Wi-Fi Power Management

# Introduction

This application note describes the Wi-Fi Connection Manager’s power management APIs available in the SDK.

# WCM Power Management

The WCM power management configuration APIs allow the application to configure the Wi-Fi interface with the following power management flags.

1. WIFI\_PM\_PS\_POLL: Sends PS poll if a beacon was missed.
2. WIFI\_PM\_DYN\_LISTEN\_INT: Dynamic listen interval.
3. WIFI\_PM\_STA\_RX\_NAP: Turns off receiver for uninteresting frames for station.
4. WIFI\_PM\_STA\_ONLY\_BROADCAST: Does not receive multicast frames that are not broadcast (only effective if rx\_nap is used).
5. WIFI\_PM\_TX\_PS: Sends outgoing frames without leaving Wi-Fi power save.
6. WIFI\_PM\_MCAST\_DONT\_CARE: Ignores the multicast flag in beacons. i.e., does not receive multicast or broadcast frames.
7. WIFI\_PM\_DTIM\_ONLY: Wakes up only at effective listen interval and does not switch to listen to every beacon in case of beacon miss.

This application note demonstrates usage of the above flags.

## WCM Power Management APIs

Following are the WCM power management APIs used in this application

1. wcm\_pm\_config()

Sets the Power Management configuration of a Wi-Fi interface.

1. wcm\_pm\_config\_get()

Gets Power Management configuration of a Wi-Fi interface.

# Source Code Walkthrough

## WCM\_PM with Different Listen Interval

### Overview

The sample code is in the path *example/wcm\_pm/src/wcm\_pm.c*. This describes how the API wcm\_pm\_config()is used to disable the pm\_flags to achieve optimized power consumption. The two power-save parameters being used in code are explained below:

1. Listen interval

Specifies how often the device will wake-up and listen for beacons. If the listen\_interval is set to a number which is not an even multiple of DTIMs, the actual listen interval will be set to an even multiple of DTIMs which is higher than listen\_interval.

1. Traffic timeout (ms)

Keeps the device awake for configured time (default 12ms) after any incoming or outgoing traffic. The Wi-Fi interface will go to Wi-Fi power save if no traffic has occurred for the traffic timeout time in milliseconds. A traffic timeout value of 0 will make the device stay awake forever.

### Sample Code Walkthrough

1. Connect to a Wi-Fi network

To connect to a Wi-Fi network, wcm\_create()API from the Wi-Fi Connection Manager is used. Initially, the Wi-Fi network interface is created using wcm\_create().

|  |
| --- |
| h = wcm\_create(NULL); |

wifi\_connect\_to\_network()API, from components library, connects to the Wi-Fi network using the AP credentials provided.

|  |
| --- |
| rval = wifi\_connect\_to\_network(&h, WCM\_CONN\_WAIT\_INFINITE, &wcm\_connected);  if(rval < 0) {  os\_printf("\nError: Unable to connect to network\n");  return 0; |

1. Print Power Management Configuration

The function reads the power management configuration such as listen interval, traffic timeout and power management flag using wcm\_pm\_config\_get() API.

wcm\_pm\_get\_sleep\_period()gets the current\_sleep\_period value.

|  |
| --- |
| uint32\_t current\_listen\_interval;  uint32\_t current\_traffic\_tmo;  uint32\_t current\_pm\_flags;  int current\_sleep\_period;  wcm\_pm\_config\_get(h, &current\_listen\_interval, &current\_traffic\_tmo,  &current\_pm\_flags);  wcm\_pm\_get\_sleep\_period(h, &current\_sleep\_period);  os\_printf("\n\*\*\*\*\*\*\n");  os\_printf("listen\_interval: %d\n", current\_listen\_interval);  os\_printf("traffic\_tmo: %d\n", current\_traffic\_tmo);  os\_printf("pm\_flags: 0x%x\n", current\_pm\_flags);  os\_printf("sleep\_period: %d ms\n", current\_sleep\_period/1000);  os\_printf("\*\*\*\*\*\*\n");  os\_printf((current\_pm\_flags & WIFI\_PM\_DTIM\_ONLY) ==0 ? "dtim\_only disabled\n" : "dtim\_only enabled\n");  os\_printf((current\_pm\_flags & WIFI\_PM\_TX\_PS) ==0 ? "tx\_ps disabled\n" : "tx\_ps enabled\n");  os\_printf((current\_pm\_flags & WIFI\_PM\_MCAST\_DONT\_CARE)==0 ? "mcast\_don't\_care disabled \n" : "mcast\_don't\_care enabled\n");  os\_printf((current\_pm\_flags & WIFI\_PM\_STA\_RX\_NAP)==0 ? "rx\_nap disabled\n" : "rx\_nap enabled\n");  os\_printf((current\_pm\_flags & WIFI\_PM\_STA\_ONLY\_BROADCAST)==0 ? "only\_broadcast disabled\n" : "only\_broadcast enabled\n");  os\_printf((current\_pm\_flags & WIFI\_PM\_PS\_POLL)==0 ? "ps\_poll disabled\n" : "ps\_poll enabled\n");  os\_printf((current\_pm\_flags & WIFI\_PM\_DYN\_LISTEN\_INT)==0 ? "dli disabled\n" : "dli enabled\n");  os\_printf("\*\*\*\*\*\*\n"); |

1. Set Power Management Configurations

Sets the Power Management configuration of the Wi-Fi interface using wcm\_pm\_config() API.

In this application, the pm\_flag is set to 0 to disable all power management flags. The application sets three different listen\_interval values and stays configured for each of the listen\_interval values for 20 seconds. For all three listen interval values, the traffic time out is set to 12.

Listen Interval 3

|  |
| --- |
| wcm\_pm\_config(h, LISTEN\_INTERVAL\_3, TRAFFIC\_TMO, pm\_mask);  os\_sleep\_us(20000000, OS\_TIMEOUT\_NO\_WAKEUP); |

Listen Interval 10

|  |
| --- |
| wcm\_pm\_config(h, LISTEN\_INTERVAL\_10, TRAFFIC\_TMO, pm\_mask);  os\_sleep\_us(20000000, OS\_TIMEOUT\_NO\_WAKEUP); |

Listen Interval 20

|  |
| --- |
| wcm\_pm\_config(h, LISTEN\_INTERVAL\_20, TRAFFIC\_TMO, pm\_mask);  os\_sleep\_us(20000000, OS\_TIMEOUT\_NO\_WAKEUP); |

### Running the Application

Program wcm\_pm.elf (*sdk\_x.y\examples\wcm\_pm\bi*n) using the Download tool:

1. Launch the Download tool provided with InnoPhase Talaria TWO SDK.
2. In the GUI window:
   1. Boot Target: Select the appropriate EVK from the drop-down
   2. ELF Input: Load the wcm\_pm.elf by clicking on Select ELF File.
   3. AP Options: Provide the SSID and Passphrase under AP Options to connect to an Access Point.
   4. Boot Arguments: Pass the following boot arguments:

|  |
| --- |
| wifi.max\_idle\_period=600, suspend=1 , wifi.arp\_grat\_period=0 |

* + 1. suspend

suspend =1 will trigger Talaria TWO suspend state.

* + 1. wifi.max\_idle\_period

Some APs disconnect stations when no traffic has occurred for a certain time. This parameter specifies how many seconds the device can be idle before it must send a keep alive to the AP. To reduce power consumption, the keep alive messages are aligned to the next beacon reception period. The actual keep alive interval can therefore be longer than specified, especially if the wifi.listen\_interval is set to a high value.

* + 1. wifi.arp\_grat\_period

Period in seconds for gratuitous ARP announcements. Set to zero to disable.

* 1. Programming: Prog RAM or Prog Flash as per requirement.

For more details on using the Download tool, refer to the document: UG\_Download\_Tool.pdf (path: *sdk\_x.y\pc\_tools\Download\_Tool\doc*).

**Note**: x and y refer to the SDK release version. For example: *sdk\_2.6\doc*.

Console output:

|  |
| --- |
| Y-BOOT 208ef13 2019-07-22 12:26:54 -0500 790da1-b-7  ROM yoda-h0-rom-16-0-gd5a8e586  FLASH:PNWWWWWAEBuild $Id: git-d198c0771 $  wifi.max\_idle\_period=600 suspend=1 wifi.arp\_grat\_period=0 np\_conf\_path=/data/nprofile.json ssid=InnoPhase\_AE passphrase=Inno@1234  $App:git-df2ae28b  SDK Ver: sdk\_2.6.2  Wifi PM Demo App  addr e0:69:3a:00:13:90  Connecting to added network : InnoPhase\_AE  [0.783,465] CONNECT:98:da:c4:73:b7:76 Channel:10 rssi:-34 dBm  wcm\_notify\_cb to App Layer - WCM\_NOTIFY\_MSG\_LINK\_UP  wcm\_notify\_cb to App Layer - WCM\_NOTIFY\_MSG\_ADDRESS  [0.837,791] MYIP 192.168.1.164  [0.837,955] IPv6 [fe80::e269:3aff:fe00:1390]-link  wcm\_notify\_cb to App Layer - WCM\_NOTIFY\_MSG\_CONNECTED  Connected to added network : InnoPhase\_AE  Suspend enabled!  setting to listen interval equivalent 3  \*\*\*\*\*\*  listen\_interval: 3  traffic\_tmo: 12  sleep\_period: 0 ms  dtim\_only enabled  tx\_ps enabled  mcast\_don't\_care enabled  rx\_nap enabled  only\_broadcast enabled  ps\_poll disabled  dli disabled  \*\*\*\*\*\*  setting to listen interval equivalent 10  \*\*\*\*\*\*  listen\_interval: 10  traffic\_tmo: 12  sleep\_period: 307 ms  dtim\_only enabled  tx\_ps enabled  mcast\_don't\_care enabled  rx\_nap enabled  only\_broadcast enabled  ps\_poll disabled  dli disabled  \*\*\*\*\*\*  setting to listen interval equivalent 20  \*\*\*\*\*\*  listen\_interval: 20  traffic\_tmo: 12  sleep\_period: 1024 ms  dtim\_only enabled  tx\_ps enabled  mcast\_don't\_care enabled  rx\_nap enabled  only\_broadcast enabled  ps\_poll disabled  dli disabled  \*\*\*\*\*\* |

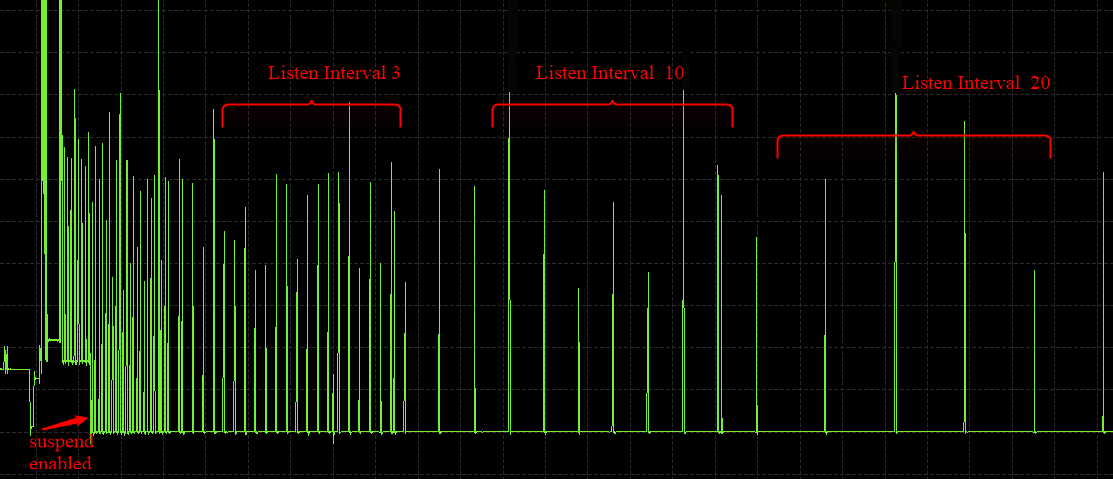


Figure 1: Otti capture

The console output comprises of the different power management configurations the sample application goes through. Figure 1 shows the Otti power consumption, where the listen interval is changed every 20 seconds.

### Wi-Fi Power Management Configuration

1. Configuring the listen\_interval

The DTIM value is typically set by the AP and can be configured inside the AP settings. To save power, Talaria TWO can carry out beacon skipping so that it only listens at certain intervals as specified by the listen\_interval parameter values in wcm\_pm\_config() API.

traffic\_tmo argument represents the traffic time out in milliseconds, described in section 5.1.2 point 2.

pm\_flags represent the power management flags, described in section 4.

|  |
| --- |
| void wcm\_pm\_config(struct wcm\_handle \*h, uint32\_t listen\_interval, uint32\_t traffic\_tmo, uint32\_t pm\_flags) |

For example:

|  |
| --- |
| wcm\_pm\_config(h, 3, 12, 0); |

The line of code sets the listen interval to 3, which is equivalent to DTIM3.

**Note**: If the listen\_interval is set to a number that is not an even multiple of DTIMs, the effective listen interval will be set to an even multiple of DTIMs which is higher than the listen\_interval.

Example 1:

|  |
| --- |
| Configured LI: 10  DTIM set on AP: 3  Effective LI: 12 |

Example 2:

|  |
| --- |
| Configured LI: 2  DTIM set on AP: 3  Effective LI: 6 |

## Rx Nap in Station Mode (WIFI\_PM\_STA\_RX\_NAP)

### Overview

During reception of uninteresting frames in an associated station mode, the receiver can be turned OFF to save power. When the device receives the beginning of a frame, it is possible to check if the frame is intended for this device or not. If the frame is not intended for this device, the Rx nap functionality will turn the receiver OFF (“*take a nap*”) for the duration of the frame. This mode will have very little impact on the power consumption if the cloud power mode is used (refer section 6.1 - Cloud Power Mode for more details).

The frames that are of interest are unicast frames, broadcast frames from AP, multicast (not broadcast) from AP, unless explicitly configured to skip multicast.

