EUROPEAN UNION

Interreg
Austria-Czech Republic
European Regional Development Fund

ATCZ175 InterOP PROJECT

WLAN Test Setup Instruction Guide

Institute of Electrodynamics, Microwave and Circuit Engineering
Technische Universität Wien

Advisor:
Assoc. Prof. Dipl.-Ing. Dr.techn. Holger Arthaber

by
Proj. Ass. Dipl.-Ing. Christian Spindelberger

October 9, 2019

TU WIEN

EMCE

Contents

Abbreviations

IP internet protocol

PHY physical

RF radio frequency

TCP transmission control protocol

UDP user datagram protocol

WLAN wireless local area network

1 Software Configuration

This configuration guide provides information about installing the required WLAN test setup. As this manual is based on a Linux operating system, basic knowledge about such systems and respective programming skills are mandatory. In the following, utilization of the required kernel and further instructions, such as firmware installation and adaptation of transmission parameters, will be discussed.

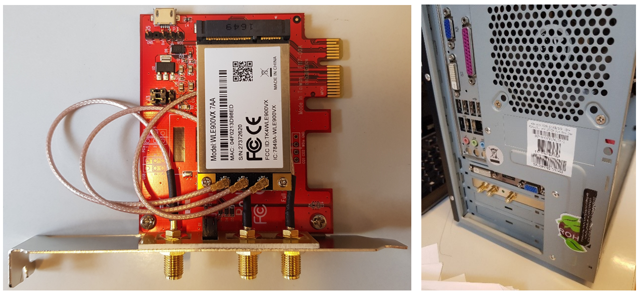

Figure 1: WLAN module (left), installed in PC (right)

The measurement setup mainly consists of two PCs with the distribution Ubuntu LTS 18.04, two *Atheros* WLAN cards (*WLE900VX*) of 10th generation (*ath10k*) with a proper mini-PCIe to PCIe adapter and additional RF equipment. As already mentioned, only single channel scenarios are investigated. The respective WLAN cards are therefore connected over one port via coaxial cables, a coupler, and a variable attenuator. The remaining ports are terminated with $50\,\Omega$ loads. Figure 1 depicts the WLAN module with the according PCIe adapter. Furthermore, three UFL-to-SMA cables are utilized to connect further components, such as the coupler and attenuator.

1.1 Precompiled Kernel

In order to characterize different interference sources, it is necessary to have full control of PHY parameters. *Candela Technologies* (CT) offers a precompiled Linux kernel, which enables specifying, for instance, utilized bandwidth or transmit power. Typically, Linux offers commands out of the box to change most of these settings. Unfortunately, not all devices are supported by current drivers. Hence, CTs kernel, which can be downloaded on their homepage (`www.candelatech.com`), must be considered.

In order to install the required kernel, download the compressed file and save it to the home directory. Subsequently, untar the file and configure the bootloader *GRUB* appropriately.

1.2 Firmware Installation

Current firmware versions for *ath10k* devices suffer from several software errors. CT implemented a further developed version, which can be downloaded for the respective WLAN module category. Mainly, two different types, called **wave-1** and **wave-2**, exist. The first type describes single user MIMO systems, which are capable of transmitting data to one single station at a time. Furthermore, such devices are limited to three spatial streams, resulting in a 3x3 MIMO system and a maximum bandwidth of 80 MHz. Improving this category led to **wave-2** enabling multi-user MIMO. Simply put, it is possible to transmit to several stations at the same time. This is achieved by beamforming and broadening the bandwidth up to 160 MHz, yielding a higher throughput compared to **wave-1**. As no multi-user scenario is investigated, a **wave-1** module was chosen.

In this setup, **wave-1** *WLE900VX* WLAN cards with a *QCA988X* chipset are utilized. In the following, all instructions are based on this device category. After choosing the right firmware version from CT homepage, the following instructions implement the desired software [1]:

```
mkdir -p /lib/firmware/ath10k/QCA988X/hw2.0/orig
mv /lib/firmware/ath10k/QCA988X/hw2.0/firmware-[3456].
   bin /lib/firmware/ath10k/QCA988X/hw2.0/orig/
cp firmware-2-ct-full-community.bin /lib/firmware/
   ath10k/QCA988X/hw2.0/firmware-2.bin
```

Furthermore, the actual **board.bin** file from the official firmware site of *Atheros* has to be saved into the directory mentioned above. After rebooting the system, one can check with **ethtool** if the right firmware version has been loaded. If the output contains a **"-ct"**, such as:

```
ethtool -i wlan0
10.1-ct-8x-_xtH-019-ddf2a35
```

the installation was successful.

