Wi-Fi

User Guide

Issue 02

Date 2015-05-06

Trademarks and Permissions

 , HISILICON, and other HiSilicon icons are trademarks of HiSilicon Technologies Co., Ltd.

All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

The purchased products, services and features are stipulated by the contract made between HiSilicon and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

HiSilicon Technologies Co., Ltd.

Address: Huawei Industrial Base
Bantian, Longgang
Shenzhen 518129
People's Republic of China

Website: http://www.hisilicon.com

Email: support@hisilicon.com

About This Document

Purpose

This document describes the configurations, basic operations, and debugging methods of the Wi-Fi module as well as precautions and solutions to common problems.

Related Versions

The following table lists the product versions related to this document.

Product Name	Version
Hi3798C	V1XX
Hi3796C	V1XX
Hi3798M	V1XX
Hi3796M	V1XX
Hi3798C	V2XX

Intended Audience

This document is intended for:

- Technical support engineers
- Software R&D engineers

Change History

Changes between document issues are cumulative. Therefore, the latest document issue contains all changes made in previous issues.

Issue 02 (2015-05-06)

This issue is the second official release, which incorporates the following changes:

Hi3798C V200 is supported.

Chapter 1 Configuration Description

Sections 1.1 and 1.2 are modified.

Chapter 2 Basic Wi-Fi Operations

Section 2.3 is modified.

Chapter 3 Tests

Section 3.4 is added.

Issue 01 (2014-10-30)

This issue is the first official release, which incorporates the following change:

Hi3796M V100 is supported.

Issue 00B01 (2014-06-20)

This issue is the first draft release.

Contents

Figures

1 Configuration Description

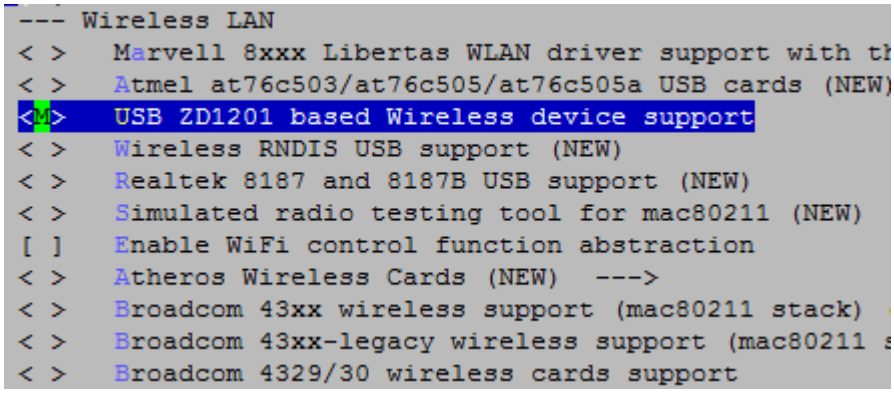
1.1 Kernel Configurations

1.1.1 Configuring WEXT

Wireless extension (WEXT) is the standard interface for the Wi-Fi driver in the kernel and user-mode process.

In some kernel versions, there is no WEXT configuration option. You need to choose **Device Drivers** > **Network device support** > **Wireless LAN**, and enable drivers that are dependent on **CONFIG_WIRELESS_EXT=y**. Then **CONFIG_WIRELESS_EXT** is automatically enabled. For example, set **USB ZD1201 based Wireless device support** to **M**, as shown in Figure 1-1.

Figure 1-1 WEXT configuration



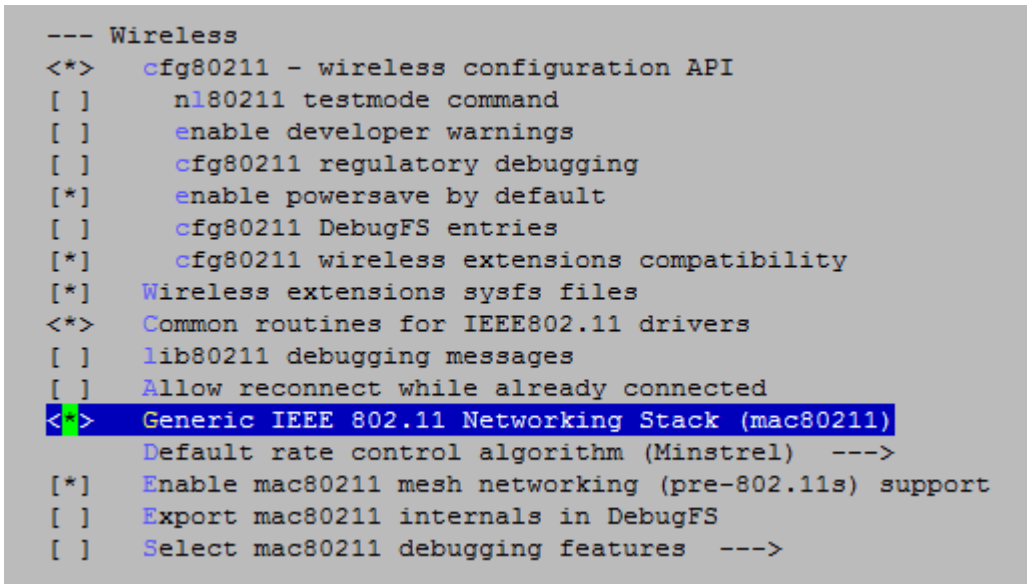
As shown in Figure 1-1, the kernel includes multiple Wi-Fi drivers, but drivers in the kernel are not used actually. Drivers are directly obtained from the vendors and integrated into the SDK. Therefore, do not enable the Wi-Fi driver of the same model in the kernel; otherwise, compilation conflict may occur.