Rx nap feature can be used in applications such as sensor or door lock that are not latency sensitive. This feature should be disabled for firmware upgrade application.

The sample code is in: *example/wcm\_pm/src/wcm\_pm\_rxnap.c*

### Sample Code Walkthrough

The following code uses wcm\_pm\_config() API to enable the station Rx NAP flag WIFI\_PM\_STA\_RX\_NAP. Suspend mode is enabled using the API os\_suspend\_enable().

|  |
| --- |
| os\_printf("Suspend enabled!\n");  os\_suspend\_enable(); |

Sets the listen interval to 10, traffic timeout to 12ms and enables Rx nap. Enables sleep mode for 20 seconds before disabling the Rx nap flag.

|  |
| --- |
| print\_wifi\_config()  os\_printf("Sleep mode for next 20 sec\n");  os\_sleep\_us(20000000, OS\_TIMEOUT\_NO\_WAKEUP); |

Sets the listen interval to 10, traffic timeout to 12ms and disables Rx nap.

|  |
| --- |
| wcm\_pm\_config(h, LISTEN\_INTERVAL\_10, TRAFFIC\_TMO, PM\_FLAGS);  print\_wifi\_config();  os\_sleep\_us(20000000, OS\_TIMEOUT\_NO\_WAKEUP); |

### Running the Application

Program wcm\_pm\_rxnap.elf (*sdk\_x.y\examples\wcm\_pm\bin*) using the Download Tool. Refer steps in section 5.3.3 to program the ELF onto Talaria TWO.

Console output:

|  |
| --- |
| Y-BOOT 208ef13 2019-07-22 12:26:54 -0500 790da1-b-7  ROM yoda-h0-rom-16-0-gd5a8e586  FLASH:PNWWWWWAEBuild $Id: git-d198c0771 $  np\_conf\_path=/data/nprofile.json ssid=InnoPhase\_AE passphrase=Inno@1234  $App:git-df2ae28b  SDK Ver: sdk\_2.6.2  Wifi PM Demo Rx nap in associated Mode  addr e0:69:3a:00:13:90  Connecting to added network : InnoPhase\_AE  [1.019,778] CONNECT:98:da:c4:73:b7:76 Channel:10 rssi:-30 dBm  wcm\_notify\_cb to App Layer - WCM\_NOTIFY\_MSG\_LINK\_UP  wcm\_notify\_cb to App Layer - WCM\_NOTIFY\_MSG\_ADDRESS  [1.071,161] MYIP 192.168.1.164  [1.071,325] IPv6 [fe80::e269:3aff:fe00:1390]-link  wcm\_notify\_cb to App Layer - WCM\_NOTIFY\_MSG\_CONNECTED  Connected to added network : InnoPhase\_AE  Suspend enabled!  \*\*\*\*\*\*  listen\_interval: 10  traffic\_tmo: 12  sleep\_period: 0 ms  dtim\_only disabled  tx\_ps disabled  mcast\_don't\_care disabled  rx\_nap enabled  only\_broadcast disabled  ps\_poll disabled  dli disabled  \*\*\*\*\*\*  Sleep mode for next 20 sec  \*\*\*\*\*\*  listen\_interval: 10  traffic\_tmo: 12  sleep\_period: 1024 ms  dtim\_only disabled  tx\_ps disabled  mcast\_don't\_care disabled  rx\_nap disabled  only\_broadcast disabled  ps\_poll disabled  dli disabled  \*\*\*\*\*\* |

1. Rx nap enabled

To test this, ping Talaria TWO IP address from the host machine. When Talaria TWO turns the receiver OFF (*takes a nap*) in Rx nap enabled case, the current consumption dips to 8mA as shown in Figure 2.

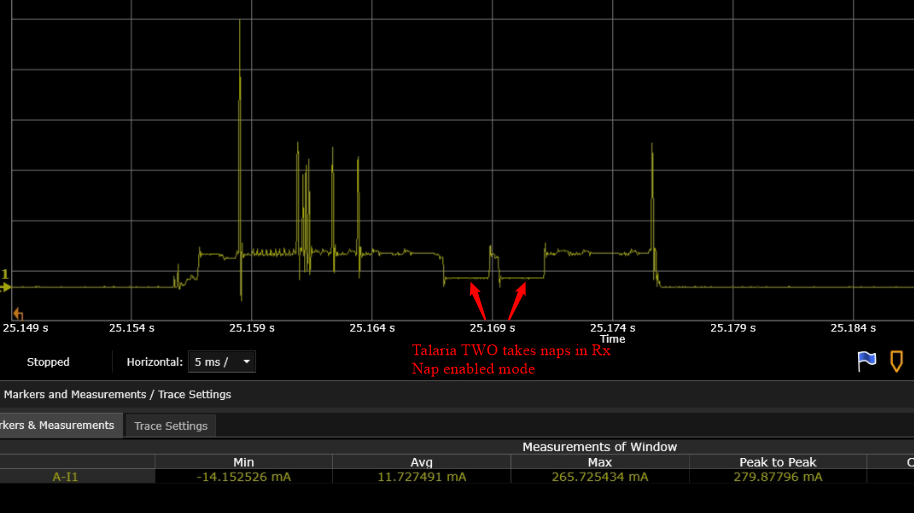


Figure : Power Analyzer- Rx Nap enabled mode

1. Rx nap disabled

When Rx nap gets disabled, Talaria TWO does not take naps for any uninteresting frames.

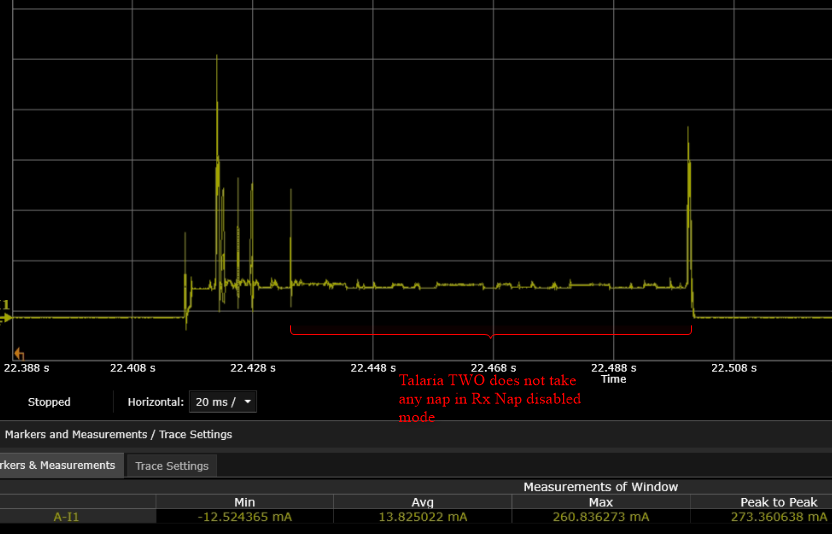


Figure : Power Analyzer- RX Nap disabled mode

## Only Broadcast (WIFI\_PM\_STA\_ONLY\_BROADCAST)

### Overview

This feature will turn the receiver OFF for multicast frames, but listens to the broadcast frames. This function is only activated if Rx nap is also set. In most cases, an application is only interested in broadcast frames to receive ARP requests.

Thus, the MAC header is always received to determine if the frame is of interest or not. The receiver will be turned off for the duration of the frame, which is calculated based on the length and rate information in the PHY header.

The sample code is located in the path example/wcm\_pm/src/wcm\_pm\_bc.c. This describes using the API wcm\_pm\_config()to enable and disable the WIFI\_PM\_STA\_ONLY\_BROADCAST flag.

### Sample Code Walkthrough

The following code uses wcm\_pm\_config() API to enable the only broadcast flag WIFI\_PM\_STA\_ONLY\_BROADCAST and WIFI\_PM\_STA\_RX\_NAP. Here the Rx\_nap is a pre-requisite for using WIFI\_PM\_STA\_ONLY\_BROADCAST flag

|  |
| --- |
| pm\_mask |= WIFI\_PM\_STA\_RX\_NAP | WIFI\_PM\_STA\_ONLY\_BROADCAST;  /\* Enable Only broadcast flag with listen interval set to 1 \*/  wcm\_pm\_config(h, LISTEN\_INTERVAL\_1, TRAFFIC\_TMO, pm\_mask);  print\_wifi\_config();  os\_sleep\_us(60000000, OS\_TIMEOUT\_NO\_WAKEUP); |

Sleep mode is enabled for 20 seconds before disabling this flag again.

|  |
| --- |
| os\_printf("Sleep mode for next 20 sec\n");  os\_sleep\_us(20000000, OS\_TIMEOUT\_NO\_WAKEUP);  os\_printf("\n\*\*\*\*\*\*\n"); |

Next, the only broadcast is disabled by setting the flag to 0.

|  |
| --- |
| wcm\_pm\_config(h, LISTEN\_INTERVAL\_1, TRAFFIC\_TMO, PM\_FLAGS);  print\_wifi\_config();  os\_sleep\_us(60000000, OS\_TIMEOUT\_NO\_WAKEUP); |

### Running the Application

Program wcm\_pm\_bc.elf (*sdk\_x.y\examples\wcm\_pm\bin*) using the Download tool:

1. Launch the Download tool provided with InnoPhase Talaria TWO SDK.
2. In the GUI window:
   1. Boot Target: Select the appropriate EVK from the drop-down
   2. ELF Input: Load the wcm\_pm\_bc.elf by clicking on Select ELF File.
   3. AP Options: Provide the SSID and Passphrase under AP Options to connect to an Access Point.
   4. Programming: Prog RAM or Prog Flash as per requirement.

For more details on using the Download tool, refer to the document: UG\_Download\_Tool.pdf (path: *sdk\_x.y\pc\_tools\Download\_Tool\doc*).

**Note**: x and y refer to the SDK release version. For example: *sdk\_2.6\doc*.

Console output:

|  |
| --- |
| Y-BOOT 208ef13 2019-07-22 12:26:54 -0500 790da1-b-7  ROM yoda-h0-rom-16-0-gd5a8e586  FLASH:PNWWWWWAEBuild $Id: git-d198c0771 $  np\_conf\_path=/data/nprofile.json ssid=InnoPhase\_AE passphrase=Inno@1234  $App:git-df2ae28b  SDK Ver: sdk\_2.6.2  Wifi PM Only Broadcast  addr e0:69:3a:00:13:90  Connecting to added network : InnoPhase\_AE  [0.781,529] CONNECT:98:da:c4:73:b7:76 Channel:10 rssi:-34 dBm  wcm\_notify\_cb to App Layer - WCM\_NOTIFY\_MSG\_LINK\_UP  wcm\_notify\_cb to App Layer - WCM\_NOTIFY\_MSG\_ADDRESS  [2.720,170] MYIP 192.168.1.164  [2.720,218] IPv6 [fe80::e269:3aff:fe00:1390]-link  wcm\_notify\_cb to App Layer - WCM\_NOTIFY\_MSG\_CONNECTED  Connected to added network : InnoPhase\_AE  Suspend enabled!  \*\*\*\*\*\*  listen\_interval: 1  traffic\_tmo: 12  sleep\_period: 102 ms  dtim\_only disabled  tx\_ps disabled  mcast\_don't\_care disabled  rx\_nap enabled  only\_broadcast enabled  ps\_poll disabled  dli disabled  \*\*\*\*\*\*  Sleep mode for next 20 sec  \*\*\*\*\*\*  listen\_interval: 1  traffic\_tmo: 12  sleep\_period: 102 ms  dtim\_only disabled  tx\_ps disabled  mcast\_don't\_care disabled  rx\_nap disabled  only\_broadcast disabled  ps\_poll disabled  dli disabled  \*\*\*\*\*\* |

1. only\_broadcast mode is enabled.

Once this flag is enabled, multicast ping request to Talaria TWO IP address from a Host (laptop/PC) is sent. Since Talaria TWO listens to broadcast frames only, multicast ping fail is as shown in Figure 4.

Note: To add multicast routing to a specific network interface, use the following command:

|  |
| --- |
| route add -net 224.0.0.0/8 dev <interface-name> |

A screenshot of a computer

Description automatically generated with medium confidence

Figure : Multicast ping output – only broadcast enabled

Wireshark capture for the multicast ping: Here, the ping request is sent, but the response is not received as shown in Figure 5.

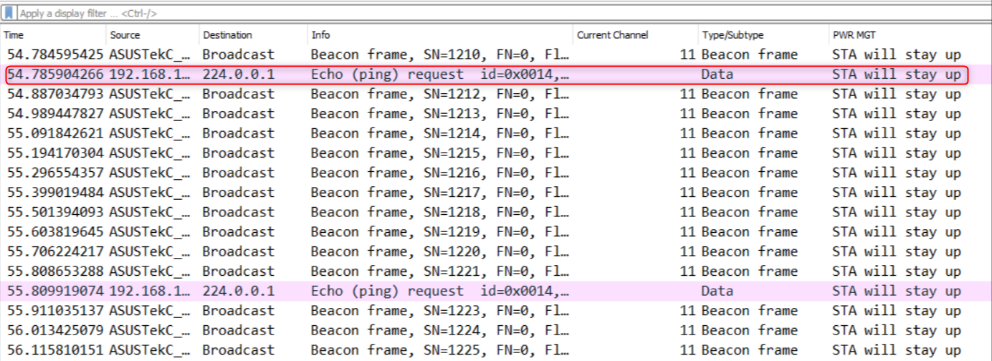


Figure : Wireshark output - only\_broadcast enabled

1. only\_broadcast mode is disabled.

Once this flag is disabled, multicast ping request to Talaria TWO IP address from a Host (laptop/PC) is sent. Multicast ping goes through successfully as shown in Figure 6.

