CSI Measurement from Associated Wi-Fi AP

The AX210/AX200 NIC can measure CSI for the 802.11a/g/n/ac/ax frames transmitted by the associated Wi-Fi AP. By generating sufficient Wi-Fi traffic, such as using the *ping* command, we can obtain CSI measurements.

To measure CSI from the AX210/AX200, follow these three steps:

1. Determine the PhyPath ID of the NIC by running the `array_status` command in a terminal. For device naming conventions of commercial NICs, please refer to the Device Naming for Commercial Wi-Fi NICs section.

2. Assuming the PhyPath ID is `3`, execute the following command:

   ```
   PicoScenes "-d debug -i 3 --mode logger --plot"
   ```

   The command mentioned above consists of four program options: `-d debug -i 3 --mode logger --plot`. These options can be interpreted as follows:
   - `-d debug`: Modifies the display level of the logging service to debug.
   - `-i 3 --mode logger`: Switches the device with ID 3 to CSI logger mode.
   - `--plot`: Live-plots the CSI measurements.

   For more detailed explanations, please refer to the Command Line Interface and Program Option Reference section.

3. Once you have collected sufficient CSI data, exit PicoScenes by pressing `Ctrl+C`.

The logged CSI data is stored in a file named `rx_<PHYPath>_<Time>.csi`, located in the *present working directory*. To analyze the data, open MATLAB, drag the .csi file into the *Command Window*, and the file will be parsed and stored as a MATLAB variable named `rx_<PHYPath>_<Time>`.

Fully-Passive CSI Measurement in Monitor Mode

The AX210/AX200 NIC is capable of measuring CSI for 802.11a/g/n/ac/ax frames observed in monitor mode. In this mode, the AX210/AX200 can passively measure CSI for all frames transmitted on the same channel, enabling association-free and injection-free fully passive CSI measurement.

To enable fully-passive CSI measurement, follow these three steps:

1. Determine the PhyPath ID of the NIC by running the `array_status` command in a terminal. Let's assume the PhyPath ID is `3`.

2. Put the NIC into monitor mode by executing the command `array_prepare_for_picoscenes 3 <CHANNEL_CONFIG>`. Replace `<CHANNEL_CONFIG>` with the desired channel configuration, specified in the same format as the *freq* setting of the Linux *iw set freq* command. For example, it could be "2412 HT20", "5200 HT40-", "5745 80 5775", and so on. Refer to Wi-Fi Channelization for more details.
3. Run the following command:

```
PicoScenes "-d debug -i 3 --mode logger --plot"
```

4. Once you have collected sufficient CSI data, exit PicoScenes by pressing Ctrl+C.

The command options `-d debug -i 3 --mode logger --plot` have the same behavior as described in the CSI Measurement from Associated Wi-Fi AP section.

The logged CSI data is stored in a file named `rx_<Id>_<Time>.csi`, located in the *present working directory*. To analyze the data, open MATLAB, drag the .csi file into the *Command Window*, and the file will be parsed and stored as a MATLAB variable named `rx_<Id>_<Time>`.

Packet Injection-Based CSI Measurement (Tx with 802.11a/g/n/ac/ax Format and 20/40/80/160 MHz CBW)

The PicoScenes Driver enables AX210/AX200 to *packet-inject* frames in 802.11a/g/n/ac/ax format with bandwidths of 20/40/80/160 MHz and up to 2x2 MIMO. By combining this capability with the CSI measurement functionality discussed in the Fully-Passive CSI Measurement in Monitor Mode section, PicoScenes provides precise, fine-grained control for CSI measurement.

To perform this example, you will need two computers, each equipped with an AX210/AX200 NIC. Please follow these three steps:

1. Determine the PhyPath ID of each NIC by using the `array_status` command. Let's assume the PhyPath ID is `3` for the first computer and `4` for the second.

2. Put both NICs into monitor mode by executing the command `array_prepare_for_picoscenes <PHYPath ID> <CHANNEL_CONFIG>`. Replace `<CHANNEL_CONFIG>` with the desired channel configuration. In this scenario, we assume the researchers want to measure the CSI of a 160 MHz channel. Run the following commands on the respective computers:

```
array_prepare_for_picoscenes 3 "5640 160 5250" #<-- Run on the first computer
array_prepare_for_picoscenes 4 "5640 160 5250" #<-- Run on the second computer
```

   Here, `5640 160 5250` represents a 160 MHz bandwidth channel centered at 5250 MHz with the primary channel at 5640 MHz. For more details, refer to Wi-Fi Channelization.

3. On the first computer, run the following command in a terminal:

```
PicoScenes "-d debug -i 3 --mode logger --plot"
```

4. On the second computer, assuming the researchers want to measure the CSI of 802.11ax format with 160 MHz bandwidth, run the following command in a terminal:

```
PicoScenes "-d debug -i 4 --mode injector --preset TX_CBW_160_HESU --repeat 1e5 --delay 5e3"
```

   The command options for the second computer, `"-d debug -i 4 --mode injector --preset TX_CBW_160_HESU --repeat 1e5 --delay 5e3"` , can be interpreted as follows:

   - `-d debug` : Modifies the display level of the logging service to debug.
   - `-i 4 --mode injector` : Switches the device <4> to packet injector mode.
   - `--preset TX_CBW_160_HESU`: Specifies the transmission (Tx) packet format using the preset named TX_CBW_160_HESU``, which corresponds to "Tx, channel bandwidth (CBW) 160 MHz, format=HESU (802.11ax single-user)".
   - `--repeat 1e5` : Transmits (or injects) 100,000 packets.
   - `--delay 5e3` : Sets the inter-packet delay to 5,000 microseconds.