1.3 Network Configuration

Connecting the two WLAN cards requires a static IP assignment for both sides (server and client). Notice that the cards can communicate only if they share the same **netmask** (255.255.255.0). In this setup, the server station has the IP address **10.0.0.1** and the client **10.0.0.2**. In order to set the network settings permanently, the **/etc/network/interfaces** file must be changed accordingly:

```
# network file /etc/network/interfaces

auto lo
iface lo inet loopback

auto wlan0
iface wlan0 inet static
address 10.0.0.1
netmask 255.255.255.0
```

With the **ifconfig** command, one can check the current IP address of the respective WLAN module after a reboot. The assigned IP address can be found next to excerpt *inet*.

```
ifconfig wlan0
wlan0:  flags=4163<....>  mtu 1500
inet 10.0.0.1  netmask 255.255.255.0
ether ff:ff:ff:ff:ff:ff
txqueuelen 1000  (Ethernet)
```

Access Point Configuration

One PC will act as an access point (server), which is configured in the hostapd file: **/etc/hostapd/hostapd.conf**. This file has a huge amount of configurable settings which define access point capabilities. For instance, it is possible to choose drivers, frequency bands, cyclic prefix lengths, and many more. The following hostapd file configures an access point called "*my_ap*" for WLAN channel 1 in the 2.4 GHz band. It enables the IEEE 802.11n standard up to 40 MHz. Key managements and a short CP of 0.4 µs have been disabled. For detailed information, please refer to the hostapd configuration file description: `https://w1.fi/cgit/hostap/plain/hostapd/hostapd.conf`

```
# hostapd file /etc/hostapd/hostapd.conf

interface=wlan0
driver=nl80211
ssid=my_ap
channel=1
ignore_broadcast_ssid=0
country_code=US
ieee80211d=1
hw_mode=g
macaddr_acl=0
```

```
auth_algs=1
ieee80211n=1
ieee80211h=1
ht_capab=[HT40+]
```

To start the **hostapd daemon** per default, please uncomment and change the following line in file **/etc/default/hostapd**:

```
DAEMON_CONF = /etc/hostapd/hostapd.conf
```

After rebooting the system, one can check with **iwconfig wlan0** if the changed settings have been configured successfully. The output should then state something like this:

```
iwconfig wlan0
wlan0    IEEE 802.11  Mode:Master
Tx-Power=30 dBm
Retry short limit:7   RTS thr:off
Fragment thr:off
Power Management:on
```

WPA Supplicant

To ensure a stable connection between the two WLAN cards (server and client) the **wpa supplicant** has to be configured. This daemon will force the client to associate with the specified access point automatically. First of all, a **wpa_supplicant.conf** file has to be created in the following directory: **/etc/wpa_supplicant**.

```
# wpa_supplicant file
# /etc/wpa_supplicant/wpa_supplicant.conf

ctrl_interface=/var/run/wpa_supplicant
eapol_version=1
ap_scan=1

network={
ssid = "my_ap"
key_mgmt=NONE
}
```

According to the **hostapd** file mentioned above, no key encryption is utilized. For a more detailed description of the **wpa_supplicant.conf** file, refer to the following link: https://w1.fi/cgit/hostap/plain/wpa_supplicant/wpa_supplicant.conf.

To activate the **wpa supplicant**, the following line must be added to the **/etc/network/interfaces** file below the desired interface:

```
wpa-conf /etc/wpa_supplicant/wpa_supplicant.conf
```

The applied configurations can be tested with the following line:

```
wpa_supplicant -i wlan0 -D wext -c /etc/wpa_supplicant/
    wpa_supplicant.conf -d
```

Afterwards, **iwconfig** shows if connecting to the desired access point was successful:

```
iwconfig wlan0
wlan0      IEEE 802.11  ESSID:"my_ap"
Mode:Managed  Frequency:2.412 GHz
Access Point: ff:ff:ff:ff:ff:ff
Bit Rate=1 Mb/s   Tx-Power=30 dBm
Retry short limit:7   RTS thr:off
Fragment thr:off
Power Management:on
Link Quality=26/70  Signal level=-84 dBm
```

The extended service set identifier (ESSID) mentions the associated access point. In this example, the ESSID is "*my_ap*", indicating a successful connection with the desired server station.

2 WLAN PHY Configuration

After successful implementation of the CT kernel and applying the settings mentioned above, full control of PHY parameters is available. Basically, all PHY settings like bandwidth, modulation or transmit power are adjusted and explained by the **ct_special** file, located in the **/sys/kernel/debug/ieee80211/ph0/ath10k/** directory. In order to change PHY characteristics, the **echo** command can be utilized. All parameter settings are encoded by an ID and a 64 bits long codeword.

2.1 Set Transmit Power

As WLAN modules suffer from insufficient isolation, one of the most important settings is the transmit power. Hence, it might happen that transmitted data packets between server and client are also received over the air. Concentrating the established traffic onto the desired wired measurement setup requires a transmit power as low as possible. The default level is 20 dBm for 2.4 GHz and 23 dBm for 5 GHz per spatial stream. The following command sets the transmit power to a minimum of 0 dBm. The ID of this command is 6, followed by 8 hexadecimal numbers defining the transmit power:

```
echo 0x600000000 > /sys/kernel/debug/ieee80211/phy0/
   ath10k/ct_special
```

It is not possible to set these parameters multiple times. Thus, it is recommended to minimize the transmit power once before proceeding with performance tests using **iperf**. In addition to this, one must be careful with this parameter, too large values may damage the device.