1.1.2 Configuring CFG80211

CFG80211 is the standard interface for the Wi-Fi driver in the kernel and user-mode process. It becomes more popular than WEXT. Only CFG80211 supports the Wi-Fi Direct function.

Choose **Network support** > **Wireless**, and set **cfg80211** and **mac80211** to **y**, as shown in Figure 1-2.

Figure 1-2 CFG80211 configuration



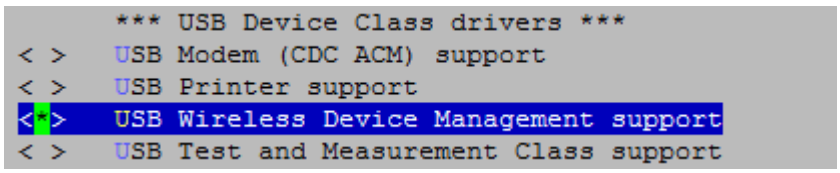
CAUTION

The Wi-Fi driver of Atheros contains the modified CFG80211 program, and you can use only this CFG80211. If the CFG80211 in the kernel is set to **y**, compilation conflict occurs for the CFG80211 in the Wi-Fi driver. Therefore, **cfg80211** and **mac80211** in the kernel must be set to **M**.

1.1.3 Configuring the USB

For details about the USB operations, see the *Peripheral User Guide*. To enable the support for the Wi-Fi, choose **Device Drivers** > **USB support**, and select **USB Wireless Device Management support**.

Figure 1-3 USB configuration



1.1.4 Configuring the Netlink

NOTE

Netlink is used to implement communications between the wpa_supplicant/hostpad modules and the kernel. Therefore, you need to configure the Netlink.

Choose **Network support** > **Networking options**, and set **Network packet filtering framework (Netfilter)** to **y**. Set **Advanced netfilter Configuration** to **y** in **Network packet filtering framework (Netfilter)**. Then set **Core Netfilter Configuration** as follows:

Figure 1-4 Netlink configuration

```
<*> Netfilter NFACCT over NFNETLINK interface
<*> Netfilter NFQUEUE over NFNETLINK interface
< > Netfilter LOG over NFNETLINK interface
<*> Netfilter connection tracking support
[ ]   Connection mark tracking support (NEW)
[ ]   Supply CT list in procfs (OBSOLETE) (NEW)
[ ]   Connection tracking events (NEW)
[ ]   Connection tracking timeout (NEW)
[ ]   Connection tracking timestamping (NEW)
< >   DCCP protocol connection tracking support (NEW)
< >   SCTP protocol connection tracking support (NEW)
< >   UDP-Lite protocol connection tracking support (NEW)
< >   Amanda backup protocol support (NEW)
< >   FTP protocol support (NEW)
< >   H.323 protocol support (NEW)
< >   IRC protocol support (NEW)
< >   NetBIOS name service protocol support (NEW)
< >   SNMP service protocol support (NEW)
< >   PPtP protocol support (NEW)
< >   SANE protocol support (NEW)
< >   SIP protocol support (NEW)
< >   TFTP protocol support (NEW)
< >   Connection tracking netlink interface (NEW)
< >   Connection tracking timeout tuning via Netlink (NEW)
[ ]   NFQUEUE integration with Connection Tracking (NEW)
-*- Netfilter Xtables support (required for ip_tables)
      *** Xtables combined modules ***
```

The preceding kernel options have been properly configured on the platform by default.

1.1.5 Configuring the NAT Forwarding

If the network sharing function of the SoftAP is required, choose **Network support** > **Networking options** > Network packet filtering framework (Netfilter) > **Core Netfilter Configuration**, and set **Netfilter connection tracking support** to **y**. Then choose **Network support** > Networking options > Network packet filtering framework (Netfilter) > **IP: Netfilter Configuration**, and configure the items as follows:

Figure 1-5 NAT configuration

```
<*> IPv4 connection tracking support (required for NAT)
[*]   proc/sysctl compatibility with old connection tracking (NEW)
<*> IP tables support (required for filtering/masq/NAT)
< >   "ah" match support
< >   "ecn" match support
< >   "ttl" match support
< >   Packet filtering
< >   ULOG target support
<*>   IPv4 NAT
<*>     MASQUERADE target support
<*>     NETMAP target support
<*>     REDIRECT target support
< >   Packet mangling
< >   raw table support (required for NOTRACK/TRACE)
<*> ARP tables support
< >   ARP packet filtering
< >   ARP payload mangling
```

The network sharing function is required by default on the Android platform and has been configured in the SDK. However, this function is not configured on the Linux platform by default.