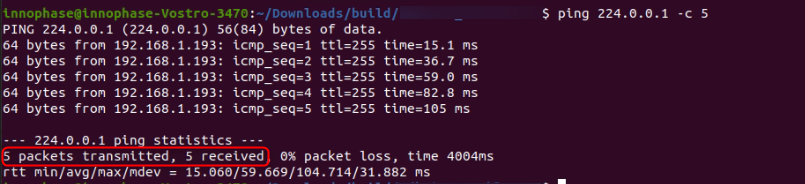


Figure : Multicast ping output – only broadcast disabled

Wireshark captures for the multicast ping: Here, the ping request is sent and a response is received as shown in Figure 7.

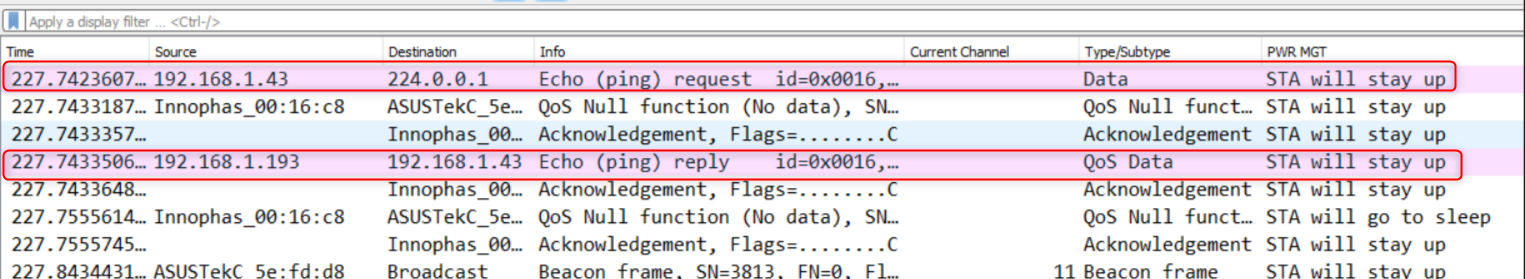


Figure : Wireshark output - only\_broadcast disabled

## Dynamic listen Interval (WIFI\_PM\_DYN\_LISTEN\_INT)

### Overview

When this feature is enabled, Talaria TWO will listen to all beacons if there has been traffic recently till the next listen interval. This will decrease latency for incoming traffic, but will increase power consumption slightly.

The sample code is located in: example/wcm\_pm/src/wcm\_pm\_dli.c. This describes using the API wcm\_pm\_config()to enable and disable the WIFI\_PM\_DYN\_LISTEN\_INT flag.

### Sample Code Walkthrough

The following code uses wcm\_pm\_config() API to enable the dynamic listen interval flag WIFI\_PM\_DYN\_LISTEN\_INT.

|  |
| --- |
| /\* Enable the Dynamic Listen Interval flag \*/  wcm\_pm\_config(h,LISTEN\_INTERVAL\_10,TRAFFIC\_TMO, WIFI\_PM\_DYN\_LISTEN\_INT);  print\_wifi\_config();  os\_sleep\_us(60000000, OS\_TIMEOUT\_NO\_WAKEUP); |

Sleep mode is enabled for 20 seconds before disabling this flag again.

|  |
| --- |
| os\_printf(“Sleep mode for next 20 sec\n”);  os\_sleep\_us(20000000, OS\_TIMEOUT\_NO\_WAKEUP); |

Next, the flag is disabled by setting the flag to 0.

|  |
| --- |
| wcm\_pm\_config(h, LISTEN\_INTERVAL\_10, TRAFFIC\_TMO, PM\_FLAGS);  print\_wifi\_config();  os\_sleep\_us(60000000, OS\_TIMEOUT\_NO\_WAKEUP); |

### Running the Application

Program wcm\_pm\_dli.elf (*sdk\_x.y\examples\wcm\_pm\bin*) using the Download tool. Refer steps in section 5.3.3 to program the ELF onto Talaria TWO.

Console output:

|  |
| --- |
| Y-BOOT 208ef13 2019-07-22 12:26:54 -0500 790da1-b-7  ROM yoda-h0-rom-16-0-gd5a8e586  FLASH:PNWWWWWAEBuild $Id: git-d198c0771 $  np\_conf\_path=/data/nprofile.json ssid=InnoPhase\_AE passphrase=Inno@1234  $App:git-df2ae28b  SDK Ver: sdk\_2.6.2  Wifi PM Dynamic listen Interval  addr e0:69:3a:00:13:90  Connecting to added network : InnoPhase\_AE  [0.790,236] CONNECT:98:da:c4:73:b7:76 Channel:10 rssi:-35 dBm  wcm\_notify\_cb to App Layer - WCM\_NOTIFY\_MSG\_LINK\_UP  wcm\_notify\_cb to App Layer - WCM\_NOTIFY\_MSG\_ADDRESS  [0.843,898] MYIP 192.168.1.104  [0.844,064] IPv6 [fe80::e269:3aff:fe00:1390]-link  wcm\_notify\_cb to App Layer - WCM\_NOTIFY\_MSG\_CONNECTED  Connected to added network : InnoPhase\_AE  Suspend enabled!  \*\*\*\*\*\*  listen\_interval: 10  traffic\_tmo: 12  sleep\_period: 1024 ms  dtim\_only disabled  tx\_ps disabled  mcast\_don't\_care disabled  rx\_nap disabled  only\_broadcast disabled  ps\_poll disabled  dli enabled  \*\*\*\*\*\*  Sleep mode for next 20 sec  \*\*\*\*\*\*  listen\_interval: 10  traffic\_tmo: 12  sleep\_period: 1024 ms  dtim\_only disabled  tx\_ps disabled  mcast\_don't\_care disabled  rx\_nap disabled  only\_broadcast disabled  ps\_poll disabled  dli disabled  \*\*\*\*\*\* |

1. Dynamic listen interval enabled

In this example, the listen interval is set to 10 and the DTIM in the AP is set to 1, so every one second the module wakes up. Since the dynamic listen interval is enabled, the module will listen to each beacon interval i.e.,100ms whenever data is available.

To test this, ping Talaria TWO IP address from the host when the application is enabled for dynamic listen interval and observe the output as shown in Figure 8.

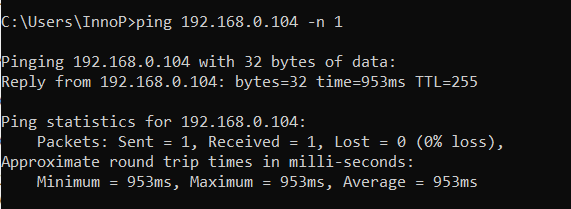


Figure : Ping output-CMD

Following is the OTTI capture of one ping traffic:

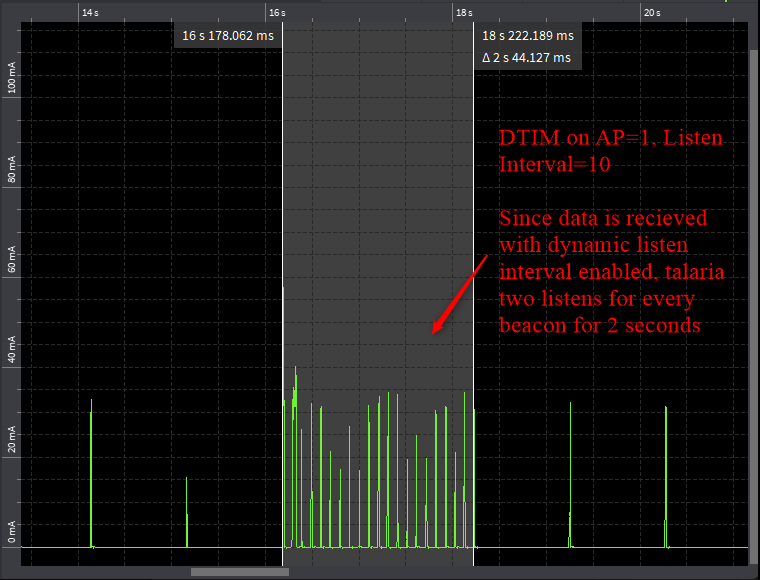


Figure : OTII capture-Dynamic listen Interval enabled

1. Dynamic Listen Interval disabled

To test this, ping Talaria TWO IP address from the host when the application is disabled for dynamic listen interval and observe the output as shown in Figure 10.

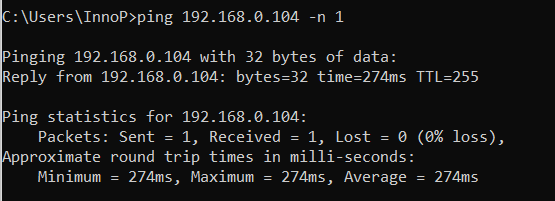


Figure : Ping output - CMD

Since dynamic listen interval is disabled, the modules listen for every 1s irrespective of any extra traffic as shown in Figure 11.

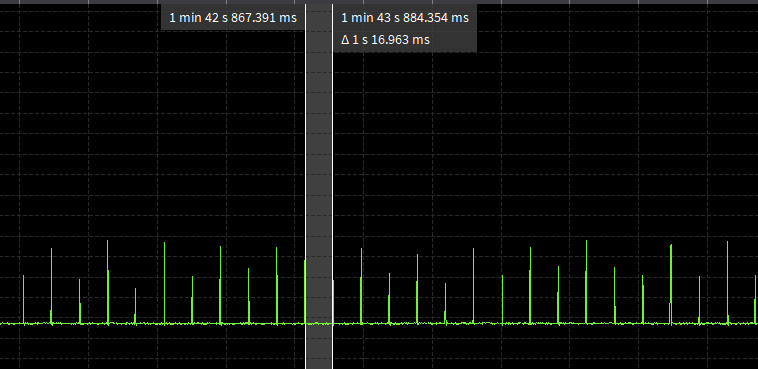


Figure : OTII capture-Dynamic listen Interval disabled

## Multicast Don’t Care (WIFI\_PM\_MCAST\_DONT\_CARE)

### Overview

When this flag is enabled, Talaria TWO will ignore the multicast bit in beacons. Use this function with care since incoming broadcast ARPs or other important broadcast/multicast traffic may be missed with this.

The sample code is located in: example/wcm\_pm/src/wcm\_pm\_mcast.c. This describes using the API wcm\_pm\_config()to enable and disable the WIFI\_PM\_MCAST\_DONT\_CARE flag.

### Sample Code Walkthrough

The following code uses wcm\_pm\_config() API to enable the multicast don’t care flag WIFI\_PM\_MCAST\_DONT\_CARE.

|  |
| --- |
| wcm\_pm\_config(h, LISTEN\_INTERVAL\_1, TRAFFIC\_TMO, WIFI\_PM\_MCAST\_DONT\_CARE);  print\_wifi\_config();  os\_sleep\_us(60000000, OS\_TIMEOUT\_NO\_WAKEUP); |

Sleep mode is enabled for 20s before disabling this flag again.

|  |
| --- |
| os\_printf("Sleep mode for next 20 sec\n");  os\_sleep\_us(20000000, OS\_TIMEOUT\_NO\_WAKEUP);  os\_printf("\n\*\*\*\*\*\*\n"); |

Next, the flag is disabled by setting the flag to 0.

|  |
| --- |
| wcm\_pm\_config(h, LISTEN\_INTERVAL\_1, TRAFFIC\_TMO, PM\_FLAGS);  print\_wifi\_config();  os\_sleep\_us(60000000, OS\_TIMEOUT\_NO\_WAKEUP); |

### Running the Application

Program wcm\_pm\_mcast.elf (*sdk\_x.y\examples\wcm\_pm\bin*) using the Download tool. Refer steps in section 5.3.3 to program the ELF onto Talaria TWO.

Console output:

|  |
| --- |
| Y-BOOT 208ef13 2019-07-22 12:26:54 -0500 790da1-b-7  ROM yoda-h0-rom-16-0-gd5a8e586  FLASH:PNWWWWWAEBuild $Id: git-d198c0771 $  np\_conf\_path=/data/nprofile.json ssid=InnoPhase\_AE passphrase=Inno@1234  $App:git-df2ae28b  SDK Ver: sdk\_2.6.2  Wifi PM Multicast Don't care  addr e0:69:3a:00:13:90  Connecting to added network : InnoPhase\_AE  [0.833,830] CONNECT:98:da:c4:73:b7:76 Channel:10 rssi:-34 dBm  wcm\_notify\_cb to App Layer - WCM\_NOTIFY\_MSG\_LINK\_UP  wcm\_notify\_cb to App Layer - WCM\_NOTIFY\_MSG\_ADDRESS  [0.874,288] MYIP 192.168.1.164  [0.874,451] IPv6 [fe80::e269:3aff:fe00:1390]-link  wcm\_notify\_cb to App Layer - WCM\_NOTIFY\_MSG\_CONNECTED  Connected to added network : InnoPhase\_AE  Suspend enabled!  \*\*\*\*\*\*  listen\_interval: 1  traffic\_tmo: 12  sleep\_period: 102 ms  dtim\_only disabled  tx\_ps disabled  mcast\_don't\_care enabled  rx\_nap disabled  only\_broadcast disabled  ps\_poll disabled  dli disabled  \*\*\*\*\*\*  Sleep mode for next 20 sec  \*\*\*\*\*\*  listen\_interval: 1  traffic\_tmo: 12  sleep\_period: 102 ms  dtim\_only disabled  tx\_ps disabled  mcast\_don't\_care disabled  rx\_nap disabled  only\_broadcast disabled  ps\_poll disabled  dli disabled  \*\*\*\*\*\* |

1. Multicast don’t care enabled

Send UDP traffic in the multicast mode once multicast don’t care is enabled. Observe the decrease in the Rx current duration from the OTII logs. To send UPD traffic, use the following link to download the app: <https://www.cistechsolutions.com/netspanner>.

