY configuration is available nearby, the **Performance Measurement** node connects to it using the best parent node available (selected by the RPL routing protocol).

There are two solutions to change the default network name:

- Using Simplicity Studio 5 prior to building the project: open the wisun_soc_network_measurement.slcp file and go to the **SOFTWARE COMPONENTS** panel. After you select the **Wi-SUN/Application/Setting** component, you can configure the default connection settings. Under **Wi-SUN network configuration**, modify the **Network Name** parameter to match the name of your Wi-SUN network.
- Use the CLI available in the application. (For more information, see chapter 4 Command Line Interface.)

Each EFR32 radio board reference number defaults to a specific Wi-SUN PHY configuration matching its radio capabilities. To change the Wi-SUN PHY used by the project, refer to chapter 4 of *UG495: Silicon Labs Wi-SUN Developer's Guide*.

2.3 Start a Ping Test

You can trigger a ping test using either interface of the Wi-SUN Network Measurement application (LCD screen or CLI). The test consists of emitting ping packets on the Wi-SUN network to an IPv6 target (other Wi-SUN nodes or external IPv6 devices). This device replies with pong packets for each ping received. When the pong is received by the **Performance Measurement** node, the Wi-SUN Network Measurement application computes a round-trip latency on the communication link. During the test, the Wi-SUN Network Measurement application monitors several metrics to provide insights into the Wi-SUN stack behavior.

2.4 Retrieve Test Results

After a ping test has ended, test results are available. They sum up the following information:

- Ping packet loss percentage
- Minimum ping latency in milliseconds
- Maximum ping latency in milliseconds
- Average ping latency in milliseconds

In addition to the ping test results, the Wi-SUN Network Measurement application also monitors metrics associated with the Wi-SUN stack.

2.4.1 Impact of Wi-SUN signaling on ping test results

Due to the characteristics of a Wi-SUN network, where network management such as PAN advertising can take a certain time (it requires sending a PAN Advertisement on each channel, and there are often as much as 129 channels), using a ping to measure network latency needs to be taken cautiously.

Over several pings there will be mostly short ping durations, and some longer durations when network signaling takes precedence over data transmission. When network signaling occurs, the ping duration will be impacted.

The 'max' and 'average' ping results will be impacted by network management. To get a valid 'latency' value to compare between PHYs configurations, we'll only keep the 'min' ping duration of at least 100 pings in tables and graphs below. The ‘Wi-SUN SoC Network Measurement’ application is designed to allow this test.

Note: The `network_size` parameter will have a significant impact on these 'signaling perturbations'. The smaller network sizes use more frequent network signaling and therefore have the highest impact.

2.5 Wi-SUN Metrics

To provide a better understanding on the Wi-SUN stack behavior the stack maintains internal statistics. The Wi-SUN Network Measurement application monitors several metrics during the tests:

- Lifetime
- MAC transmission count
- MAC transmission failed
- RPL rank
- Expected Transmission Count (ETX)
- Received Signal Level OUT (RSL out)
- Received Signal Level IN (RSL in)

The metrics are maintained for each neighbor device in the routed network (that is, the parent and all of the children in the RPL mesh tree). Each metric is described in the following sections.

2.5.1 Lifetime

The lifetime is a counter maintained with each neighbor to evaluate the state of the connection with this device. After each successful unicast communication with the neighbor, the counter is reset to 2200 seconds. When no unicast communication is exchanged with the neighbor, the counter is decreased each second.

2.5.2 MAC Transmission Count

The MAC transmission count (or MAC TX count) represents the number of packets transmitted over the air. A single ping attempt can result in several MAC-level transmission if the first attempts have not been successful. The number also considers packets not related to the ping test itself. Packets routed by the device to its parent and network maintenance related communications also increase this number.

2.5.3 MAC Transmission Failed

The MAC transmit failed (or MAC TX failed) represents the number of unsuccessful transmission attempts. In the Wi-SUN FAN specification, the peer must acknowledge every unicast packet. If a packet is not acknowledged within a given timeout, the transmission is considered failed. The packet is retransmitted by the Wi-SUN stack MAC layer until it reaches the maximum number of transmission attempts.

2.5.4 RPL Rank

The RPL rank represents the rank or hop distance from the tree root (that is, the Wi-SUN border router). For more information, see section 3.5 of RFC 6550.

2.5.5 Expected Transmission Count

The Expected Transmission Count (ETX) is an Exponentially Weighted Moving Average (EWMA) of the number of expected packet transmissions required for error-free reception at destination. The ETX is calculated as (frame transmission attempts)/(received frame acknowledgements) *128 with a maximum value of 1024, where 0 received frame acknowledgments sets ETX to the maximum value. This metric is maintained as part of the RPL protocol (RFC 6550) and participates in the best parent selection.

2.5.6 Received Signal Level OUT

The Received Signal Level out (RSL out) is an EWMA of the received signal level for the node-to-neighbor direction. The RSL is calculated as the received signal level relative to standard thermal noise (290°K) at 1 Hz bandwidth or -174 dBm. This provides a range of -174 (0) to +80 (254) dBm.

2.5.7 Received Signal Level IN

The Received Signal Level in (RSL_in) is an EWMA of the received signal level for the neighbor-to-node direction. The RSL is calculated as the received signal level relative to standard thermal noise (290°K) at 1 Hz bandwidth (= -174 dBm). This provides a range of -174 (0) to +80 (254) dBm in 1 dBm steps. The RSL_in value is communicated through an information element in the Wi-SUN packet exchanges between the device and its neighbor.

2.6 Test Configuration

The Wi-SUN Network Measurement application provides several settings to configure the ping test. The capabilities vary depending on the interface used but some of the configurations are shared:

- **Packet count**: Defines the number of ping packets sent per target during the tests (default value 10). If the test targets the border router and the parent, 10 ping attempts would be sent to the border router followed by 10 attempts targeting the parent.
- **Packet length**: Defines the payload size of the ping packet sent to the targets (default value 40). The size corresponds to the ICMPv6 payload including type field (1 byte), code (1 byte), checksum (2 bytes), identifier (2 bytes), sequence number (2 bytes), and finally data field (dependent on the packet length configuration).
- **Target devices**: Defines which Wi-SUN devices are targeted by the ping test. There are three options:
  - Every known device, that is, the border router, the parent, and the device children
  - The border router
  - The parent

Note: If the device running the Wi-SUN Network Measurement application is connected directly to the border router, the parent device and the border router are equivalent.