To enable support for IPv6, choose **Network support** > **Networking options** > **Network packet filtering framework (Netfilter)** > **IPv6: Netfilter Configuration**, and set the items based on the configurations for IPv4.

1.2 Bootargs Configurations

1.2.1 Configuring the Atomic Memory

The RT3070, RT5370, RT5372, RT5572, and MT7601U drivers use a large atomic memory. The default size of the memory is 256 KB in the kernel, which is insufficient. Therefore, the driver may fail to be loaded because the required memory cannot be obtained, and the system displays "Failed to allocate memory". If you use the preceding Wi-Fi models, you need to set the atomic memory to 1 MB as follows (the MT7632U requires 2 MB atomic memory and you need to set **coherent_pool=2M** in bootargs):

Step 1 Add **coherent_pool=1M** to the bootargs configurations as follows:

```
bootargs=mem=1G console=ttyAMA0,115200 root=/dev/mtdblock3
rootfstype=yaffs2
mtdparts=hi_sfc:512K(boot),64K(bootargs);hinand:6M(kernel),96M(rootfs),20
M(test),-(other) coherent_pool=1M
```

Note that a space is required between the added data and the previous content.

Step 2 Generate the bootargs file by running **mkbootargs**, and then burn the file into the board.

----End

1.3 Compilation Configurations

1.3.1 Compilation Configurations for Linux

To configure compilation options for Linux, perform the following steps:

Step 1 Run **make menuconfig** in the SDK root directory, choose **component**, and enable **WiFi Support**.

Step 2 Choose **WiFi Support** > **WiFi Device Type**, and set the Wi-Fi chip type.

CAUTION

All supported Wi-Fi chips are listed here. You can select multiple Wi-Fi chips at the same time. However, you are advised to enable only the required Wi-Fi chips because the compilation may be slowed and the file system becomes too large if too many chips are supported.

Step 3 Choose **WiFi Support** > **WiFi Working Mode**, and set the Wi-Fi mode to STA or SoftAP mode or both.

Step 4 Set the file system size.

If too many Wi-Fi chips are selected, the file system may fail to be generated because its size may exceed the default size 96 MB. In this case, you need to set the size of the file system to a larger value. (If no error message is displayed during file system generation, skip this step.) Choose **Rootfs** > **File System Config** > **eMMC Rootfs Size**, and change **96** to a larger value.

----End

1.3.2 Compilation Configurations for Android

NOTE

Some user-mode programs need to be compiled on the Android platform. Therefore, the Wi-Fi is configured in the common configuration file **BoardConfig.mk** of the Android platform but not the SDK.

BoardConfig.mk lists all Wi-Fi chips supported by the Android platform, each of which is configured as BOARD_WLAN_DEVICE_*chip name*. The value **y** indicates that the compiled image supports the Wi-Fi chip, the value **n** or other values indicate that the compiled image does not support the Wi-Fi chip.

For example, if Hi3798C V100 supports RTL8188EUS, you can modify **device/hisilicon/Hi3798CV100/BoardConfig.mk** as follows:

```
BOARD_WLAN_DEVICE_RTL8188EUS := y
```

2 Basic Wi-Fi Operations

2.1 Operation Examples for the STA Mode

2.1.1 Checking the Wi-Fi Device

Check whether the Wi-Fi device works properly before enabling the Wi-Fi.

Run the shell command **lsusb** or **busybox lsusb**.

The USB device IDs are displayed:

```
Bus 001 Device 004: ID 0bda:8179
Bus 001 Device 001: ID 1d6b:0002
Bus 002 Device 001: ID 1d6b:0001
```

0bda:8179 is the ID of the USB device RTL8188EUS. If the ID is displayed, the device is detected.

If the **lsusb** and **busybox lsusb** commands are not supported, go to the **/sys/bus/usb/devices** directory. There are multiple sub directories in the **/sys/bus/usb/devices** directory. Each sub directory stores the information of a USB device, and the **uevent** file stores information such as device models and device IDs. View the **PRODUCT=*xx/xx/xx*** statement in the **uevent** file in each sub directory to check whether there is a Wi-Fi device ID. If yes, the Wi-Fi device is detected; if no, the Wi-Fi device is not inserted, not powered on, or damaged.

Figure 2-1 RTL8188EUS USB device ID

2.1.2 Loading Drivers

To load drivers, perform the following steps:

Step 1 Load the CFG80211 driver.

If the CFG80211 driver is used, load **cfg80211.ko** first.