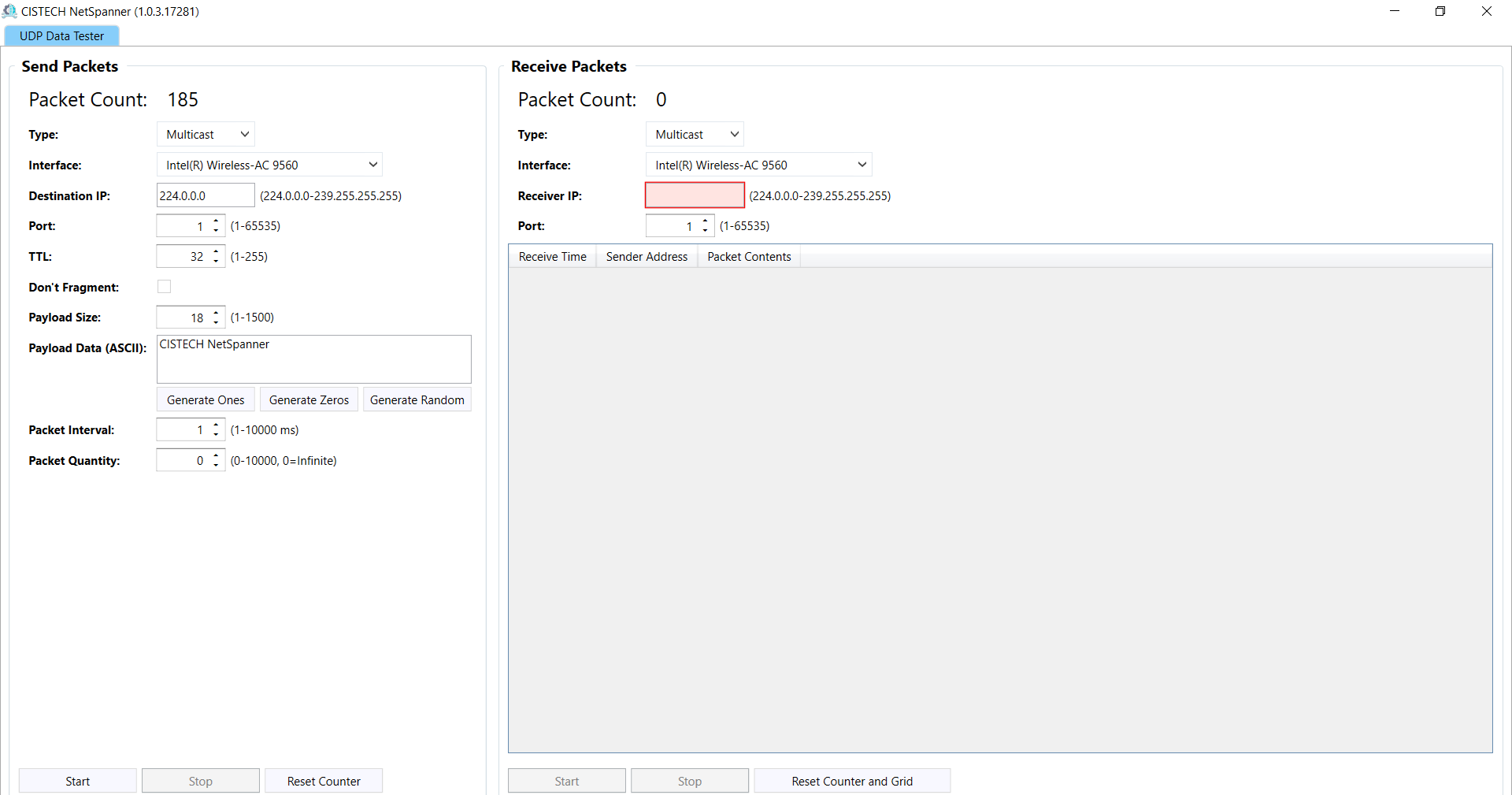


Figure : UDP multicast traffic

The following Wireshark logs show the UDP data sent in multicast mode. However, Talaria TWO ignores this traffic because the multicast don’t care is enabled.

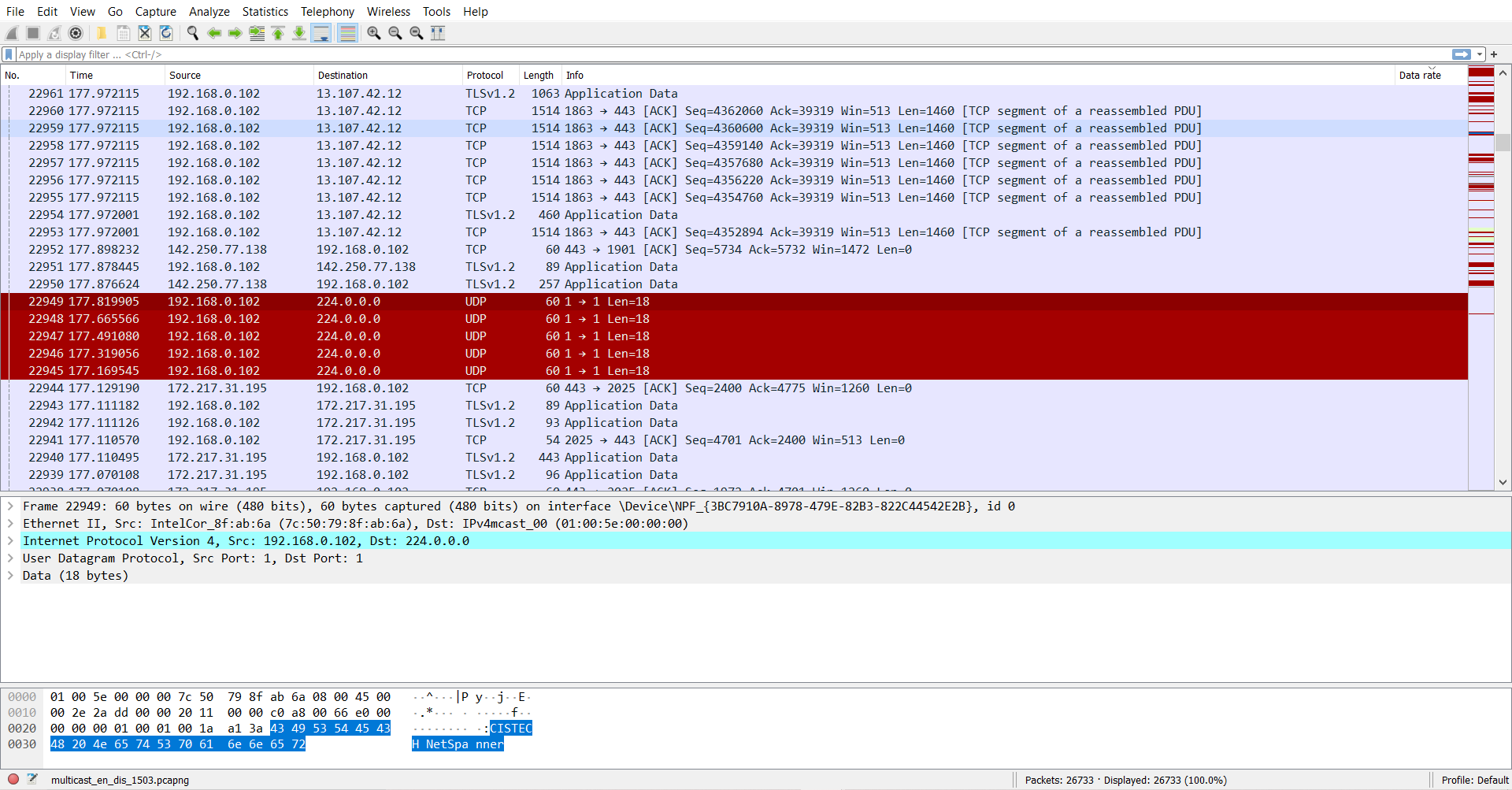


Figure : Wireshark capture-Multicast don't care enabled

Since the Multicast don't care is enabled, the Rx current duration is approximately 2ms as shown in the OTTI logs in Figure 14.

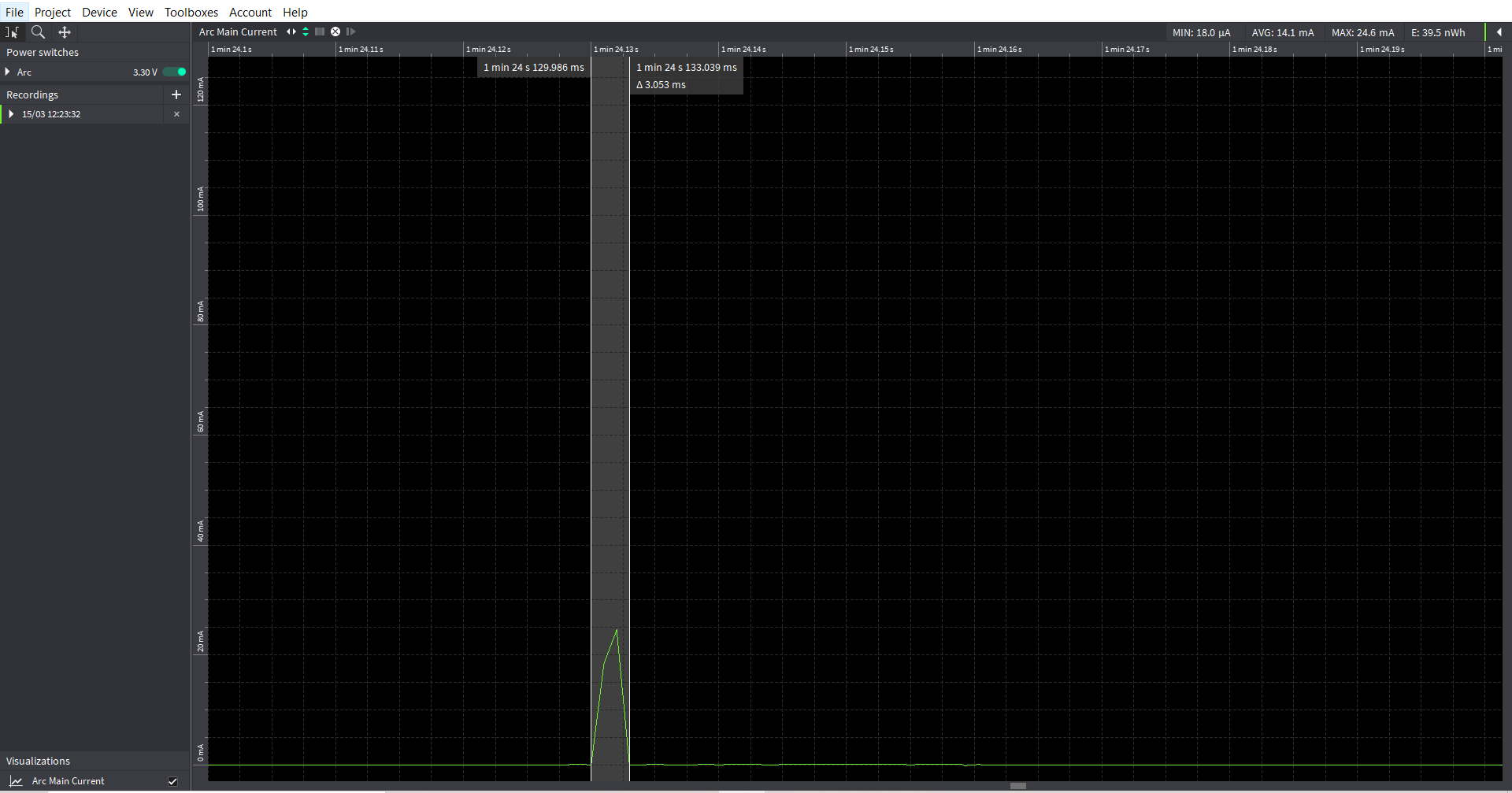


Figure : Multicast don't care enabled-OTII log

1. Multicast don’t care disabled

Send UDP traffic in the multicast mode once multicast don’t care is disabled. Observe increase in the Rx current duration from the OTII logs shown in Figure 15.

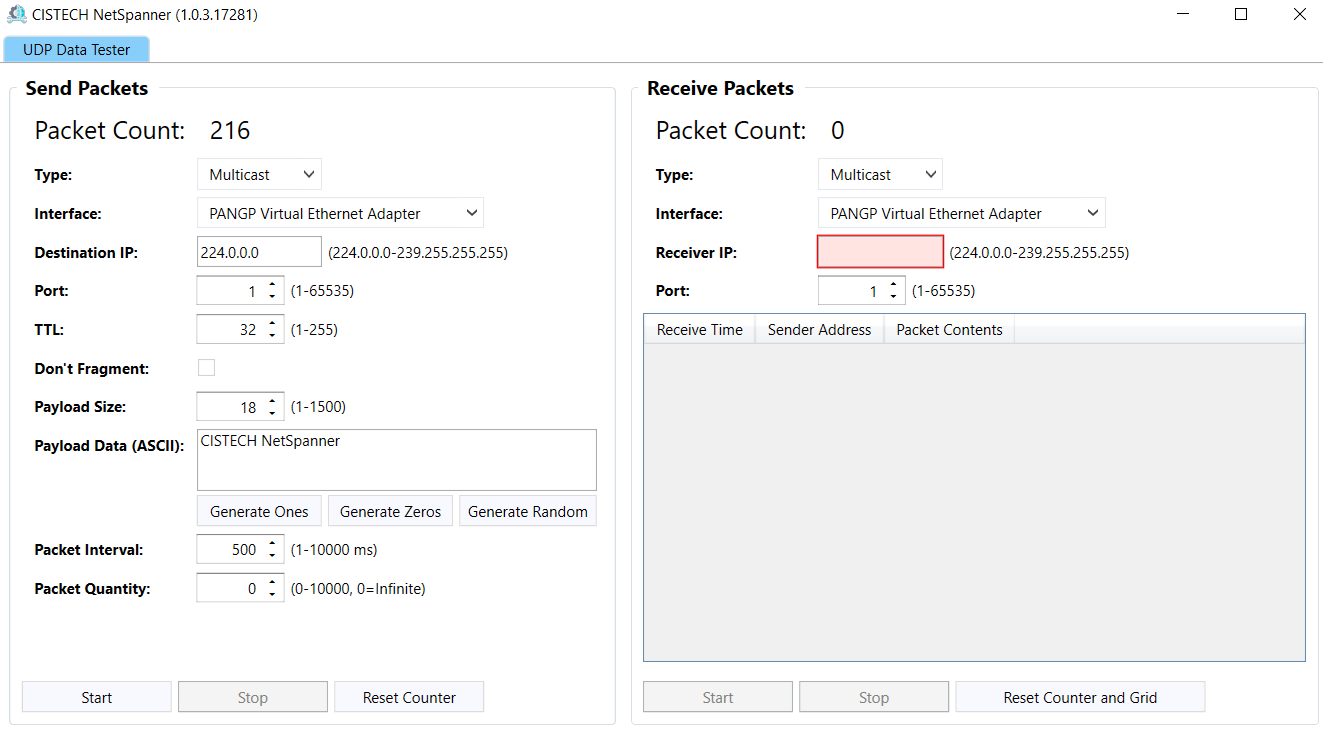


Figure : UDP musticast traffic

The following Wireshark logs show the UDP data sent in multicast mode. Talaria TWO receives this traffic because the multicast don’t care is disabled.