3 LCD Screen Interface

The first interface that can be used to interact with the Wi-SUN Network Measurement application is the mainboard LCD screen and two push buttons. They provide an easy-to-use interface to configure/start ping tests and consult the test results. The LCD interface can leverage most of the application features with few limitations related to the limited interface it offers (for example, it is not possible to input an IPv6 address to start a test with a “unknown” device).

Note: When using the LCD screen interface, keep track of the actions associated with each push button because they can change depending on the context.

3.1 Connection Screen

The Wi-SUN Network Measurement application starts with a connection attempt to the Wi-SUN network configured in the project. The network name is displayed in the upper part of the screen. The connection steps are shown in the lower section (Wi-SUN connection steps 1 to 5 on the LCD screen).

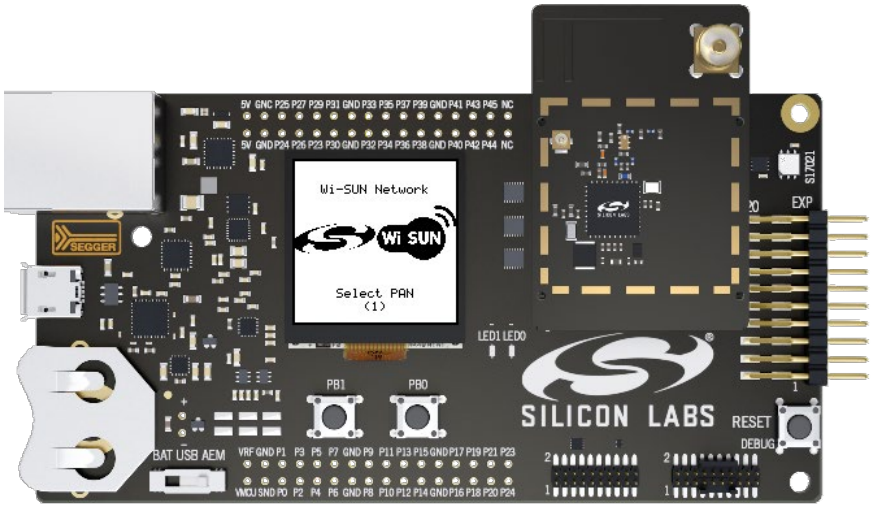


Figure 3.1. Connection Screen

When the node is successfully connected to the Wi-SUN network, the Wi-SUN Network Measurement application switches the screen to display the main menu.

Note: If the Wi-SUN Network Measurement application is stuck in the connection step 1 **Select PAN**, verify the configuration used matches the border router configuration (network name and Wi-SUN PHY used).

3.2 Main Menu

The Network Measurement main menu provides access to the Wi-SUN Network Measurement application main features. Scroll down through the options using push button 1 (**PB1**). Select a sub-menu using **PB0**.

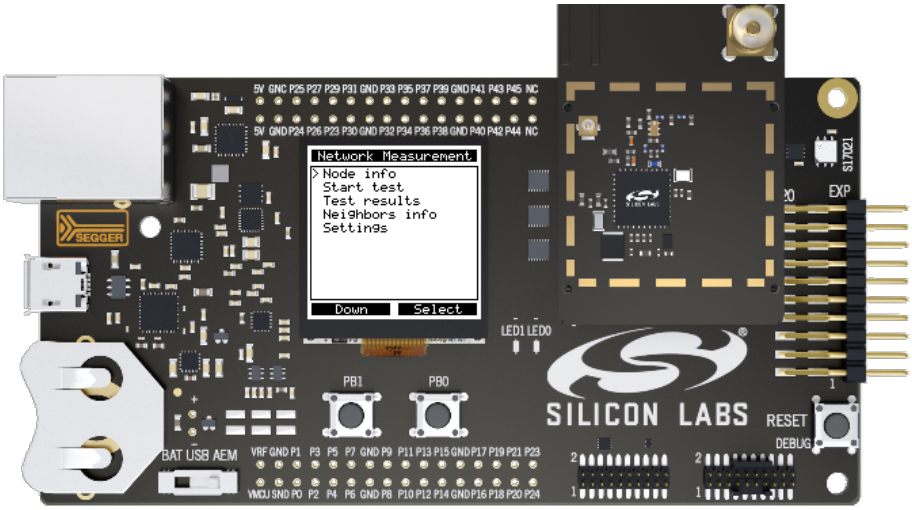


Figure 3.2. Main Menu

3.3 Node Info Screen

The Node Info screen displays information related to the Wi-SUN device and its configuration (network name, Wi-SUN PHY configuration, TX power, IPV6 addresses, and so on). You can scroll through the information available using **PB0**.

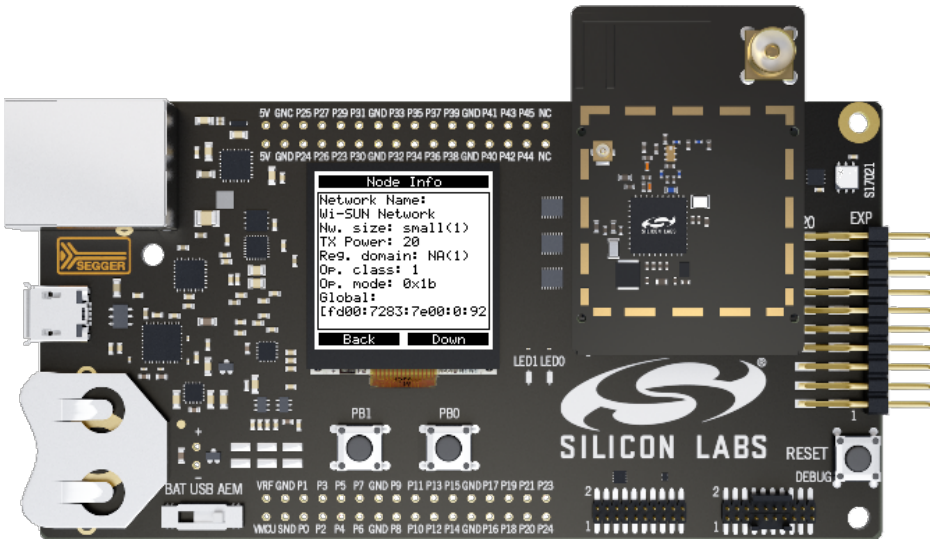


Figure 3.3. Node Info Screen

3.4 Start Test Option

If you select **Start Test** on the main menu, the Wi-SUN Network Measurement application starts a ping test with the settings previously configured with the other menus. A progress bar indicates the number of ping packets sent over the packet count configured for the test. The test is replicated for each configured target device (border router, parent, and children).