Go to the directory that stores **cfg80211.ko** and run the following shell command:

```
insmod cfg80211.ko
```

Step 2 Load the Wi-Fi driver by running the following shell command in the directory that stores the .ko file:

```
insmod rtl8188eu.ko
```

Typically the driver varies with the Wi-Fi chip, but some Wi-Fi chips share the same driver. For example, the drivers for RTL8188ETV and RTL8188EUS are the same, and those for RTL8188CUS and RTL8192CU are also the same.

Step 3 Check whether the driver is successfully loaded by running the following shell command:

```
iwconfig
```

If the network port wlan0 exists, the driver is successfully initialized, and the Wi-Fi device is available.

Figure 2-2 Iwconfig execution result

```
# iwconfig
lo        no wireless extensions.

eth0      no wireless extensions.

wlan0     unassociated  Nickname:"<WIFI@REALTEK>"
          Mode:Auto  Frequency=2.412 GHz  Access Point: Not-Associated
          Sensitivity:0/0
          Retry:off   RTS thr:off   Fragment thr:off
          Encryption key:off
          Power Management:off
          Link Quality:0  Signal level:0  Noise level:0
          Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
          Tx excessive retries:0  Invalid misc:0   Missed beacon:0
```

Step 4 Enable the Wi-Fi network port by running the following shell command:

```
ifconfig wlan0 up
```

After the preceding command is executed, the Wi-Fi is available, and you can perform the scan and connect operations.

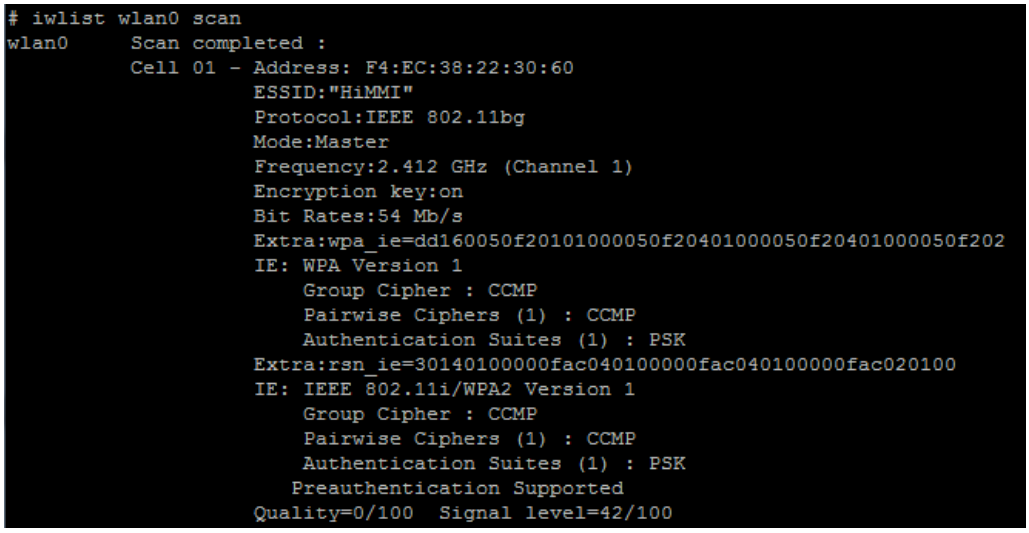
----End

2.1.3 Scanning for APs

Run the following shell command:

```
iwlist wlan0 scan
```

Figure 2-3 Scan result



The detected APs are displayed in the format of Cell *xx*, and each AP corresponds to a Cell *xx*.

The AP information includes the following:

- **Address**: MAC address
- **ESSID**: AP name, that is, SSID
- **Protocol**: IEEE80211 protocol, 11b/g/n
- **Frequency**: frequency
- **Encryption key** (authentication encryption information): WEP, WPA-PSK, WPA2-PSK, WPA, WPA2
- **Quality**: signal quality. This data is sometimes inaccurate and can be ignored.
- **Signal level**: signal strength. The larger the value, the stronger the signal strength. The display mode of the signal level varies with the Wi-Fi chip, for example, *xx*/100 or *xx* dBm.

The display format of the preceding information varies with the Wi-Fi chip.

CAUTION

When you scan for APs by running the **iwlist** command, the scan result is returned not necessarily after all frequencies are scanned. Therefore, some APs cannot be detected, especially for MT7601U. MT7601U scans each frequency for a long time period, and therefore only APs at one or two frequencies can be detected during the first scan.

2.1.4 Connecting to an AP

The wpa_supplicant process is used to connect the Wi-Fi device to an AP. wpa_supplicant is an open-source code which is used on Linux and Android to implement the Wi-Fi connection

process. It includes protocols such as WEP, WPA/WPA2, WPA-PSK/WPA2-PSK, WAPI, WPS, P2P, and EAP.

To connect to an AP, perform the following steps:

Step 1 Start the wpa_supplicant process by running the following shell command:

```
wpa_supplicant –iwlan0 –Dnl80211 –c/etc/Wireless/wpa_supplicant.conf&
```

- **-iwlan0** indicates that the network port wlan0 is used.
- **-Dnl80211** indicates that the cfg80211 interface is used (libnl for user-mode interfaces and cfg80211 for the kernel). Another option is **-iwext**, indicating that the wext interface is used.
- **-c/xxx/wpa_supplicant.conf** indicates the configuration file of wpa_supplicant. Ensure that the file exists.