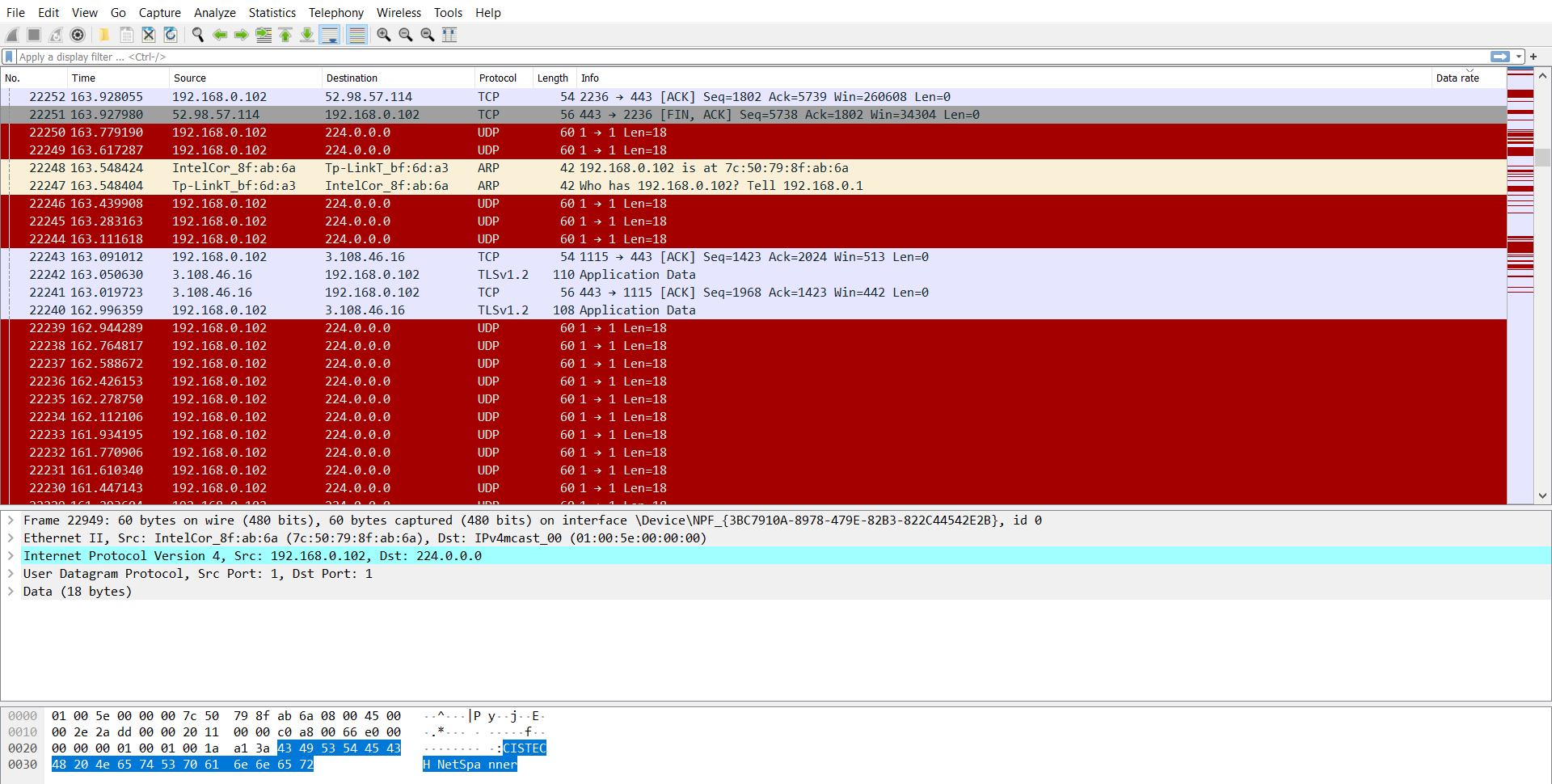


Figure : Wireshark capture-Multicast don't care disabled

Since the Multicast don't care is disabled, Rx current duration is approximately 9.213ms.

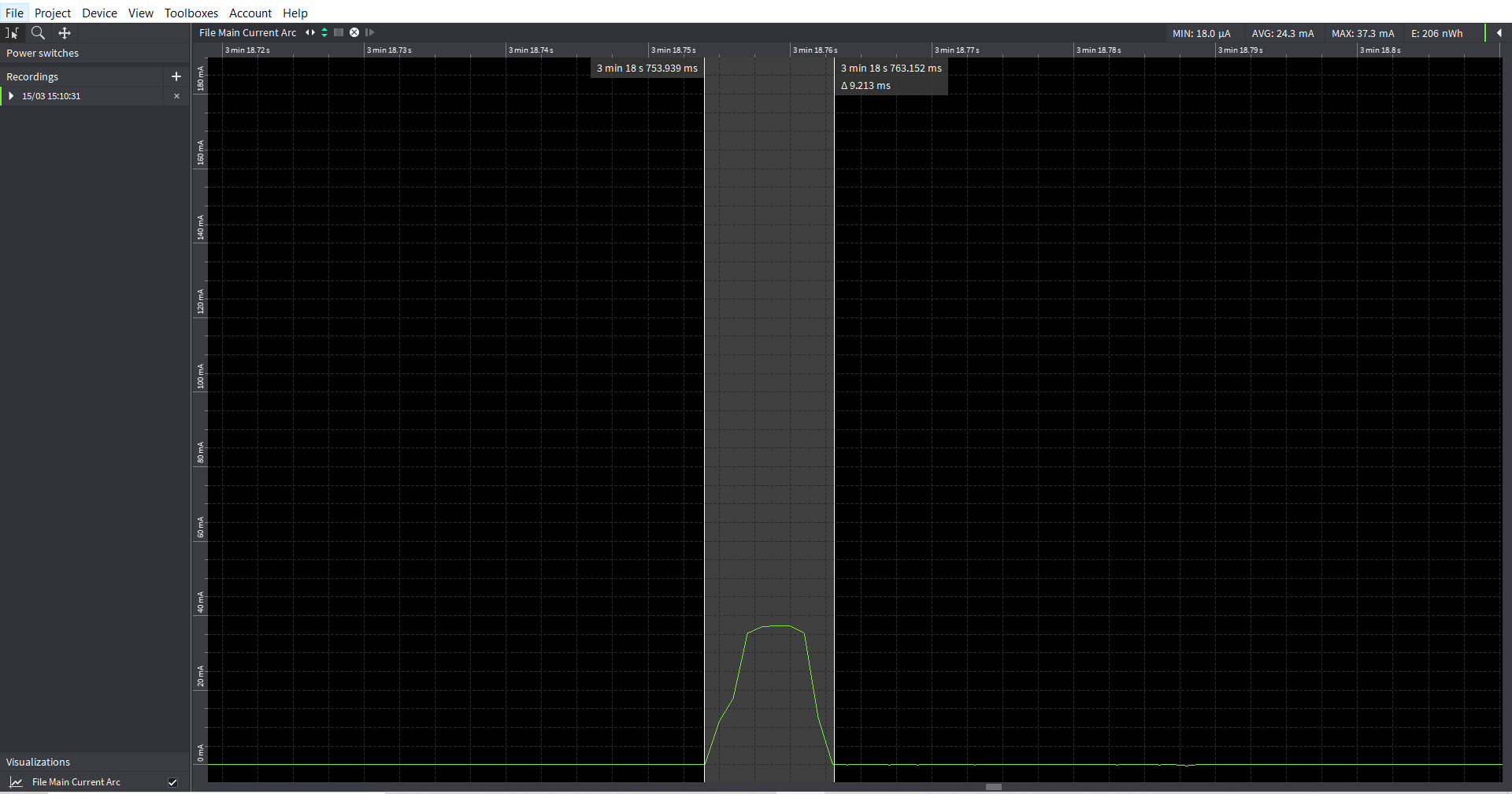


Figure : Multicast don't care disabled-OTII log

## Power Save Poll (WIFI\_PM\_PS\_POLL)

### Overview

PS-Poll (power save poll) can be used to poll (and receive) unicast data from the AP while in power save.

If Talaria TWO does not receive the beacon at TBTT (TU equal to 1024µs) then a PS-Poll is sent after the beacon window. The window size of the PS-Poll should be around 8ms.

After that, Talaria TWO waits for an ACK or a timeout on the ACK reception. Then, irrespective of whether there is ACK or not Talaria TWO listens for up to 8ms for potential unicast data from the AP. If unicast data is received, the reception will continue as long as needed.

### Sample Code Walkthrough

The following code uses wcm\_pm\_config() API to enable the power save poll flag WIFI\_PM\_PS\_POLL.

|  |
| --- |
| wcm\_pm\_config(h, LISTEN\_INTERVAL\_1, TRAFFIC\_TMO, WIFI\_PM\_PS\_POLL);  print\_wifi\_config();  os\_sleep\_us(60000000, OS\_TIMEOUT\_NO\_WAKEUP); |

Next, the flag is disabled by setting the flag to 0.

|  |
| --- |
| wcm\_pm\_config(h, LISTEN\_INTERVAL\_1, TRAFFIC\_TMO, PM\_FLAGS);  print\_wifi\_config();  os\_sleep\_us(60000000, OS\_TIMEOUT\_NO\_WAKEUP); |

### Running the Application

Program wcm\_pm\_ps.elf (*sdk\_x.y\examples\wcm\_pm\bin*) using the Download tool. Refer steps in section 5.1.3 to program the ELF onto Talaria TWO.

In the GUI window, Boot Arguments section, pass the following boot arguments:

|  |
| --- |
| wifi.log\_beacon=1 |

|  |
| --- |
| Y-BOOT 208ef13 2019-07-22 12:26:54 -0500 790da1-b-7  ROM yoda-h0-rom-16-0-gd5a8e586  FLASH:PNWWWWWAEBuild $Id: git-d198c0771 $  wifi.log\_beacon=1 np\_conf\_path=/data/nprofile.json ssid=InnoPhase\_AE passphrase=Inno@1234  $App:git-df2ae28b  SDK Ver: sdk\_2.6.2  Wifi PM Power save  addr e0:69:3a:00:13:90  Connecting to added network : InnoPhase\_AE  [0.833,830] CONNECT:98:da:c4:73:b7:76 Channel:10 rssi:-34 dBm  wcm\_notify\_cb to App Layer - WCM\_NOTIFY\_MSG\_LINK\_UP  wcm\_notify\_cb to App Layer - WCM\_NOTIFY\_MSG\_ADDRESS  [0.874,288] MYIP 192.168.1.164  [0.874,451] IPv6 [fe80::e269:3aff:fe00:1390]-link  wcm\_notify\_cb to App Layer - WCM\_NOTIFY\_MSG\_CONNECTED  Connected to added network : InnoPhase\_AE  Suspend enabled!  [9.603,954] Beacon misses: 1  [9.698,970] Beacon received again.  [15.133,654] Beacon misses: 1  [15.231,082] Beacon received again.  \*\*\*\*\*\*  listen\_interval: 1  traffic\_tmo: 12  sleep\_period: 102 ms  dtim\_only disabled  tx\_ps disabled  mcast\_don't\_care disabled  rx\_nap disabled  only\_broadcast disabled  ps\_poll enabled  dli disabled  \*\*\*\*\*\*  [28.445,889] Beacon misses: 1  [28.445,989] Sending pspoll (ntries 1)  [28.540,938] Beacon received again.  [30.289,121] Beacon misses: 1  [30.289,183] Sending pspoll (ntries 1)  [30.292,002] Beacon received again.  [33.361,172] Beacon misses: 1  [33.361,236] Sending pspoll (ntries 1)  [33.456,236] Beacon received again.  [41.450,910] Beacon misses: 1  [41.450,972] Sending pspoll (ntries 1)  [41.545,952] Beacon received again.  [48.721,440] Beacon misses: 1  [48.721,502] Sending pspoll (ntries 1)  [48.816,470] Beacon received again.  [61.112,059] Beacon misses: 1  [61.112,121] Sending pspoll (ntries 1)  [61.207,103] Beacon received again.  [74.629,085] Beacon misses: 1  [74.629,147] Sending pspoll (ntries 1)  [74.725,949] Beacon received again.  \*\*\*\*\*\*  listen\_interval: 1  traffic\_tmo: 12  sleep\_period: 102 ms  dtim\_only disabled  tx\_ps disabled  mcast\_don't\_care disabled  rx\_nap disabled  only\_broadcast disabled  ps\_poll disabled  dli disabled  \*\*\*\*\*\* |

1. Power save mode enabled

ps\_poll is sent when a beacon is missed as shown in Figure 18.