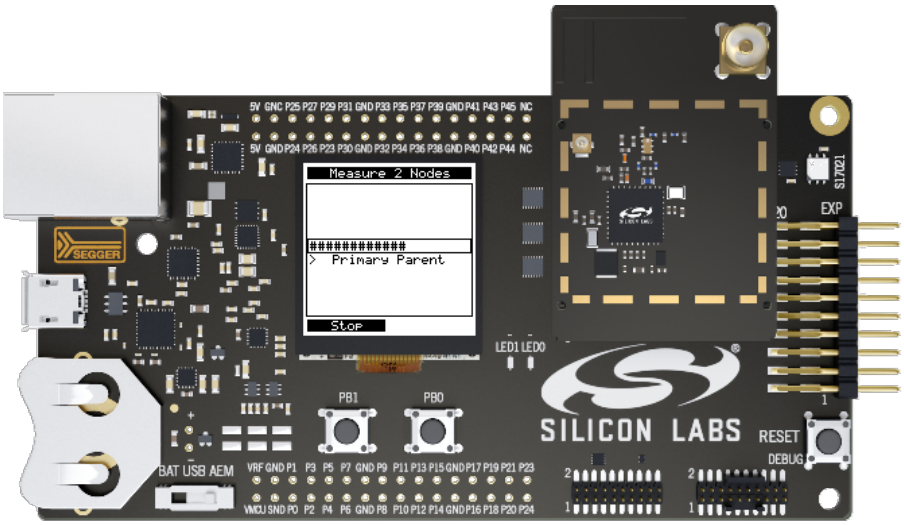


Figure 3.4. Test in Progress Screen

You can stop the test for a given target at any time by pressing **PB1**. When the tests are completed, pressing **PB0** opens the test results display.

3.5 Test Results Display

You can access the Test Results display either after a test or from the main menu. The results are split for each available target. When you select a target, you access its test results, including the ping packet loss and associated percentage, the minimum/maximum/average ping latency, the test settings, and the associated RPL metrics.

3.6 Neighbors Info Screen

The Neighbors Info screen provides access to metrics individualized per neighbor device. This includes the mandatory parent in the Wi-SUN network but also all the potential children using the device to communicate with the rest of the network. The information includes the device IPv6 address and numerous RPL-related metrics.

3.7 Settings Menu

The Settings menu enables you to configure the ping test settings like packet count, packet length, and target devices. In each sub-menu, the currently configured value is displayed in the upper black box. You can select a new value using the two push buttons.

In the following figure, the application packet count is currently configured to 10 packets.

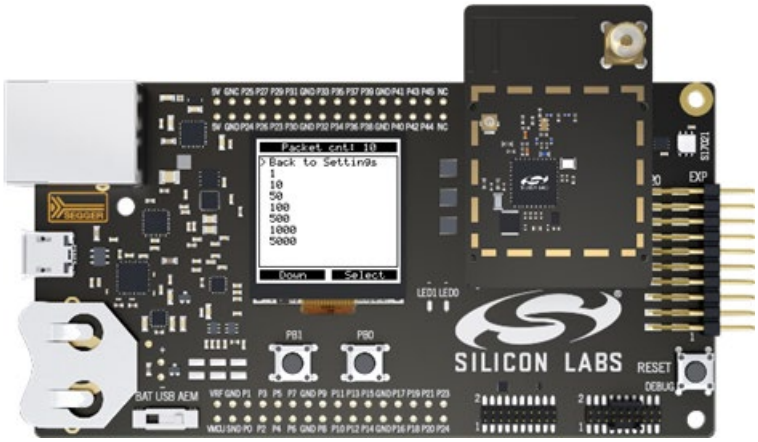


Figure 3.5. Packet Count Configuration Menu

4 Command Line Interface

The CLI provides an alternative interface to the LCD screen with additional features and test scripting. In addition to exposing a broader API, the CLI enables you to:

- Configure the network name, network size, and TX output power.
- Control the connection state.
- Connect or disconnect the Wi-SUN node.
- Input IPv6 addresses to perform ping tests with devices other than the border router, parent, and children.
- Configure IPv6 UDP iperf tests

The Wi-SUN Network Measurement application leverages the common Wi-SUN application CLI for standard Wi-SUN actions. It adds a specific “measure” command that can start ping tests equivalent to the one triggered from the LCD screen interface.

Three command domains are available: ‘about’, ‘wisun’ and ‘iperf’. You can list these by executing **help**:

```
> help

  about                         Get info about the running app
  iperf                         iPerf2 commands
  wisun                         Wi-SUN commands
```

The ‘about’ command domain is limited to displaying information about the current application (it is provided by the ‘app info’ component)

```
about
Wi-SUN Network Measurement Application
Versions:
  * Compiler (GCC): 10.3.1
  * Micrium OS kernel: 5.12.1
  * Wi-SUN: 1.1.0
  * mbed TLS: 3.1.0

>
```

You can access the list of ‘wisun’ commands by executing **wisun help** at the command prompt:

```
> wisun help

  join_fan10                    Connect to a Wi-SUN network using FAN1.0 settings
  join_fan11                    Connect to a Wi-SUN network using FAN1.1 settings
  join_explicit                 Connect to a Wi-SUN network using explicit PHY settings
  disconnect                    Disconnect from the Wi-SUN network
  set                           Set a variable
                                [*] empty | help | [string] Key [string] Value
  get                           Get a variable
                                [*] empty | help | [string] Key
  save                          Save variables to non-volatile storage
  reset                         Reset variables to default settings
  measure                       Measure a remote host or quick measurement
                                [string] Remote address or 'all' or 'parent' or 'br'
                                [uint16] Count of measurement packets
                                [uint16] Measurement packet length
  ping                          Ping a remote host

                                [string] Remote address
```

and the list of ‘iperf’ commands by executing iperf help at the command prompt:

```
> iperf help
  get                           Get configuration parameters
                                [*] empty | [string] Key
  set                           Set configuration parameters
                                [*] empty | [string] Key [string] Value
  server                        Start preconfigured iPerf Server test
```

```
                                [*] help
  client                        Start preconfigured iPerf Client test
                                [*] help
```