After the command is executed, run the **ps** command to check whether the wpa_supplicant process exists. If yes, it works properly. If no, increase the wpa_supplicant print level and find out the cause from the logs.

```
wpa_supplicant –iwlan0 –Dnl80211 –c/etc/Wireless/wpa_supplicant.conf –ddd
&
```

Step 2 Start the wpa_cli process by running the following shell command:

```
wpa_cli –iwlan0
```

If the preceding command is successfully executed, the symbol ">" is displayed.

If "Could not connect to wpa_supplicant - re-trying" is displayed, socket connection cannot be set up between wpa_cli and wpa_supplicant. In this case, check whether the wpa_supplicant process exists, whether **/var/run/wpa_supplicant/wlan0** exists, and whether **ctrl_interface=/var/run/wpa_supplicant** exists in **wpa_supplicant.conf**.

Step 3 Scan for APs.

Run the **scan** command after >, and run **scan_results** after **CTRL-EVENT-SCAN-RESULTS** is received. The scan result is displayed.

Figure 2-4 wpa_cli scan result

```
> scan
OK
<3>CTRL-EVENT-SCAN-RESULTS
<3>WPS-AP-AVAILABLE

> > scan_results
bssid / frequency / signal level / flags / ssid
78:a1:06:48:e2:e8       2472    -65     [WPA-PSK-CCMP][WPA2-PSK-CCMP][WPS][ESS] B21-1
40:4d:8e:81:08:f1       2462    -69     [WPA-PSK-TKIP][ESS]     B25_chenxie
f4:ec:38:22:30:60       2412    -74     [WPA-PSK-CCMP][WPA2-PSK-CCMP-preauth][ESS]      HiMMI
8c:21:0a:a5:cd:b2       2437    -48     [WEP][ESS]      B21
```

Step 4 Connect to an AP.

1. Take the connection to an open AP as an example. Run **add_network** after >. Assume that the returned network ID is 0.
2. Configure the network SSID by running **set_network 0 ssid** (SSID of the AP).
3. Configure the network encryption mode by running **set_network 0 key_mgmt NONE**.

4. Enable the network by running **enable_network 0**.

If "CTRL-EVENT-CONNECTED" is received, the AP is successfully connected.

Figure 2-5 Connecting to an AP

Step 5 Obtain the IP address.

Enter **q** to exit wpa_cli, and run the shell command **udhcpc –i wlan0**.

After obtaining the IP address, run the **ping** command to check it is available.

----End

2.2 Operation Examples for the SoftAP Mode

The **iwpriv** command is used to operate the SoftAP for RT3070, RT5370, RT5372, RT5572, and MT7601U, while the hostapd process is used for other Wi-Fi devices. The hostapd process is similar to wpa_supplicant. It contains various authentication protocols and connection processes of the AP end, while wpa_supplicant belongs to the STA end.

2.2.1 Checking the Wi-Fi Device and Loading the Driver

The processes for checking the Wi-Fi device and loading the driver are the same as those for the STA mode.

2.2.2 Configuring and Enabling the SoftAP by Running iwpriv

To configure and enable the SoftAP by running the **iwpriv** command, perform the following steps:

Step 1 Enable the SoftAP by running the following shell command:

```
ifconfig wlan0 up
```

The driver reads parameters in **/etc/Wireless/RT2870AP/RT2870AP.dat**, and then initializes and enables the SoftAP.

Step 2 Configure the channel by running the following shell command:

```
iwpriv wlan0 set Channel=6
```

The channel ID ranges from 1 to 11.

Step 3 Configure the encryption mode.

- OPEN

```
iwpriv wlan0 set AuthMode=OPEN
iwpriv wlan0 set EncrypType=NONE
```

- WEP

```
iwpriv wlan0 set AuthMode=WEPAUTO
iwpriv wlan0 set EncrypType=WEP
iwpriv wlan0 set DefaultKeyID=1
iwpriv wlan0 set Key1=xxxxx
```

- WPA2-PSK

```
iwpriv wlan0 set AuthMode=WPA2PSK
iwpriv wlan0 set EncrypType=AES
iwpriv wlan0 set WPAPSK=xxxxxxxx
```

Step 4 Configure the SSID by running the following shell command:

```
iwpriv wlan0 set SSID=XXX
```

----End

2.2.3 Configuring and Enabling the SoftAP by Using the hostapd Process

To configure and enable the SoftAP by using the hostapd process, perform the following steps:

Step 1 Modify **hostapd.conf**.