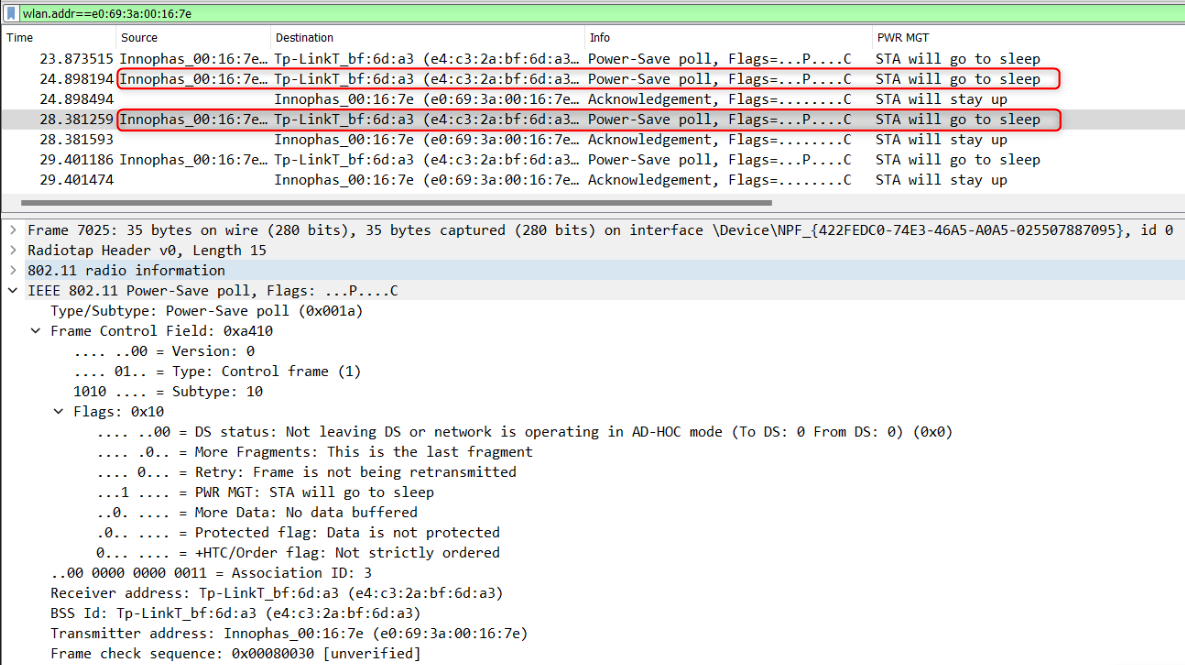


Figure : Wireshark capture-ps\_poll enabled

ps\_poll spike is observed. Here, the transmitting current window size is 7.605ms as shown in Figure 19.

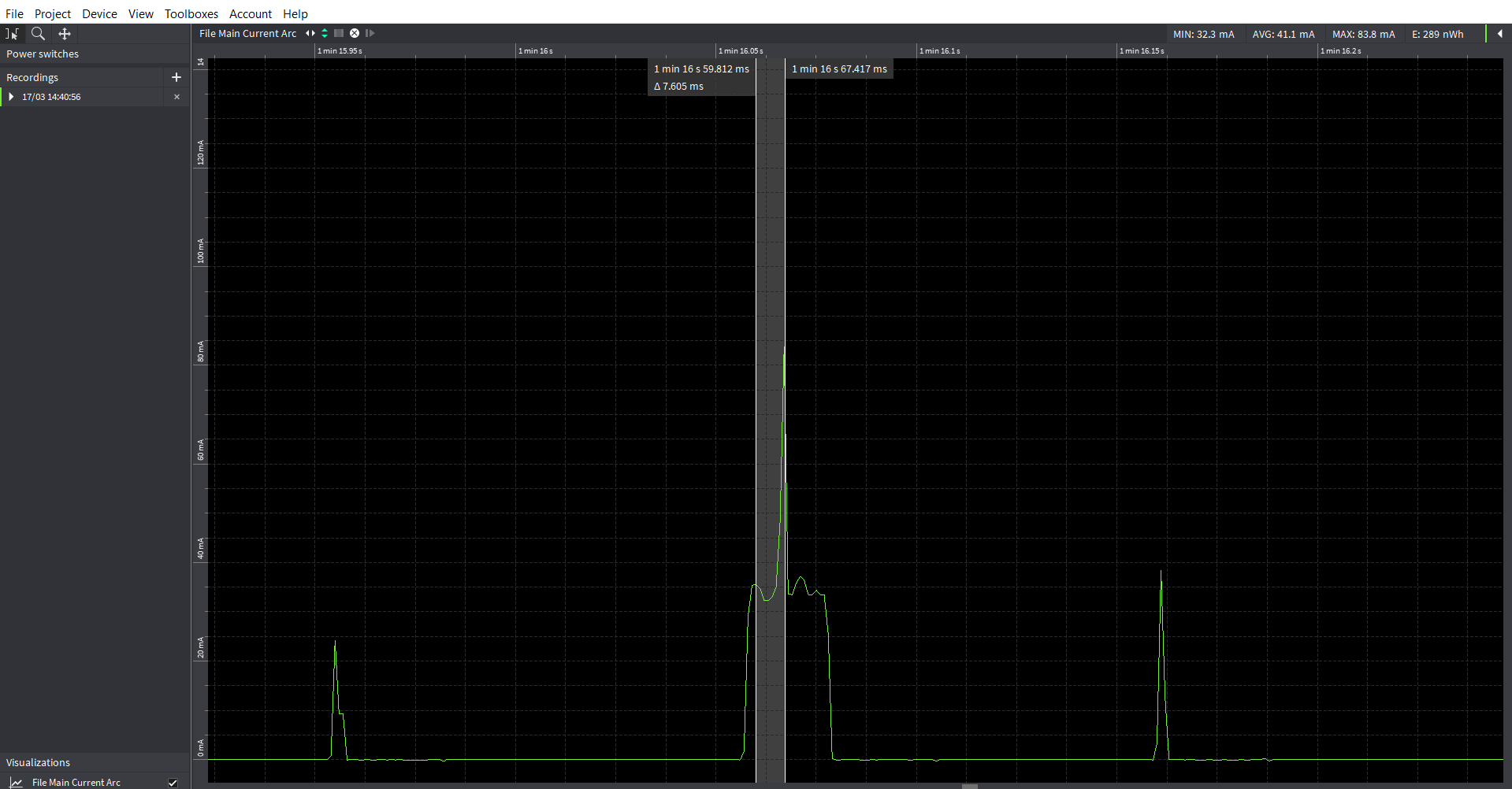


Figure : OTII logs- Power save poll enabled

1. Power-save mode disabled

There are no ps-polls in the Wireshark capture in Figure 20 in case of beacon miss since the ps-poll is disabled.

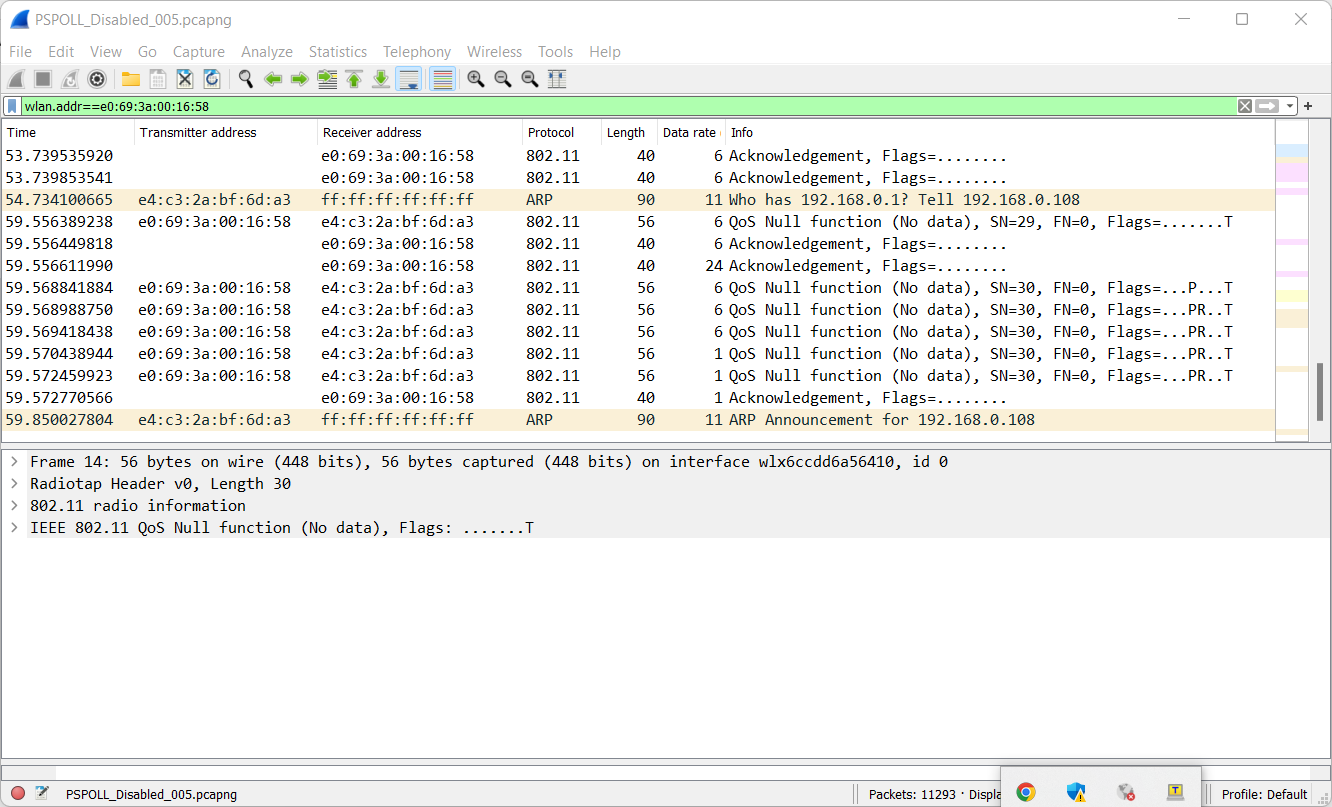


Figure : Wireshark capture-ps\_poll disabled

No ps\_poll is sent. Hence, Tx current spike is not observed as shown in Figure 21.

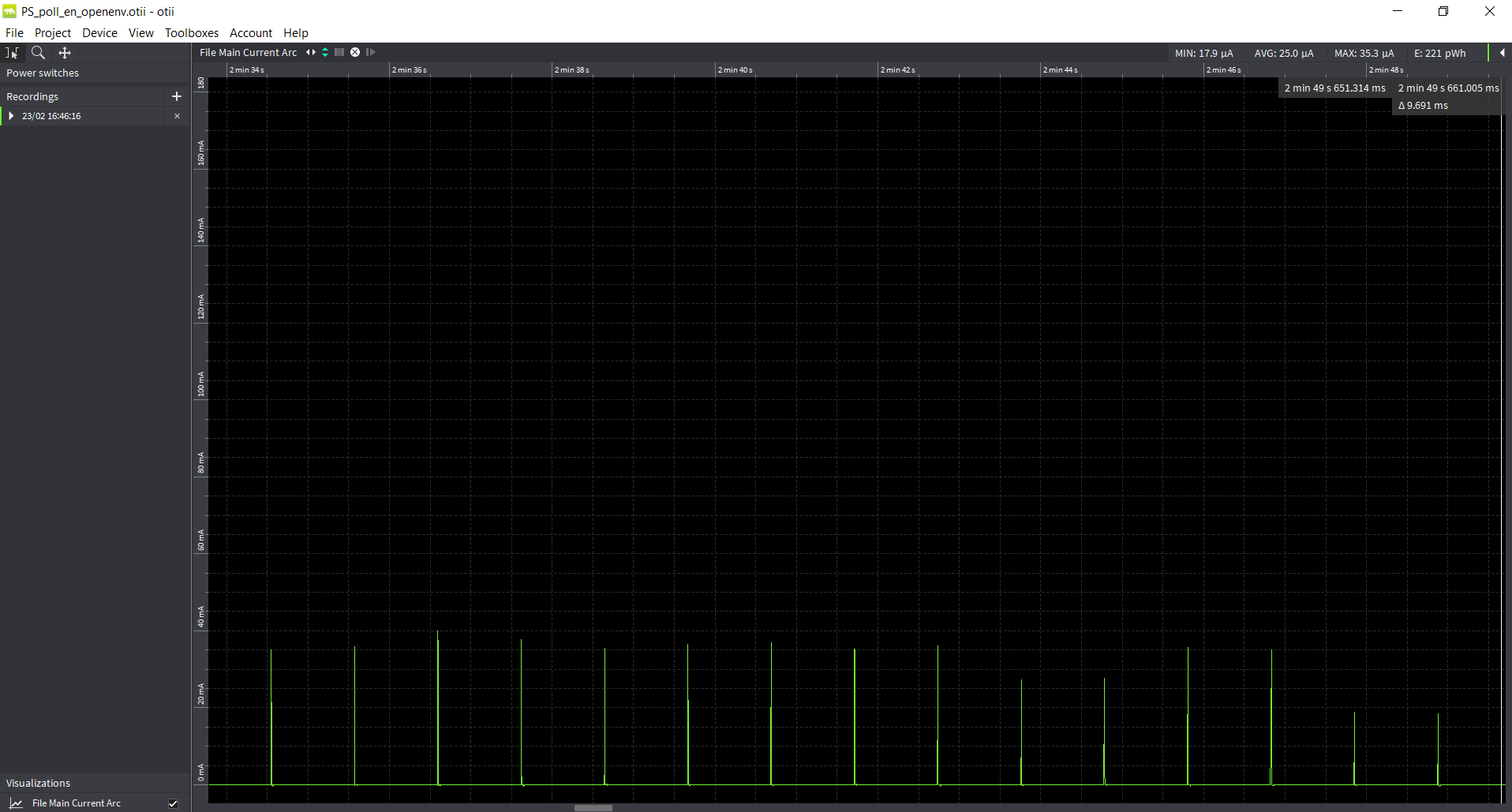


Figure : OTII logs- Power save poll disabled

## Transmit Power Save (WIFI\_PM\_TX\_PS)

### Overview

Send outgoing frames without leaving Wi-Fi power save (when possible). This function will save power for sensor-type applications where the response from the server will be received after the traffic timeout. The response from the server will instead be fetched after the next beacon is received. Some power is also saved when sending gratuitous ARPs or ICMP messages.