For ‘wisun’ or ‘iperf’ command domains, ‘set’ and ‘get’ functions are available to check (‘<domain> get’) or change (‘<domain> set’) current settings. To only list existing settings, use ‘<domain> get <subdomain> help’

The Wi-SUN Network Measurement application maintains Wi-SUN network settings that you can access by executing **wisun get wisun**. This returns all the settings available in the Wi-SUN CLI component. You can check whether a setting is only readable by executing **wisun get wisun help** or if it can be modified (by executing **wisun set**) as shown in the following examples.

```
> wisun get wisun help

[rw] wisun.network_name
[rw] wisun.network_size
[rw] wisun.tx_power
[rw] wisun.regulatory_domain
[rw] wisun.operating_class
[rw] wisun.operating_mode
[ro] wisun.connection_state
[ro] wisun.ip_address_global
[ro] wisun.ip_address_link_local
[ro] wisun.ip_address_border_router
[rw] wisun.regulation                                   Regional regulation [uint8]
[rw] wisun.regulation_warning_threshold                 Transmission warning threshold in
                                                        percent (-1 to disable) [int8]

[rw] wisun.regulation_alert_threshold                  Transmission alert threshold in percent
                                                       (-1 to disable) [int8]
```

Similarly, the writable settings are listed when you execute **wisun set wisun help**.

```
> wisun set wisun help

[rw] wisun.network_name
[rw] wisun.network_size
[rw] wisun.tx_power
[rw] wisun.regulatory_domain
[rw] wisun.operating_class
[rw] wisun.operating_mode
[rw] wisun.regulation                                   Regional regulation [uint8]
[rw] wisun.regulation_warning_threshold                 Transmission warning threshold in
                                                        percent (-1 to disable) [int8]

[rw] wisun.regulation_alert_threshold                  Transmission alert threshold in percent

                                                       (-1 to disable) [int8]
```

(This list is the same as the ‘get’ list without the read-only ([ro]) values)

The iperf command domain has addition help about the possible parameters, accessible using **iperf get help**

```
> iperf get help
Help of iperf 'get' and 'set' methods

Available sub-domains :
 options
 results

[csut]: corresponds to client/server/udp/tcp. Indicates in which case the parameter applies

Type 'iperf [get or set] <sub-domain> help'

eg. 'iperf get options help' to get the help of all options
    'iperf get options' for all current options values
```

```
    'iperf get results' for test results

    'iperf get options.<option>' for a specific option value
```

This shows that there are 2 iperf sub-domains: ‘options’ and ‘results’.

Since options may differ depending on the test case (Client or Server, UDP or TCP), an indicator is added on each line:

```
[csut]: corresponds to client/server/udp/tcp. Indicates in which case the parameter applies
```

So, options can be displayed using **iperf get options help**

```
iperf get options help
[csu ] options.port                    Server port to listen on/connect to [uint16] (default 5001)
[c u ] options.remote_addr             IPv6 remote host address  [string]
[c u ] options.bandwidth               Unused if packet_number is set. Otherwise used to compute
                                       Packet_number as bandwidth*duration/buffer_length  [uint32]
                                       (default 20000)
[csu ] options.bw_format               Unused if packet_number is set. One of [bits/s, Kbits/s,
                                       Mbits/s, Gbits/s, bytes/s, Kbytes/s, Mbytes/s, Gbytes/s]
                                       [string] (default bits/s)
[c u ] options.packet_number           Number of packets to send. (Rules out bandwidth)  [uint16]
                                       (default 0=unset)
[c u ] options.buffer_length           buffer_length
[c u ] options.duration                Test duration in seconds [uint16] (default 10)
[csu ] options.interval                Seconds between periodic bandwidth reports  [uint16]
                                       (default 1)

>
```

Above, we see that the iperf server has only a limited number of options: port, bw_format and interval, while the iperf client has more.

By default, options.packet_number is not set, and the actual value used during an iperf client test is computed based on bandwidth, duration and buffer_length. This is used for most tests.

Similarly, results can be displayed using **iperf get results help**

```
> iperf get results help
[csu ] results.json                    Last test result in json format
[csu ] results.text                    Last test result in text format
```

5 IPv6 UDP iperf testing

It is possible to test iperf throughput in both direction:

Using **iperf client**, the Wi-Sun node is the client sending Datagrams to an iperf server which needs to be started on another IPv6 address.

Using **iperf server**, the Wi-SUN node is the server and is waiting for datagrams coming from another IPv6 address on the defined options.port (default 5001, the usual iperf port).

In both cases:

- The Border Router must be started
- The Wi-SUN node(s) need to be connected to the Wi-SUN network
- We test using nodes directly connected to the Border Router (single hop). With multiple hops, the throughput is to be divided by the number of hops.
- The iperf server must be started before the client

5.1 Add IPv6 address to Border Router tun0

To allow using Linux regular iperf (iperf2) on the Border Router, the 'tun0' interface needs to have a known IPv6 address.

We use the IPv6_prefix set for wsbrd with suffix '::1'. It is visible in the tun0 Iv6 address or in the wsbrd configuration file

Retrieving the tun0 IPv6 prefix

```
ip address show tun0 | grep global
```

```
inet6 fd00:6172:6d00:0:4de2:8ac8:d8cb:780f/64 scope global dynamic mngtmpaddr stable-privacy
```

Adding a fixed/known IPv6 address to tun0

```
sudo ip -6 address add <IPv6_prefix[0:14]>::1 dev tun0
```

example:

```
sudo ip -6 address add fd00:6172:6d00::1 dev tun0
```

Check:

```
ip address show tun0 | grep global | grep ::1
```

```
inet6 fd00:6172:6d00::1/64 scope global
```

Using this method, it is possible for any Wi-SUN node to know what this address is, checking its own 'global' IPv6 address and replacing the last digits with '::1'.

5.2 Wi-SUN node as client

We test the Wi-SUN node as a client (sending Datagrams to the server) using the Raspberry Pi supporting the Border Router application, wsbrd. The iperf server must be started first, then the client.