The hostapd process requires the configuration file **hostapd.conf**. You can set the SSID, channel, and encryption mode in the configuration file. The following shows the examples of configuration files:

- OPEN

```
interface=wlan0
driver=nl80211
ctrl_interface=/var/run/hostapd
ssid=HisiAP
channel=6
hw_mode=g
ieee80211n=1
ht_capab=[SHORT-GI-20][SHORT-GI-40] [HT40-]
```

- WEP

```
interface=wlan0
driver=nl80211
ctrl_interface=/var/run/hostapd
ssid=HisiAP
```

```
channel=6
hw_mode=g
wep_default_key=0
wep_key0="12345"
```

- WPA2-PSK

```
interface=wlan0
driver=nl80211
ctrl_interface=/var/run/hostapd
ssid=HisiAP
channel=6
hw_mode=g
ieee80211n=1
ht_capab=[SHORT-GI-20][SHORT-GI-40] [HT40-]
wpa=3
wpa_key_mgmt=WPA-PSK
wpa_pairwise=TKIP CCMP
wpa_passphrase=12345678
```

The hostapd is an open-source code. For details about parameters in the configuration file, search for network resources.

Step 2 Start the hostapd process by running the following shell command:

```
hostapd /etc/Wireless/hostapd.conf &
```

After the command is executed, run the **ps** command to check whether the hostapd process exists. If yes, it works properly, and the SoftAP can be detected by the STA device. If no, increase the hostapd print level and find out the cause from the logs. For example:

```
hostapd /etc/Wireless/hostapd.conf –ddd &
```

----End

2.2.4 Enabling udhcpd

Run the following shell commands:

```
ifconfig wlan0 192.168.1.1
udhcpd –fS /etc/udhcpd.conf
```

Ensure that **/etc/udhcpd.conf** exists, and the configured network segment is 192.168.1.*x*. After the preceding commands are executed, the STA device can connect to the SoftAP.

2.2.5 Sharing the Network

In the network sharing scenario, the board connects to the extranet over the Ethernet in the uplink direction, and it shares the network with STA devices over the SoftAP in the downlink direction so that STA devices connected to the SoftAP can also access the extranet.

Run the **shell** command:

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

```
iptables -t nat -A POSTROUTING -o eth0 -j ASQUERADE
```

2.3 Configuring the Country or Region to Which the Wi-Fi Device Applies

The frequency range varies according to the country or region. For example, for the 2.4 GHz frequency band, the US supports channels 1 to 11, China and Europe support channels 1 to 13, and Japan supports channels 1 to 14. The situation is similar for the 5 GHz frequency band. The Wi-Fi device needs to be configured based on the country or region in which the device is to be launched.

The configuration method varies according to the Wi-Fi device.

- To use RTL8188EUS in the US, add the parameter **rtw_channel_plan=0x22** as follows when loading the driver:

  ```
  insmod rtl8188eu.ko rtw_channel_plan=0x22
  ```

- To use MT7601U in the US, modify the driver configuration file (for example, **/etc/Wireless/RT2870STA/RT2870STA.dat**) as follows:

  ```
  CountryRegion=0
  CountryCode=US
  ```

This document does not describe all the configuration methods. For details, consult the Wi-Fi chip vendor.

3 Tests

3.1 Function Tests

3.1.1 STA Mode

The function tests for the STA mode include but are not limited to the following:

- Change the SSID of the AP, and scan for and connect to the AP by using a board.
- Change the channel of the AP and scan for and connect to the AP by using a board.
- Change the wireless protocol of the AP, and scan for and connect to the AP by using a board.
- Change the encryption mode of the AP, and scan for and connect to the AP by using a board.
- Connect to different types of APs to test the compatibility.
- Connect to the AP and run the **ping** command continuously for 12 hours.
- Increase interference gradually to test the connection stability.

3.1.2 SoftAP Mode

The function tests for the SoftAP mode include but are not limited to the following:

- Change the SSID of the SoftAP, and search for and connect to the SoftAP by using a mobile phone.
- Change the channel of the SoftAP, and search for and connect to the SoftAP by using a mobile phone.
- Change the encryption mode of the SoftAP, and search for and connect to the SoftAP by using a mobile phone.
- Connect to the SoftAP by using different mobile phones to test the compatibility.
- Connect a mobile phone to the SoftAP and run the **ping** command continuously for 12 hours.
- Increase interference gradually to test the connection stability.

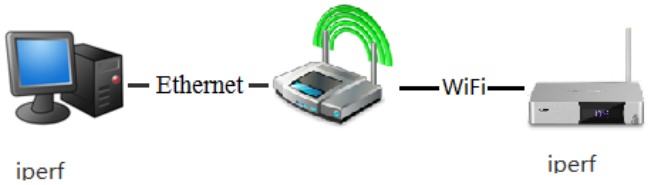
3.2 Throughput Tests

The throughput tests show the Wi-Fi performance and are widely used and proved by chip vendors, module vendors, and Wi-Fi device vendors. The most frequently used throughput tool is iperf.

The iperf tool is integrated in the SDK. On the Linux platform, you need to run **make menuconfig**, choose **Rootfs** > **Board Tools Config**, and set **Iperf Support** to **y** to enable the iperf compilation. This tool is compiled into the image by default on the Android platform.