The sample code located in the path example/wcm\_pm/src/wcm\_pm\_tx\_ps.c. This describes using the API wcm\_pm\_config()to enable and disable the WIFI\_PM\_TX\_PS flag.

### Sample Code Walkthrough

The following code uses wcm\_pm\_config() API to enable the transmit power save flag WIFI\_PM\_TX\_PS.

Gratuitous APR is enabled using the API wcm\_arp\_grat\_enable().

|  |
| --- |
| os\_printf("Gratuitous ARP enabled!\n");  wcm\_arp\_grat\_enable(h,20);  wcm\_pm\_config(h, LISTEN\_INTERVAL\_10, TRAFFIC\_TMO, WIFI\_PM\_TX\_PS);  print\_wifi\_config();  os\_sleep\_us(60000000, OS\_TIMEOUT\_NO\_WAKEUP); |

Sleep mode is enabled for 20s before disabling this flag again.

|  |
| --- |
| os\_printf("Sleep mode for next 20 sec\n");  os\_sleep\_us(20000000, OS\_TIMEOUT\_NO\_WAKEUP);  os\_printf("\n\*\*\*\*\*\*\n"); |

Next, the flag is disabled by setting the flag to 0.

|  |
| --- |
| wcm\_pm\_config(h, LISTEN\_INTERVAL\_10, TRAFFIC\_TMO, PM\_FLAGS);  print\_wifi\_config();  os\_sleep\_us(60000000, OS\_TIMEOUT\_NO\_WAKEUP); |

### Running the Application

Program wcm\_pm\_tx\_ps.elf (*sdk\_x.y\examples\wcm\_pm\bin*) using the Download tool. Refer steps in section 5.3.3 to program the ELF onto Talaria TWO.

Console output:

|  |
| --- |
| Y-BOOT 208ef13 2019-07-22 12:26:54 -0500 790da1-b-7  ROM yoda-h0-rom-16-0-gd5a8e586  FLASH:PNWWWWWAEBuild $Id: git-d198c0771 $  np\_conf\_path=/data/nprofile.json ssid=InnoPhase\_AE passphrase=Inno@1234  $App:git-df2ae28b  SDK Ver: sdk\_2.6.2  Wifi PM Demo Transmit PS  addr e0:69:3a:00:13:90  Connecting to added network : InnoPhase\_AE  [0.807,203] CONNECT:98:da:c4:73:b7:76 Channel:10 rssi:-34 dBm  wcm\_notify\_cb to App Layer - WCM\_NOTIFY\_MSG\_ADDRESS  wcm\_notify\_cb to App Layer - WCM\_NOTIFY\_MSG\_LINK\_UP  [0.842,999] MYIP 192.168.1.164  [0.843,163] IPv6 [fe80::e269:3aff:fe00:1390]-link  wcm\_notify\_cb to App Layer - WCM\_NOTIFY\_MSG\_CONNECTED  Connected to added network : InnoPhase\_AE  Suspend enabled!  Gratuitous ARP enabled!  \*\*\*\*\*\*  listen\_interval: 10  traffic\_tmo: 12  sleep\_period: 1024 ms  dtim\_only disabled  tx\_ps enabled  mcast\_don't\_care disabled  rx\_nap disabled  only\_broadcast disabled  ps\_poll disabled  dli disabled  \*\*\*\*\*\*  Sleep mode for next 20 sec  \*\*\*\*\*\*  listen\_interval: 10  traffic\_tmo: 12  sleep\_period: 1024 ms  dtim\_only disabled  tx\_ps disabled  mcast\_don't\_care disabled  rx\_nap disabled  only\_broadcast disabled  ps\_poll disabled  dli disabled  \*\*\*\*\*\* |

1. Transmit power save enabled

There is no null frame before and after the Gratuitous ARP, since the transmit power save is enabled.

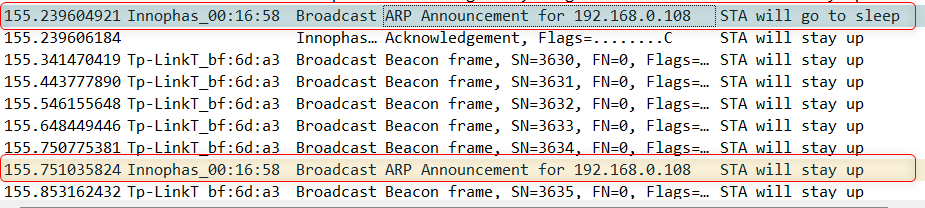


Figure : Wireshark capture- Transmit power save

1. Transmit power save disabled

There is a null frame before and after the ARP the Gratuitous ARP.

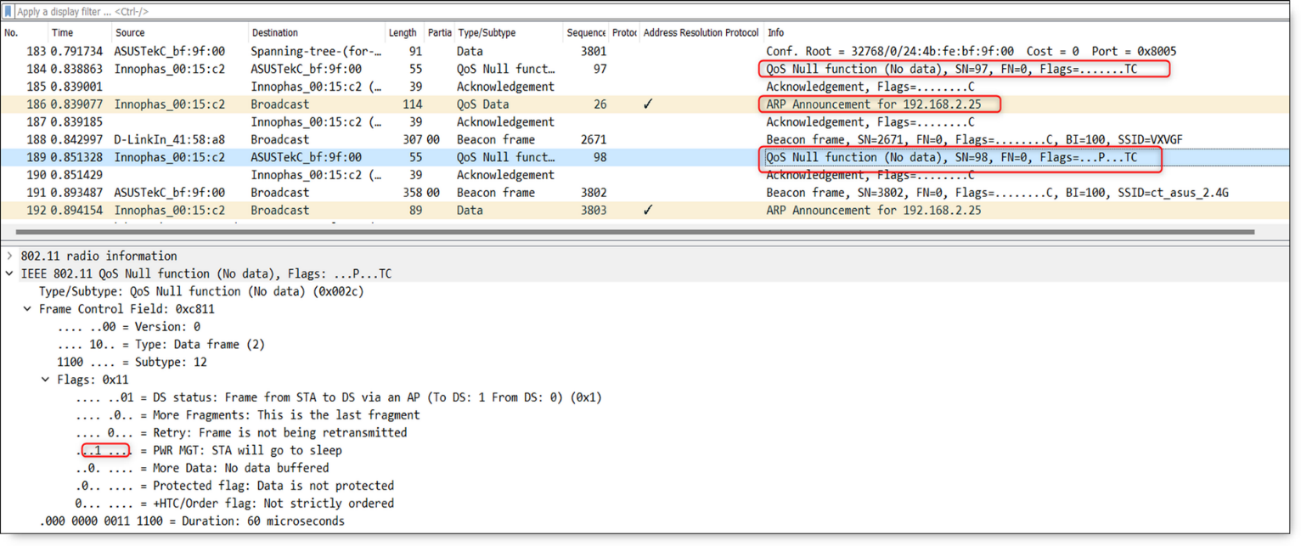


Figure : Wireshark capture- Transmit power save disabled

## Traffic Timeout (traffic\_tmo)

### Overview

The traffic\_tmo parameter can be set using wcm\_pm\_config() API. Refer section 5.1.1 point 2 for definition of this feature.

Traffic timeout value can be increased for the applications that require decreased latency such as firmware upgrade. However, this will increase the power consumption.

For line powered applications traffic timeout can be set to 0 where Wi-Fi is always turned on and power save will not be used.

The sample code is in: *example/wcm\_pm/src/wcm\_pm\_tmo.c.*

### Sample Code Walkthrough

The following code uses wcm\_pm\_config() API to enable the traffic\_tmo parameter. Suspend mode is enabled using the API os\_suspend\_enable().

|  |
| --- |
| os\_printf("Suspend enabled!\n");  os\_suspend\_enable(); |

Sets the traffic timeout to 0 and listen interval to 10. Sleep mode is enabled for 10s before changing this parameter value.

|  |
| --- |
| wcm\_pm\_config(h, LISTEN\_INTERVAL\_10, TRAFFIC\_TMO\_0, PM\_FLAGS);  print\_wifi\_config();  os\_sleep\_us(10000000, OS\_TIMEOUT\_NO\_WAKEUP); |

Next, sets the traffic timeout to 12ms and enables sleep mode to 20 seconds.

|  |
| --- |
| wcm\_pm\_config(h, LISTEN\_INTERVAL\_10, TRAFFIC\_TMO\_12, PM\_FLAGS);  print\_wifi\_config();  os\_sleep\_us(60000000, OS\_TIMEOUT\_NO\_WAKEUP); |

### Running the Application

Program wcm\_pm\_tmo.elf (*sdk\_x.y\examples\wcm\_pm\bin*) using the Download tool. Refer steps in section 5.3.3 to program the ELF onto Talaria TWO.

Console output:

|  |
| --- |
| Y-BOOT 208ef13 2019-07-22 12:26:54 -0500 790da1-b-7  ROM yoda-h0-rom-16-0-gd5a8e586  FLASH:PNWWWWWAEBuild $Id: git-d198c0771 $  np\_conf\_path=/data/nprofile.json ssid=InnoPhase\_AE passphrase=Inno@1234  $App:git-df2ae28b  SDK Ver: sdk\_2.6.2  Wifi PM Traffic Timeout Demo App  addr e0:69:3a:00:13:90  Connecting to added network : InnoPhase\_AE  [0.825,019] CONNECT:98:da:c4:73:b7:76 Channel:10 rssi:-34 dBm  wcm\_notify\_cb to App Layer - WCM\_NOTIFY\_MSG\_LINK\_UP  wcm\_notify\_cb to App Layer - WCM\_NOTIFY\_MSG\_ADDRESS  [2.688,072] MYIP 192.168.1.164  [2.688,121] IPv6 [fe80::e269:3aff:fe00:1390]-link  wcm\_notify\_cb to App Layer - WCM\_NOTIFY\_MSG\_CONNECTED  Connected to added network : InnoPhase\_AE  Suspend enabled!  \*\*\*\*\*\*  listen\_interval: 10  traffic\_tmo: 0  sleep\_period: 102 ms  dtim\_only disabled  tx\_ps disabled  mcast\_don't\_care disabled  rx\_nap disabled  only\_broadcast disabled  ps\_poll disabled  dli disabled  \*\*\*\*\*\*  Sleep mode for next 20 sec  \*\*\*\*\*\*  listen\_interval: 10  traffic\_tmo: 12  sleep\_period: 1024 ms  dtim\_only disabled  tx\_ps disabled  mcast\_don't\_care disabled  rx\_nap disabled  only\_broadcast disabled  ps\_poll disabled  dli disabled  \*\*\*\*\*\* |

1. Traffic timeout set to 0 and 12ms

Talaria TWO stays in active state when Traffic timeout is set to 0 and enters suspend state when suspend mode gets enabled with listen interval configured to 10 and traffic timeout to 12ms.

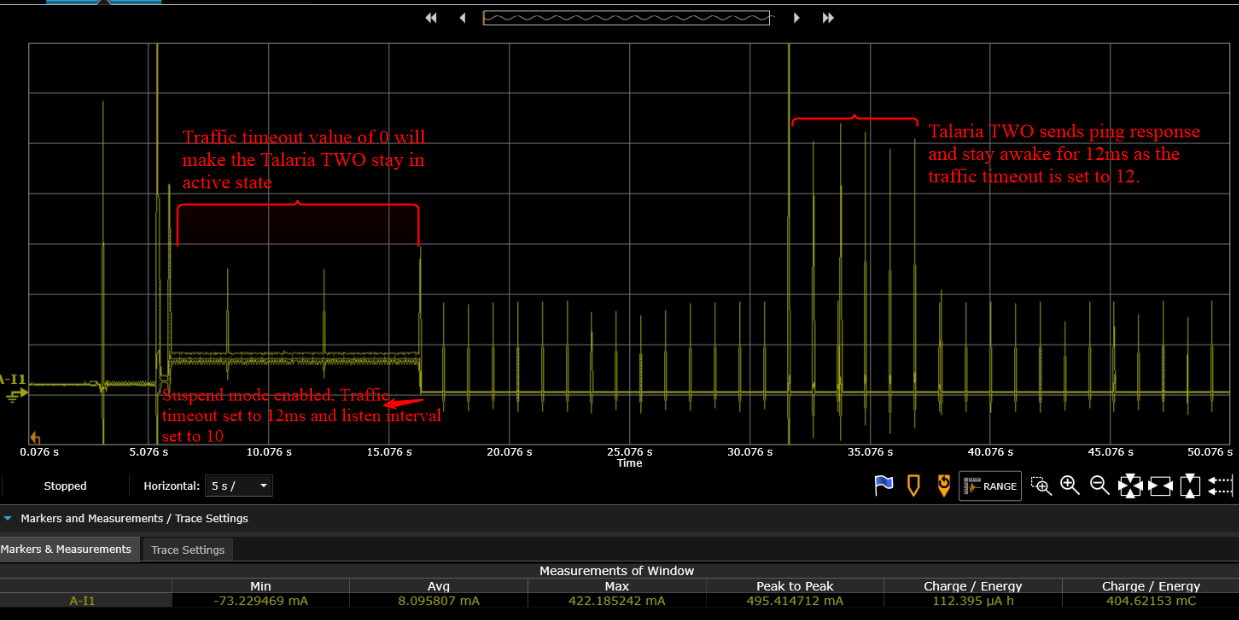


Figure : Power Analyzer- Traffic timeout switched from 0 to 12ms

To test this, ping from host machine to Talaria TWO to observe device staying in awake state for 12ms after incoming and outgoing traffic.



Figure : Power Analyzer- Talaria TWO stays awake for 12ms after traffic

## DTIM Only (WIFI\_PM\_DTIM\_ONLY)

### Overview

When enabled, Talaria TWO wakes up only at effective listen interval and does not switch to listen to every beacon in case of beacon miss. This feature can be enabled to save power if the application is not latency sensitive.