On the Linux Border Router:

- (One time) Install iperf2 (`apt-get install iperf`) if not already installed.
- (After each restart of wsbrd) Add a fixed/known IPv6 address to tun0 as described above.
  - **Note**: You will need to set this IPv6 address on the client as 'options.remote_addr'
- Start iperf2 as a UDP IPv6 server

```
iperf --server --udp --ipv6_domain --interval 1 --format KBytes
```

  - or (using short arguments)

```
iperf -s -u -V -i 1 -f KBytes
```

- The iperf server will start waiting for incoming data:

```
------------------------------------------------------------
```

```
Server listening on UDP port 5001
UDP buffer size:  176 KByte (default)
------------------------------------------------------------
```

On the Wi-SUN Node:

Note: Apart from the server's IPV6 address, all default values should match a typical iperf test and can keep their default values.

- Check that the Wi-SUN Node is connected to the Wi-SUN network

  ```
  > wisun get wisun.connection_state

  wisun.join_state = Operational (5)
  ```

- Check that the parent is the Border Router (to make sure we're testing on a single hop)

  TBD (not available in current iperf app)

- Set the IPv6 address to match the fixed server's IPV6 address set above

  ```
  wisun get wisun.ip_address_global

  wisun.ip_address_global = fd00:6172:6d00:fe6d:5a15
  ```

  (where the IPV6 prefix is the 14 first characters)

  ```
  iperf set options.remote_addr <server's IPV6 address>
  ```

  example:

  ```
  iperf set options.remote_addr fd00:6172:6d00::1
  ```

- (optionally) Set the bandwidth format. This is the unit used by options.bandwidth. we recommend setting it to 'bits/s'.

  ```
  iperf set options.bw_format bits/s
  ```

- (optionally) Set the number of payload bytes to the maximum before fragmentation occurs using UDP, i.e.1234. This is to maximize the throughput.

  ```
  iperf set options.buffer_length 1234
  ```

- (optionally) Set the test duration in seconds (default = 10)

  ```
  iperf set options.duration 60
  ```

- (optionally) Set the iperf port (default = 5001)

  ```
  iperf set options.port 5001
  ```

- (optionally) Set the iperf bandwidth (default = 20000). The bandwidth should be in a range attainable by the selected Wi-SUN PHY. In our initial tests, we're using Wi-SUN FAN, NA-915MHz, 1b (2FSK 50kbps mi=1.0), so 50 kbits/sec is the maximum we could expect if using 100% of the link as a permanent link (which is not the case, according to Wi-SUN specification). Half this is what we will reasonably expect to achieve.

  ```
  iperf set options.bandwidth 20000
  ```

- (optionally) Set the display interval in seconds (default = 1)

  ```
  iperf set options.interval 1
  ```

- (optionally) Set the number of packets to transmit (default = 0/unset).
  - Set this only if you want to set an exact number of packets instead of options.bandwidth for options.duration seconds.

  ```
  iperf set options.packet_number 10
  ```

- Check your options settings (only checking relevant options, marked with [c.u.])

  ```
  iperf get options
  [csu ] options.port = 5001
  ```

```
[c u ] options.remote_addr = "fd00:6172:6d00::1"
[c u ] options.bandwidth = 20000
[csu ] options.bw_format = bits/s
[c u ] options.packet_number = 0
[c u ] options.buffer_length = 1234 bytes
[c u ] options.duration = 60 s
[csu ] options.interval = 1 s
```

- Start the iperf client test
    - *iperf client*

Note: In this setup, the iperf server being running on a Linux machine doesn't need to be restarted upon each test. It can be left running.

Test results are displayed in the Wi-SUN Node's console (Client) and in the Linux (Server) console in a format very similar to regular iperf2.

5.3 Wi-Sun Node as server

Here we test the Wi-SUN node as an iperf server (receiving Datagrams from clients) using the Raspberry Pi supporting the Border Router application, wsbrd as the client. As before, the iperf server must be started first, then the client. The difference with regular iperf is that **iperf server** must be called once per test.

On the Wi-SUN Node:

Note: Apart from the server's IPV6 address, all default values should match a typical iperf test and can keep their default values.

- Check that the Wi-SUN Node is connected to the Wi-SUN network

```
> wisun get wisun.connection_state

wisun.connection_state = Operational (5)
```

- Check that the parent is the Border Router (to make sure we're testing on a single hop)

  TBD (not available in current iperf app)

- (optionally) Set the bandwidth display format.

```
iperf set options.bw_format bits/s
```

- (optionally) Set the iperf port (default = 5001)

```
iperf set options.port 5001
```

- (optionally) Set the display interval in seconds (default = 1)

```
iperf set options.interval 1
```

- Check your options settings (only checking relevant options, marked with [.su.])

```
iperf get options
[csu ] options.port = 5001
[csu ] options.bw_format = bits/s
[csu ] options.interval = 1 s
```

- Display the global IPv6 address

```
wisun get wisun.ip_address_global

wisun.ip_address_global = fd00:6172:6d00:0:5e02:72ff:fe96:c6a3
```

- Start the iperf server test
    - *iperf server*

Note: In this setup, the iperf server is running on the Wi-SUN node and only accepts one test. **iperf server** needs to be relaunched for each test.

On the Linux Border Router:

- (One time) Install iperf2 (`apt-get install iperf`) if not already installed
- Start iperf2 as a UDP IPv6 client using the Wi-SUN node's global IPv6 address

```
iperf --client fd00:6172:6d00:0:5e02:72ff:fe96:c6a3 --udp --ipv6_domain --interval 1 --time 10
--bandwidth 20 --format k
```

  - or (using short arguments)

```
iperf -c fd00:6172:6d00:0:5e02:72ff:fe96:c6a3 -u -V -i 1 -t 10 -b 20 -f k
```

Test results are displayed in the Wi-SUN Node's console (Client) and in the Linux (Server) console in a format very similar to regular iperf2.

5.4 Wi-SUN Throughput Measurement

5.4.1 Maximizing Throughput

To maximize the iperf throughput, we set on the Wi-SUN node options.buffer_length to the maximum size before fragmentation occurs of 1234 bytes. In this case, the Wi-SUN overhead (on top of UDP) is minimal, and user throughput is maximum.

Again, to maximize the iperf throughput, the iperf implementation in the Wi-SUN Network Measurement application uses the UDP protocol (TCP would require acknowledges which would negatively impact the throughput).

5.4.2 Finding the Throughput limit

Due to the characteristics of a Wi-SUN network, where PAN advertising can take a certain time (it requires sending a PAN Advertisement on each channel, and there are often as much as 129 channels), using an iperf test to measure network throughput needs to be taken cautiously.

During the iperf test there will be periods of time when iperf data flows without interruption, leading to a momentary 'best case' throughput.