Figure 3-1 shows the test environment. A PC connects to the AP by using a cable, a board connects to the AP by using the Wi-Fi, and the PC and the board can pin each other successfully. The iperf tool is installed on both the PC and the board. Assume that the IP address for the PC is 192.168.1.100, and that for the board is 192.168.1.101.

Figure 3-1 Networking for throughput tests



3.2.2 TCP Transmit Throughput Test

To test the transmit throughput, perform the following steps:

Step 1 Go to the directory of the iperf tool in the command-line interface (CLI) on the PC and run the following command:

```
iperf -s
```

Step 2 Go to the directory of the iperf tool by using shell on the board and run the following command:

```
iperf -c 192.168.1.100 -t 10 -i 1
```

Figure 3-2 Transmit throughput test example

```
# iperf -c 192.168.1.100 -t 10 -i 1
Client connecting to 192.168.1.100, TCP port 5001
TCP window size:  512 KByte (default)
[  3] local 192.168.1.101 port 44753 connected with 192.168.1.100 port 5001
[  3]  0.0- 1.0 sec  8.40 MBytes  70.5 Mbits/sec
[  3]  1.0- 2.0 sec  8.57 MBytes  71.9 Mbits/sec
[  3]  2.0- 3.0 sec  8.65 MBytes  72.5 Mbits/sec
[  3]  3.0- 4.0 sec  8.52 MBytes  71.4 Mbits/sec
[  3]  4.0- 5.0 sec  8.57 MBytes  71.9 Mbits/sec
[  3]  5.0- 6.0 sec  8.52 MBytes  71.4 Mbits/sec
[  3]  6.0- 7.0 sec  8.59 MBytes  72.1 Mbits/sec
[  3]  7.0- 8.0 sec  8.52 MBytes  71.5 Mbits/sec
[  3]  8.0- 9.0 sec  8.72 MBytes  73.1 Mbits/sec
[  3]  9.0-10.0 sec  8.62 MBytes  72.4 Mbits/sec
[  3]  0.0-10.0 sec  85.7 MBytes  71.6 Mbits/sec
```

where:

- **iperf -s**: Starts the server end.
- **iperf –c 192.168.1.100**: Starts the client and connects to 192.168.1.100.
- **-t 10**: Tests the throughput for 10 seconds.
- **-i 1**: Displays the result every other second.

The displayed test result "0.0-10.0 sec 85.7 MBytes 71.6 Mbit/sec" indicates that the average throughput is 71.6 Mbit/s.

----End

3.2.3 TCP Receive Throughput Test

To test the receive throughput, perform the following steps:

Step 1 Go to the directory of the iperf tool by using shell on the board and run the following command:

```
iperf –s
```

Step 2 Go to the directory of the iperf tool in the CLI on the PC and run the following command:

```
iperf –c 192.168.1.101 –t 10 –i 1 –w 1M
```

Figure 3-3 Receive throughput test example

```
# iperf -s -i 1
Server listening on TCP port 5001
TCP window size: 1.00 MByte (default)
GetDesiredTssiAndCurrentTssi: BBP TSSI INFO is not ready. (BbpR47 = 0x94)
RT5390_AsicTxAlcGetAutoAgcOffset: Incorrect desired TSSI or current TSSI
[  4] local 192.168.1.101 port 5001 connected with 192.168.1.100 port 59938
[  4]  0.0- 1.0 sec  10.1 MBytes  85.0 Mbits/sec
[  4]  1.0- 2.0 sec  10.3 MBytes  86.5 Mbits/sec
[  4]  2.0- 3.0 sec  10.1 MBytes  84.4 Mbits/sec
[  4]  3.0- 4.0 sec  9.86 MBytes  82.8 Mbits/sec
[  4]  4.0- 5.0 sec  9.83 MBytes  82.4 Mbits/sec
[  4]  5.0- 6.0 sec  9.92 MBytes  83.3 Mbits/sec
[  4]  6.0- 7.0 sec  9.33 MBytes  78.3 Mbits/sec
[  4]  7.0- 8.0 sec  9.99 MBytes  83.8 Mbits/sec
[  4]  8.0- 9.0 sec  9.70 MBytes  81.4 Mbits/sec
[  4]  9.0-10.0 sec  10.0 MBytes  84.2 Mbits/sec
[  4]  0.0-10.1 sec   100 MBytes  83.3 Mbits/sec
```

The iperf tool can also be used to perform the User Datagram Protocol (UDP) test. The speed of a single UDP thread is limited on some PCs, and therefore multiple threads are required.

The throughput tests for the SoftAP are similar.

CAUTION

The speed of some PCs is affected by the installed software. Ensure that the PC speed is not affected. The 802.11n protocol cannot be used in WEP safe mode, and therefore the speed is low, typically over 20 Mbit/s.