5. Once you have collected sufficient CSI data on the first computer, exit PicoScenes by pressing Ctrl+C.

   The logged CSI data is stored in a file named `rx_<Id>_<Time>.csi` , located in the *present working directory* of the first computer. To analyze the data, open MATLAB, drag the .csi file into the *Command Window*, and the file will be parsed and stored as a MATLAB variable named `rx_<Id>_<Time>` .

> **Tips**
>
> You can refer to PicoScenes Presets for a full list of presets.

Packet Injection with MCS Setting and Antenna Selection

PicoScenes allows users to specify the MCS (Modulation and Coding Scheme) value and Tx/Rx antenna selection for AX210/AX200 NICs. To demonstrate this, we will modify the commands for the Packet Injection-Based CSI Measurement (Tx with 802.11a/g/n/ac/ax Format and 20/40/80/160 MHz CBW) scenario.

On the first computer, if you want to use only the 1st antenna for Rx, modify the command as follows:

```
PicoScenes "-d debug -i 3 --mode logger --rxcm 1 --plot"
```

The additional `--rxcm 1` option sets the Rx chainmask to 1, indicating the use of the 1st antenna for Rx. The `--rxcm` option allows you to specify the antenna selection using a bitwise style: 1 for the 1st antenna, 2 for the 2nd antenna, 3 for the first 2 antennas, 4 for the 3rd antenna, 5 for the 1st and 3rd antennas, and so on.

On the second computer, if you want to use only the 2nd antenna for Tx and specify the MCS value as 5, modify the command as follows:

```
PicoScenes "-d debug -i 4 --mode injector --preset TX_CBW_160_HESU --repeat 1e5 --delay 5e3 --txcm 2 --mcs 5"
```

The additional `--txcm 2` option sets the Tx chainmask to 2, indicating the use of the 2nd antenna for Tx. The `--txcm` option follows the same value style as `--rxcm` , but for transmission. The `--mcs 5` option sets the Tx MCS to 5.

If you want to measure the largest CSI with a 160 MHz bandwidth and 2x2 MIMO, further modifications are required. On the first computer, to receive 2x2 MIMO frames, you need to use 2 antennas for Rx. You can explicitly set `--rxcm 3` as shown below or simply remove the `--rxcm` option, which defaults to using `--rxcm 3` :

```
PicoScenes "-d debug -i 3 --mode logger --rxcm 3 --plot"
```

On the second computer, to transmit 2x2 MIMO frames, you also need to use 2 antennas for Tx. You can explicitly set `--txcm 3` as shown below or simply remove the `--txcm` option, which defaults to using `--txcm 3` :

```
PicoScenes "-d debug -i 4 --mode injector --preset TX_CBW_160_HESU --repeat 1e5 --delay 5e3 --mcs 5 --sts 2"
```

The additional `--sts 2` option sets the number of Space-Time Streams `N_{STS}=2` to 2, indicating the use of two antennas to transmit 2x2 MIMO frames.

Specifying Channel and Bandwidth in Real-time

PicoScenes provides the `--channel` option to change channel settings in real-time, without re-execution of the `array_prepare_for_picoscenes` command. For example, assuming you have an AX210/AX200 NIC with ID <3> working at an 80 MHz CBW channel "5180 80 5210" (refer to Wi-Fi

Channelization for details), and you want to change the NIC to listen on a 160 MHz CBW channel "5955 160 6025", you can directly run the command:

```
PicoScenes "-d debug -i 3 --channel '5955 160 6025' --preset TX_CBW_160_HESU --mode logger --plot"
```

The `--channel '5955 160 6025'` option directly changes the channels without requiring the `array_prepare_for_picoscenes` script to be executed again.

Concurrent Multi-NIC Operation on a Single Computer

PicoScenes supports the installation and control of multiple Wi-Fi NICs on a single computer. To set up a multi-NIC configuration, please refer to the Installation of (Multiple) Commercial Wi-Fi NICs section. Assuming you have installed two or more AX210 or AX200 NICs on your computer and you want to use one NIC for transmission (Tx) and the rest for reception (Rx) and CSI measurement, you can use the following commands:

```
array_prepare_for_picoscenes "3 4 5" "5955 160 6025"

PicoScenes "-d debug;
        -i 5 --mode logger --plot;
        -i 4 --mode logger --plot;
        -i 3 --mode injector --preset TX_CBW_160_HESU --repeat 1e5 --delay 5e3;
        -q;"
```

Let's explain these two commands:

- The `array_prepare_for_picoscenes` command adds monitor interfaces for NICs 3, 4, and 5, and sets their working channels to '5955 160 6025'. Refer to the Wi-Fi Channelization section for more examples.
- The CLI input is a multi-line input, where each line corresponds to a NIC. Lines are separated by semicolons (**;**).
    - The second and third lines put NIC 4 and 5 into logger mode and activate the corresponding live plot. Please note that *logger mode is non-blocking*. This non-blocking design enables concurrent Rx and CSI measurement for multiple NICs.
    - The fourth line specifies that NIC 3 should transmit frames with a 160 MHz CBW in HESU format for 100,000 times.
    - The last line `-q` or `--quit` indicates that the program should exit when there are no more jobs to process.

Tips