There will also be periods when network signaling takes precedence over data transmission. When network signaling occurs, the iperf throughput will be negatively impacted.

To get a valid throughput value to compare between PHY configurations, we look for the **max sustainable throughput without packet loss** on a long duration iperf test of **at least 60 seconds**. The idea is that a 60 second test with a given bandwidth will fill up the buffers up to overflowing if the stack is unable to cope with the requested throughput.
The effect of such overflowing is regularly lost packets during the duration of the test, reflecting in the final server report as a non-zero lost packets count.

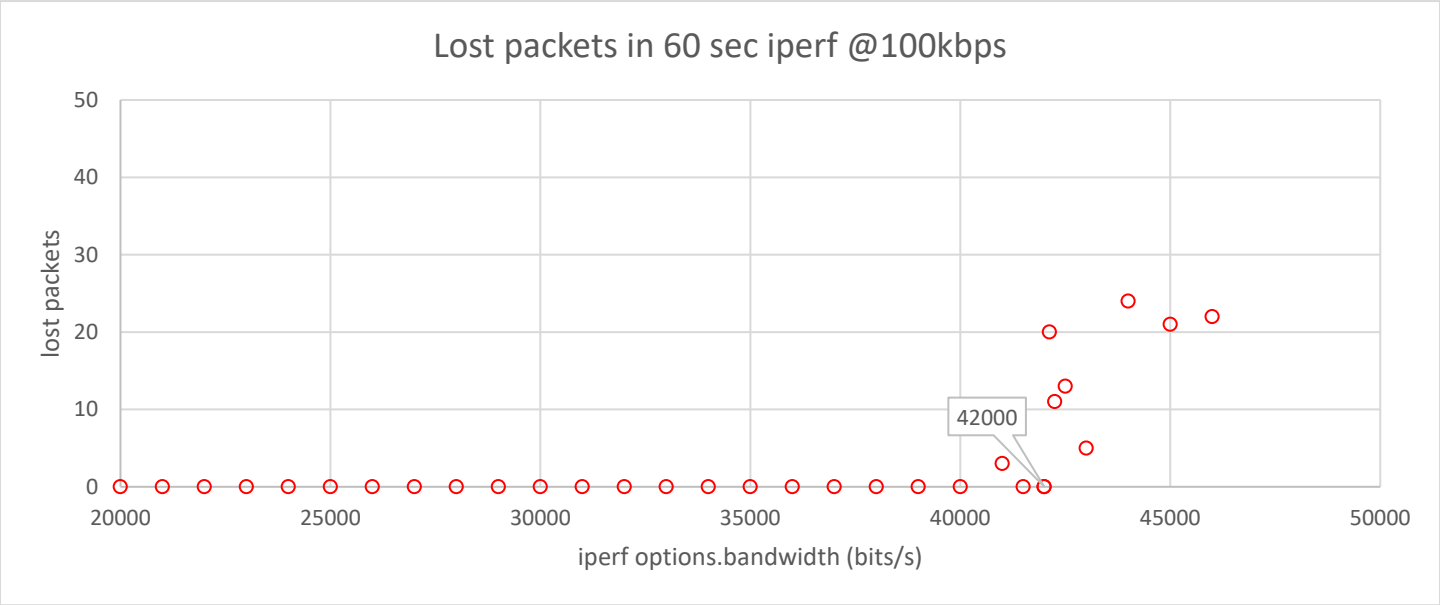
The test consists in finding the maximum options.bandwidth which allows 0 lost packet over 60 sec. We approach this limit through several consecutive iperf client tests.The figure we look for is the number of lost packets on the server side (since we're using UDP, the client doesn't know if the data passes through).

- If the number of lost packets is 0, the Wi-SUN PHY can withstand such a throughput. Increase the options.bandwidth and retest.
- If the number of lost packets is NOT 0, the Wi-SUN PHY is unable to withstand such a throughput. Decrease the options.bandwidth and retest.

Do this is decreasing steps until you have a small enough options.bandwidth step (we used a 1000 bit step in our tests).

The value we keep is the maximum options.bandwidth without packet loss.

Below is an illustration of such a limit search, where the limit is 42000 bit/s:



6 CoAP Communication Between the Linux Border Router and the Network Performance Application

The Network Measurement Application supports a CoAP server to allow the Wi-SUN Border Router to get data from the node using the linux libcoap client. Follow the steps below to get CoAP network performance data on your Border Router, without the need to directly access the CLI on the Wi-SUN node.

6.1 Install the libcoap library on the Border Router

```
$ sudo apt-get install libcoap-1-0
$ sudo apt-get install libcoap-1-0-bin
```

6.2 Get your node's global address

From the node cli, make sure that you are connected to the network.

```
> wisun get wisun.connection_state
wisun.connection_state = Operational (5)
```

Then, get your global IPv6 address

```
> wisun get wisun.ip_address_global
wisun.ip_address_global = fd12:3456::20d:6fff:fe20:be0a
```

6.3 Discover the Available Resources

The CoAP protocol supports an interoperable discovery feature. A CoAP client can request the attributes hosted by a CoAP server. In the case of the Network Performance application, the available resources can be retrieved using the libcoap GET method with the standard discovery entry-point.

```
$ coap-client -m get -N coap://[fd12:3456::20d:6fff:fe20:be0a]:5683/.well-known/core
v:1 t:NON c:GET i:32bf {} [ ]
</cli>;ct=40,</cli/nbrinfo>;rt="nbrinfo";if="cli",</cli/iperf>;rt="ip-
erf";if="cli",</cli/ping>;rt="ping";if="cli"
```

The above shows that we have access to the following commands:

</cli>;ct=40,

</cli/nbrinfo>;rt="nbrinfo";if="cli",

</cli/iperf>;rt="iperf";if="cli",

</cli/ping>;rt="ping";if="cli"

Check the node and Border Router info using:

$ coap-client -m get -N coap://[fd00:6172:6d00:0:b6e3:f9ff:fea6:3aa]:5683/cli/nbrinfo

```
v:1 t:NON c:GET i:ff62 {} [ ]
{
"Border Router":
{"addr":"fd00:6172:6d00:0:92fd:9fff:fe6d:5a15","lt":0,"txc":0,"txf":0,"txmsc":0,"txmsf":0,"rplr":0,
"etx":0,"rslo":0,"rsli":0},
"Primary Parent":
{"addr":"fd00:6172:6d00:0:92fd:9fff:fe6d:5a15","lt":2037,"txc":2096,"txf":9,"txmsc":0,"txmsf":0,"rp
lr":128,"etx":150,"rslo":142,"rsli":138},
"Child0":
{"addr":"fd00:6172:6d00:0:5e02:72ff:fe96:c97f","lt":2159,"txc":162,"txf":0,"txmsc":0,"txmsf":0,"rpl
r":65535,"etx":276,"rslo":152,"rsli":151}
}
```