The sample code is in: *examples/wcm\_pm/src/wcm\_pm\_dtim\_only.c*.

### Sample Code Walkthrough

The following code uses wcm\_pm\_config() API to enable the WIFI\_PM\_DTIM\_ONLY flag . Suspend mode is enabled using the API os\_suspend\_enable().

|  |
| --- |
| wifi\_main();  os\_printf("Suspend enabled!\n");  os\_suspend\_enable(); |

Sets the DTIM ONLY flag, prints the power management configurations and enables sleep mode for 20 seconds.

|  |
| --- |
| /\* Enable DTIM switching by disabling the DTIM\_ONLY power save flag \*/  wcm\_pm\_config(h, LISTEN\_INTERVAL\_10, TRAFFIC\_TMO, PM\_FLAGS);  print\_wifi\_config(); |

Disables the DTIM ONLY flag and prints the power management configurations.

|  |
| --- |
| /\* Disable DTIM switching by enabling the DTIM\_ONLY power save flag \*/  wcm\_pm\_config(h, LISTEN\_INTERVAL\_10, TRAFFIC\_TMO, WIFI\_PM\_DTIM\_ONLY);  print\_wifi\_config(); |

### Running the Application

Program wcm\_pm\_dtim\_only.elf (*sdk\_x.y\examples\wcm\_pm\bin*) using the Download tool. Refer steps in section 5.3.3 to program the ELF onto Talaria TWO.

Console output:

|  |
| --- |
| Y-BOOT 208ef13 2019-07-22 12:26:54 -0500 790da1-b-7  ROM yoda-h0-rom-16-0-gd5a8e586  FLASH:PNWWWWWAEBuild $Id: git-d198c0771 $  np\_conf\_path=/data/nprofile.json ssid=InnoPhase\_AE passphrase=Inno@1234  $App:git-df2ae28b  SDK Ver: sdk\_2.6.2  Wifi PM Demo DTIM\_Only  addr e0:69:3a:00:13:90  Connecting to added network : InnoPhase\_AE  [1.041,472] CONNECT:98:da:c4:73:b7:76 Channel:10 rssi:-35 dBm  wcm\_notify\_cb to App Layer - WCM\_NOTIFY\_MSG\_LINK\_UP  wcm\_notify\_cb to App Layer - WCM\_NOTIFY\_MSG\_ADDRESS  [1.130,188] MYIP 192.168.1.164  [1.130,354] IPv6 [fe80::e269:3aff:fe00:1390]-link  wcm\_notify\_cb to App Layer - WCM\_NOTIFY\_MSG\_CONNECTED  Connected to added network : InnoPhase\_AE  Suspend enabled!  \*\*\*\*\*\*  listen\_interval: 10  traffic\_tmo: 12  sleep\_period: 102 ms  dtim\_only disabled  tx\_ps disabled  mcast\_don't\_care disabled  rx\_nap disabled  only\_broadcast disabled  ps\_poll disabled  dli disabled  \*\*\*\*\*\*  Sleep mode for next 20 sec  \*\*\*\*\*\*  listen\_interval: 10  traffic\_tmo: 12  sleep\_period: 1024 ms  dtim\_only enabled  tx\_ps disabled  mcast\_don't\_care disabled  rx\_nap disabled  only\_broadcast disabled  ps\_poll disabled  dli disabled  \*\*\*\*\*\* |

1. DTIM ONLY mode disabled.

Talaria TWO listens to every 10th beacon since listen interval is set to 10 and DTIM on AP=1. When there is a beacon miss, device switches to DTIM=1.

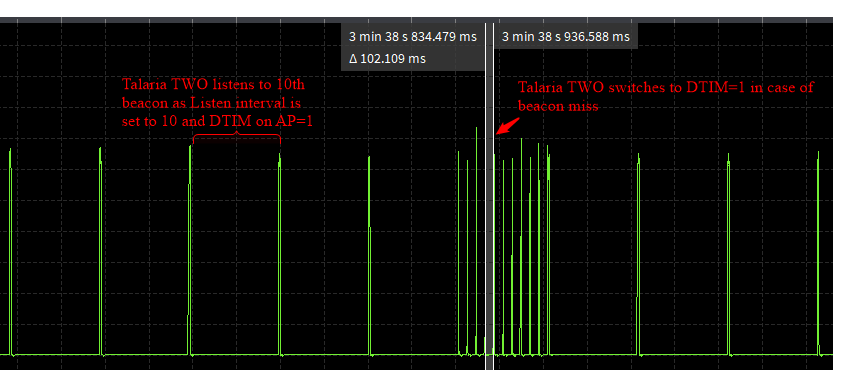


Figure : DTIM ONLY mode disabled

1. DTIM ONLY mode enabled.

Talaria TWO listens to 10th beacon as Listen interval is set to 10 and DTIM on AP=1. No DTIM switching occurs in case of beacon miss.

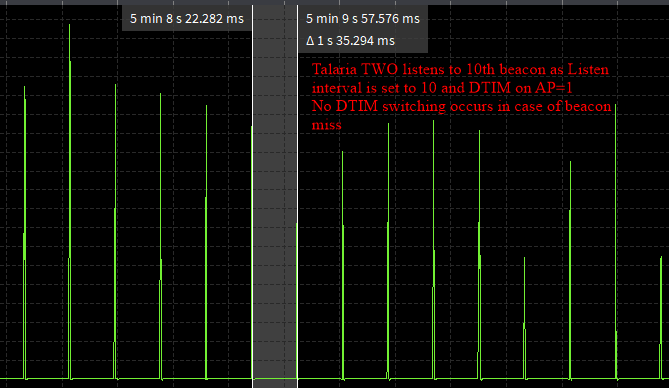


Figure : DTIM ONLY mode enabled

# Power Management Boot Arguments

## Cloud Power Mode

### Overview

When there is no internal LAN traffic needed and all communication goes through the gateway, for instance to the cloud, it is possible to reduce the power consumption significantly with the cloud power mode.

In Cloud Power Mode, Talaria TWO can turn off the multicast/ broadcast reception and thereby reduce the power consumption. In this mode, “multicast don’t care” feature will be enabled by default.

The Access Point’s ARP cache is updated by sending ICMP packet with Time to Live (TTL) set to 1, so the access point need not send broadcast ARPs to Talaria TWO. This ICMP method helps in maintaining connection and save power.

Cloud Power Mode feature can be used in a low power device such as sensor which is not sensitive to data latency.

### Boot Arguments

To enable the cloud power mode, use the following boot argument:

|  |
| --- |
| wifi.cloud\_pm=1 |

### ARP Cache Update on Access Point

1. Send a ping, to for instance 1.1.1.1 (google DNS), but set the TTL (Time To Live) field in the IP header to 1.
2. When this packet passes the first router, the TTL field will be decremented to zero, and the frame will be discarded. But this router must now send an ICMP TTL expired message back to the source (Talaria TWO).
3. When the router sends the TTL expired message it needs to have a valid ARP cache entry for Talaria TWO and the ARP cache timeout should be extended in the router.

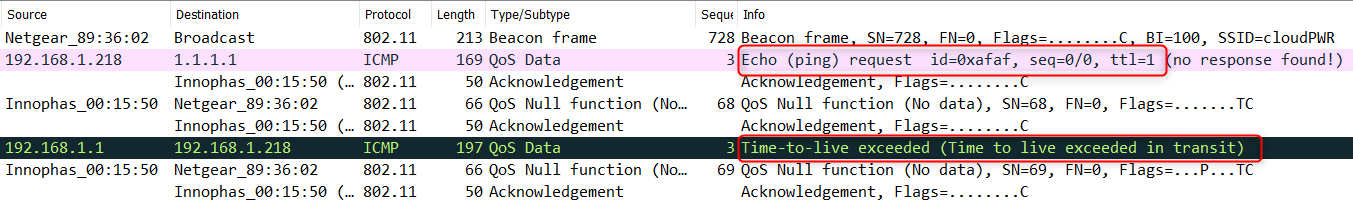


Figure : Wireshark sniffer- ICMP packet exchange

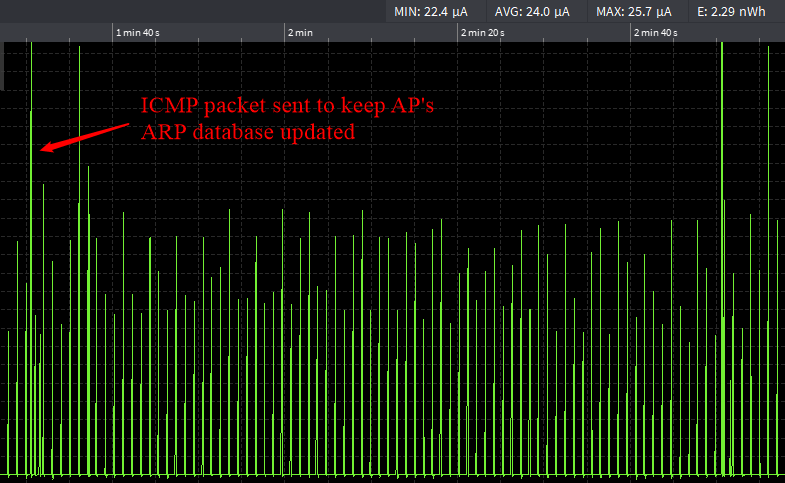


Figure : OTII logs- cloud power mode enabled

## Wake Time After Keepalive

### Overview

Talaria TWO sends periodic keep alive NULL frames to the Access Point for the duration indicated in the Max Idle Period (in seconds) boot argument. The keep\_alive\_wake\_time boot argument is the time when device remains awake after the ack of the NULL frame (in milliseconds).

There could be instances when AP just silently disconnects the station. In such scenario, when Talaria TWO sends a keep-alive frame (NULL frame + ack), the AP will send a de-authentication frame after the ack. In this de-authentication method, it will be stated that the device is no longer associated. Hence, the purpose of keep\_alive\_wake\_time is just to be able to receive the de-authentication message so that Talaria TWO can reconnect.

### Boot Arguments

The default value is set to 8ms.

|  |
| --- |
| wifi.keep\_alive\_wake\_time=<int> |

### Running the Application

Program wifi\_connect.elf (*sdk\_x.y\examples\using\_wifi\bin*) using the Download tool. Refer steps in section 5.1.3 to program the ELF onto Talaria TWO.

In the GUI window, Boot Arguments section, pass the following boot arguments:

|  |
| --- |
| suspend=1,wifi.max\_idle\_period=5 |

Console output:

|  |
| --- |
| UART:SNWWWWAE  Build $Id: git-d198c0771 $  hio.baudrate=921600  flash: Gordon ready!  Y-BOOT 208ef13 2019-07-22 12:26:54 -0500 790da1-b-7  ROM yoda-h0-rom-16-0-gd5a8e586  FLASH:PNWWWWWAEBuild $Id: git-d198c0771 $  suspend=1 wifi.max\_idle\_period=5 np\_conf\_path=/data/nprofile.json ssid=InnoPhase\_AE passphrase=Inno@1234  $App:git-df2ae28b  SDK Ver: sdk\_2.6.2  Wifi Connect Demo App  addr e0:69:3a:00:13:90  host name talaria2  Connecting to added network : InnoPhase\_AE  [0.802,362] CONNECT:98:da:c4:73:b7:76 Channel:10 rssi:-33 dBm  wcm\_notify\_cb to App Layer - WCM\_NOTIFY\_MSG\_LINK\_UP  wcm\_notify\_cb to App Layer - WCM\_NOTIFY\_MSG\_ADDRESS  [0.855,973] MYIP 192.168.1.164  [0.856,253] IPv6 [fe80::e269:3aff:fe00:1390]-link  wcm\_notify\_cb to App Layer - WCM\_NOTIFY\_MSG\_CONNECTED  Connected to added network : InnoPhase\_AE  ------------ Program Exit ------------------- |

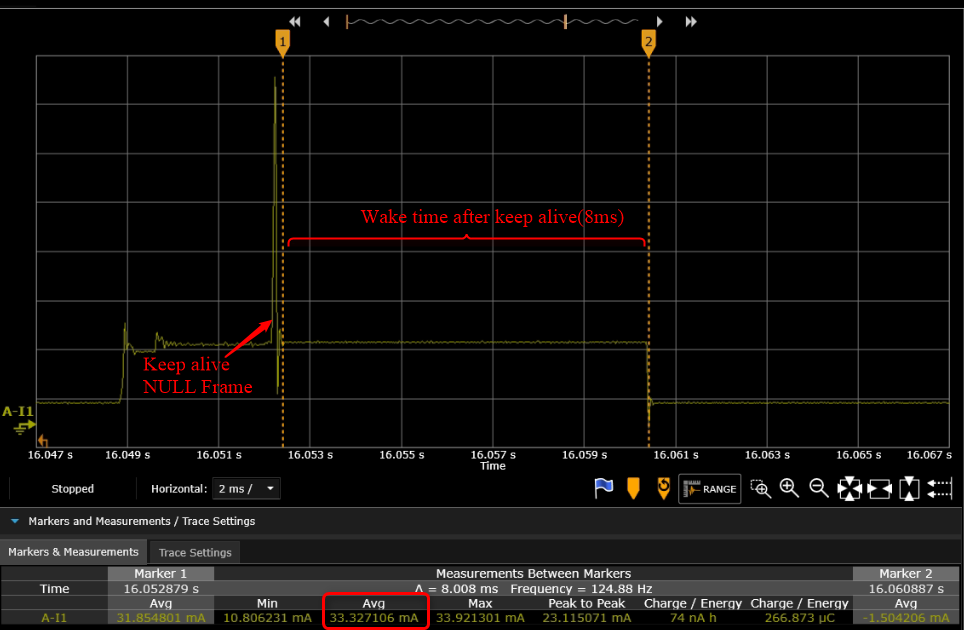


Figure : Power analyzer- wake time after keepalive