HT-MCS	**Modulation & Coding**	**Data rate** **BW** = 20 MHz **CP** = 0.8 µs	**Data rate** **BW** = 40 MHz **CP** = 0.8 µs
0	BPSK-1/2	6.5 Mbit/s	13.5 Mbit/s
1	QPSK-1/2	13 Mbit/s	27 Mbit/s
2	QPSK-3/4	19.5 Mbit/s	40.5 Mbit/s
3	16-QAM-1/2	26 Mbit/s	54 Mbit/s
4	16-QAM-3/4	39 Mbit/s	81 Mbit/s
5	64-QAM-2/3	52 Mbit/s	108 Mbit/s
6	64-QAM-3/4	58.5 Mbit/s	121.5 Mbit/s
7	64-QAM-5/6	65 Mbit/s	135 Mbit/s

Table 1: IEEE 802.11n modulation indices

2.2 Set Modulation Parameters

Utilizing the **iw** command offers the opportunity to set certain modulation parameters, yielding different throughputs and PERs. The IEEE 802.11n standard, also called high throughput (HT) modus, defines modulations utilizing indices (MCS). Table 1 quotes the MCS parameters for single spatial streams. The following example defines the high throughput (HT) modulation index 0 for 2.4 GHz:

```
iw dev wlan0 set bitrates legacy-2.4 ht-mcs-2.4 0 vht-
   mcs-2.4
```

Furthermore, the legacy rate (WLAN DSSS for 2.4 GHz) and very high throughput values (IEEE 802.11ac) can be set. As the main focus in this project lies on IEEE 802.11n, another example for setting MCS-3 in the 5 GHz band is given:

```
iw dev wlan0 set bitrates legacy-5 ht-mcs-5 3 vht-mcs-5
```

2.3 Defining Certain Transmit Bandwidths

According to Table 1, a lot of different modulation types exist. Therefore, the respective bandwidth has to be adapted to ensure that the desired modulation is utilized. The definition of three possible bandwidths is stated in the following example:

```
# 20MHz
echo 0xE00000006 > /sys/kernel/debug/ieee80211/phy0/
   ath10k/ct_special
# 40MHz
echo 0xE00000005 > /sys/kernel/debug/ieee80211/phy0/
   ath10k/ct_special
# 80MHz
echo 0xE00000003 > /sys/kernel/debug/ieee80211/phy0/
   ath10k/ct_special
```

Further encoding parameters and their definitions can be found in the **ct_special** file. It must be mentioned that the applied transmit bandwidth will only take effect until the modulation parameters have been set [1].

3 Performance Tests

Now, all necessary settings for a successful performance test using **iperf3** have been introduced. After connecting all necessary devices, an initialization sequence of the test setup will be explained. The following adjustments must be applied on both sides, server and client.

1. Set the transmit power to 0 dBm:

   ```
   echo 0x600000000 > /sys/kernel/debug/ieee80211/phy0
      /ath10k/ct_special
   ```

2. Disable fragmentation- and RTS thresholds:

   ```
   iwconfig wlan0 rts off
   ```

3. Set the desired transmit bandwidth (20 MHz):

   ```
   echo 0xE00000006 > /sys/kernel/debug/ieee80211/phy0
      /ath10k/ct_special
   ```

4. Setting desired modulation index (HT-MCS-0):

   ```
   iw dev wlan0 set bitrates legacy-2.4 ht-mcs-2.4 0
      vht-mcs-2.4
   ```

After initializing both systems, the **iperf3** client can be utilized for performance measurements. Mainly, two performance tests, based on UDP and TCP, are available. Independent of the utilized protocol, the **iperf3** server must be started on the access point, listening to a specified port. If the server is ready for data reception, the client can start **iperf3** as well. The following sequence extends the initialization process mentioned above.

5. Start **iperf3** server (IP: 10.0.0.1) on the access point:

   ```
   iperf3 -s
   ```

6. Start the performance test (UDP, packet size 100 Byte):

   ```
   iperf3 -c 10.0.0.1 -u -b 10G -l 100B -f m
   ```

According to the listing above, the server station starts listening to incoming packets. Subsequently, the client starts a UDP test with the respective destination IP address (10.0.0.1), a maximum target bandwidth of 10 Gbits/s, and a packet size of 100 Byte. The target bandwidth must always be set to higher values than the desired modulation is capable to achieve. Otherwise, the **iperf3** performance results will not yield maximum throughput values. For further informations about adjusting **iperf3**, refer to `https://iperf.fr/iperf-doc.php`.

References

[1] Candela Technologies. `https://www.candelatech.com/` 2, 7