6.4 Calling iperf using CoAP

It is possible to get the available iperf options using the '-e' text payload coap-client parameter:

$ coap-client -m get -N -e "iperf help" coap://[fd00:6172:6d00:0:b6e3:f9ff:fea6:3aa]:5683/cli/iperf

```
v:1 t:NON c:GET i:4807 {} [ ]

iPerf Remote Control CLI Help.

  Writable options:
    -options.remote_addr
    -options.bandwidth
    -options.duration
    -options.interval
  Example: iperf set options.bandwidth 10000

  Start iPerf command: iperf <client|server>
```

Set the iperf options using:

$ coap-client -m get -N -e "iperf set options.bandwidth 10000" coap://[fd00:6172:6d00:0:b6e3:f9ff:fea6:3aa]:5683/cli/iperf

```
v:1 t:NON c:GET i:d665 {} [ ]
10000
```

$ coap-client -m get -N -e "iperf set options.remote_addr fd00:6172:6d00:0:92fd:9fff:fe6d:5a15" coap://[fd00:6172:6d00:0:b6e3:f9ff:fea6:3aa]:5683/cli/iperf

```
v:1 t:NON c:GET i:4c8a {} [ ]
fd00:6172:6d00:0:92fd:9fff:fe6d:5a15
```

Launch the iperf test using:

$ coap-client -m get -N -e "iperf client" coap://[fd00:6172:6d00:0:b6e3:f9ff:fea6:3aa]:5683/cli/iperf

```
v:1 t:NON c:GET i:10d8 {} [ ]
{
  "Test-3": {
    "id":     3,
    "status": "DONE",
    "error":  "NONE",
    "options": {
      "mode":        "CLIENT",
      "protocol":    "IPV6_UDP",
      "port":        5001,
      "remote_addr": "fd00:6172:6d00::192fd:9fff:fe6d:5a15",
      "bandwidth":   10000,
      "packet_nbr":  10,
      "buf_len":     1234,
      "duration_ms": 10000,
      "win_size":    0,
      "persistent":  false,
      "interval_ms": 1000,
      "bw_format": "bits/s"
    },
    "statistic": {
      "nbr_calls":             10,
      "bytes":                 12340,
      "tot_packets":           10,
      "nbr_rcv_snt_packets":   10,
      "errs":                  0,
      "transitory_error_cnts": 0,
      "last_recv_pkt_cnt":     0,
      "ts_curr_recv_ms":       0,
      "ts_prev_recv_ms":       0,
```

```
      "ts_curr_sent_ms":       19794004,
      "ts_prev_sent_ms":       0,
      "udp_jitter":            0,
      "udp_rx_last_pkt":       0,
      "udp_lost_pkt":          0,
      "udp_out_of_order":      0,
      "udp_dup_pkt":           0,
      "udp_async_error":       false,
      "end_err":               false,
      "ts_start_ms":           19783880,
      "ts_end_ms":             19794012,
      "bandwidth":             9743,
      "finack_tot_len":        0,
      "finack_duration_ms":    0,
      "finack_pkt":            0
    }
  }
}
```

7 Ping Latency Improvement

Using the Wi-SUN Network Measurement application in its default configuration (Lowest speed Wi-SUN PHY, Network size set to *small,* and default broadcast dwell interval), the ping latency performance is not optimal. This chapter describes ways to optimize the solution ping latency and explains some of the trade-offs associated with the modifications.

7.1 AN1330 Application Note

AN1330: Silicon Labs Wi-SUN Mesh Network Performance reports the ping latency internal tests performed by Silicon Labs in real-world deployments. The tests focus on characterizing the expected Wi-SUN stack latency in point-to-point communications but also in round trip messages travelling over several hops. The tests are performed with the Wi-SUN stack default (non-optimized) configuration using the 50 Kbps North America Wi-SUN PHY. The results provide a good reference point to compare your own test results. Keep in mind that using a different region PHY impacts the results in the following figure.

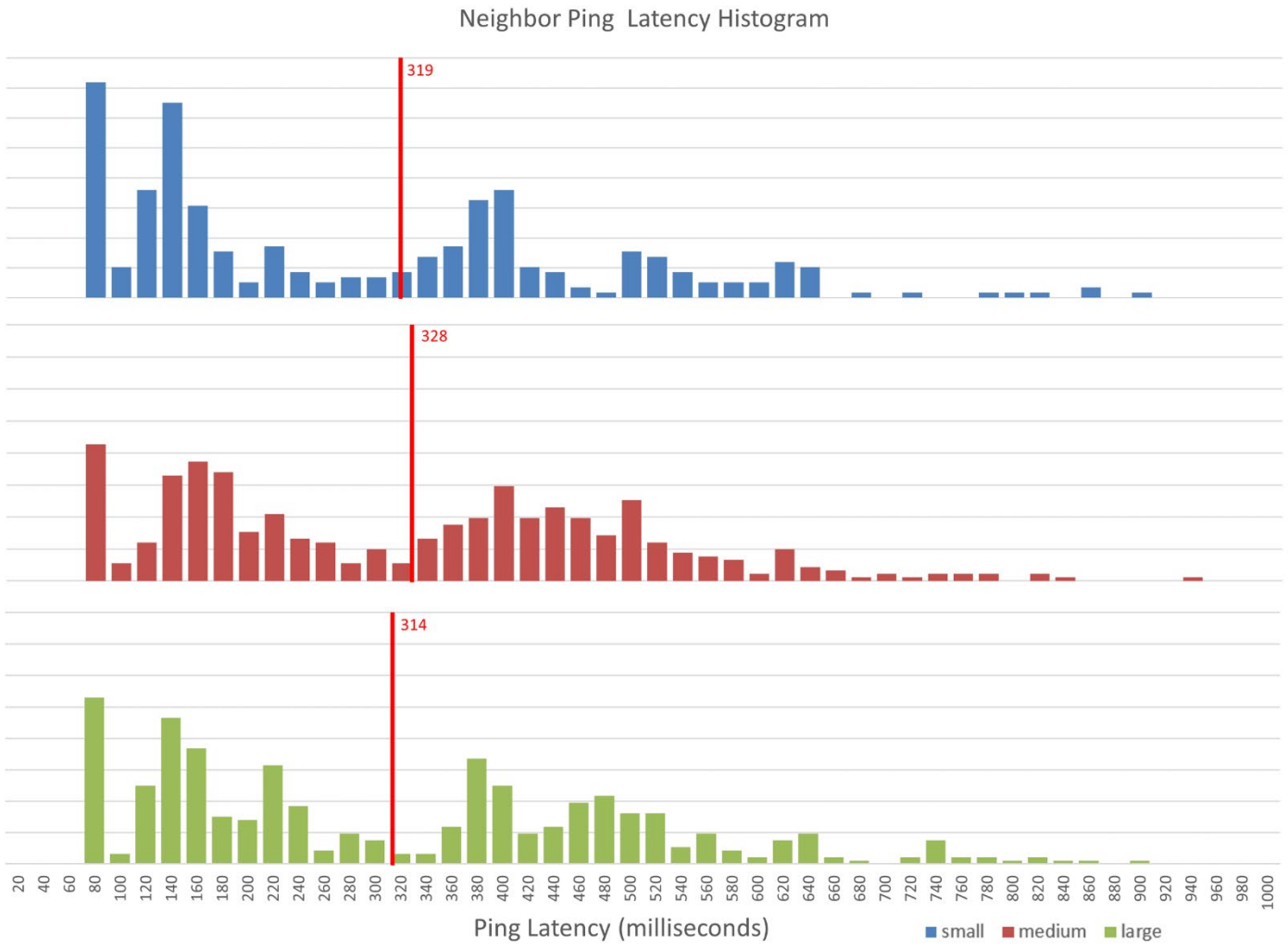


Figure 5.1. Neighbor Ping Latency Histogram

7.2 Wi-SUN PHY Impact

The first critical configuration impacting the ping latency is the Wi-SUN PHY which is used by every device in a Wi-SUN network. A point-to-point ping is composed of four messages over the air. A ping with a payload of 512 bytes triggers 1332 bytes exchanged on the air. It represents approximately 30 milliseconds spent sending messages on the radio medium (see the following figure).

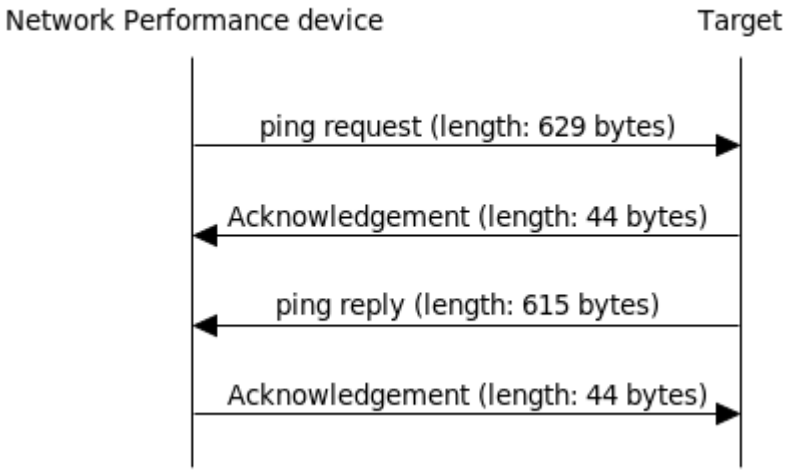


Figure 5.2. 512-byte Ping Diagram

The Wi-SUN PHY also impacts the transmissions triggered to manage the Wi-SUN network. Both the speed and number of channels available in the PHY factor into the penalty that is applied to a ping latency sent close to a network management event. The two main time-consuming network management mechanisms are the PAN advertisement and PAN configuration events. For example, a PAN advertisement sent using the 50 kbps North America PHY using 128 channels holds the radio for approximately two seconds (about 15 milliseconds spent on each channel). The PAN advertisement and configuration events explain the periodical delays that can be noticed during a long ping test. Using a Wi-SUN PHY with a higher modulation speed and fewer channels highly reduces the impact of those network management events on the ping latency (for example, 300 kbps North America PHY with 41 channels). The drawback or trade-offs are the effective distance for a good communication between nodes in the network is reduced. Finally, you need to consider how changing the network size setting can affect the network management event intervals (see 5.3 Network Configuration and Size Setting).

In addition to the PHY speed itself, the frequency hopping mechanism and the associated channel switches can delay transmissions of packets to prevent sending a message on a channel change. The default unicast dwell interval is 255 milliseconds.

7.3 Network Configuration and Size Setting

The network configuration plays an integral role in the way packets flow through the Wi-SUN network. These network configurations impact the ping latency:

- The border router broadcast interval defines at which interval a broadcast dwell interval happens. By default, the border router broadcast interval is set to 1 seconds and 20 milliseconds. You can modify it on the border router side. The configuration of the border router broadcast interval propagates throughout the Wi-SUN network and every device part of the network uses the same interval. If an application focuses on unicast messages, Silicon Labs recommends that you increase the border route broadcast interval.
- The broadcast dwell interval defines the time of a broadcast frequency hopping slot (by default, 255 milliseconds equal to the unicast default dwell interval). The longer the broadcast dwell interval, the less time remains for unicast communications. Silicon Labs recommends that you reduce the broadcast dwell interval for a unicast-focused application. You change this configuration on the border router side.
- The network size setting dictates the intervals between network management events among other things. To optimize the ping latency, Silicon Labs recommends that you use the *large* or *medium* configuration depending on your network deployment size to increase the network management exchanges interval. Be aware of the potential trade-off with the network connection speed and network configuration update speed.

Considering all the options available for a Wi-SUN solution developer, you can greatly improve the ping latency from the initial results retrieved from the default solution configuration. Keep in mind that every setting change is a trade-off with another Wi-SUN network performance component.

Simplicity Studio

One-click access to MCU and wireless tools, documentation, software, source code libraries & more. Available for Windows, Mac and Linux!

IoT Portfolio
www.silabs.com/IoT

SW/HW
www.silabs.com/simplicity

Quality
www.silabs.com/quality

Support & Community
www.silabs.com/community

SILICON LABS

Silicon Laboratories Inc.
400 West Cesar Chavez
Austin, TX 78701
USA

www.silabs.com