----End

3.2.4 UDP Transmit Throughput Test

To test the transmit throughput, perform the following steps:

Step 1 Go to the directory of the iperf tool in the CLI on the PC and run the following command:

```
iperf –s -u –l 32k
```

Step 2 Go to the directory of the iperf tool by using shell on the board and run the following command:

```
iperf –c 192.168.1.100 –u –t 10 –i 1 –l 32k –b 100M
```

----End

3.2.5 UDP Receive Throughput Test

To test the receive throughput, perform the following steps:

Step 1 Go to the directory of the iperf tool by using shell on the board and run the following command:

```
iperf –s -u
```

Step 2 Go to the directory of the iperf tool in the CLI on the PC and run the following command:

```
iperf –c 192.168.1.101 –u –t 10 –i 1 –l 32k –b 100M
```

----End

3.3 RF Specifications Test

The throughput tests reflect the Wi-Fi performance and are mandatory during product development. Some companies also conduct the RF specifications test, which accurately verifies whether the Wi-Fi RF meets specifications. The RF specifications test is mandatory during module production. Therefore, if a Wi-Fi module is used, this test is optional. However, the Wi-Fi RF performance may be affected due to board interference and unclean GND traces during hardware design, you are advised to conduct this test if possible.

The RF specifications include the following: receive sensitivity, power suppression of the adjacent channel, transmit power, error tolerance of the transmit carrier frequency, packet loss rate, error vector magnitude (EVM), transmit adjacent channel power ratio (ACPR), and receive ACPR.

The test tools include spectrum analyzer, power measurer, and network analyzer.

For details about the test methods, see the instructions of the test tools.

3.4 Antenna Tests

The antenna is an important factor that affects the Wi-Fi performance. Therefore, the antenna specifications tests are mandatory during product development. STBs are tested during the antenna performance tests, and the test environment of the antenna vendor can be used.

The antenna specifications include but are not restricted to the following:

- Efficiency: ratio of the power radiated by the antenna (power that is converted into the electromagnetic wave) to the active power input to the antenna. It is a major specification of the antenna.
- Gain: ratio of the signal power density of the actual antenna to that of an ideal radiating element at the same spatial point. It is another key specification of the antenna and can be used to determine the antenna performance by working with the directions.
- Standing wave ratio (SWR): It reflects the antenna match conditions and ensures that signals can enter the antenna. It is an important specification for ensuring the antenna efficiency. Specifications of the same type include return loss, reflection coefficient, and input impedance.

4 Precautions for Hardware Design

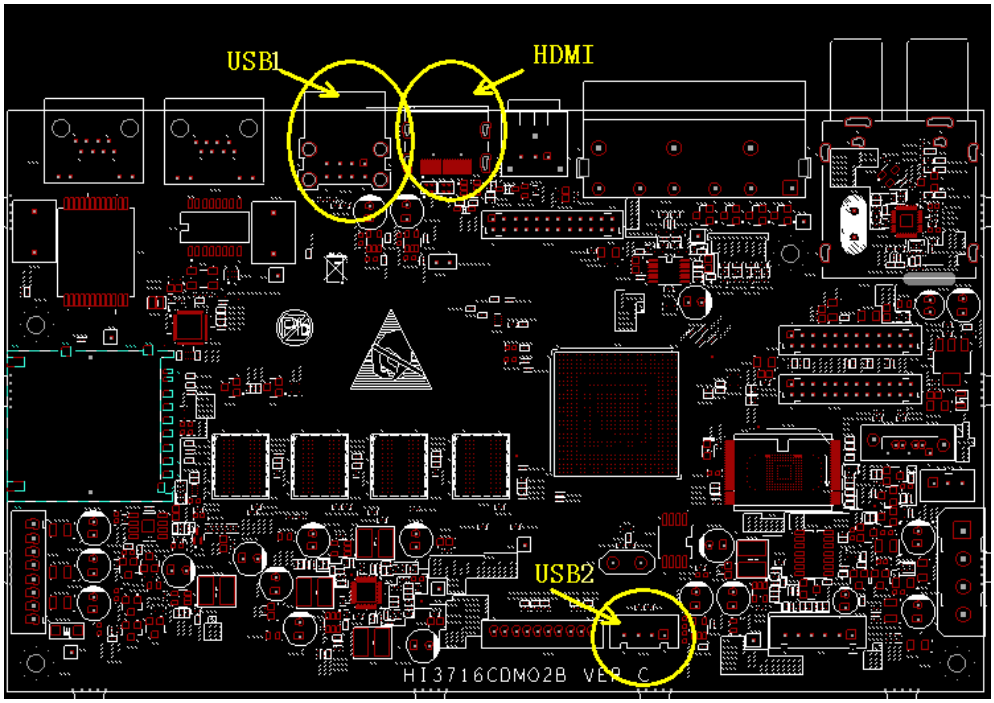
4.1 Interference from the HDMI Interface

When the high-definition multimedia interface (HDMI) uses the 74. 2 MHz frequency, its 33 times frequency falls within the 2.4 GHz frequency band of the Wi-Fi, which seriously interferes with the Wi-Fi signals. If the HDMI interface uses the 148.5 MHz frequency, its 16 times frequency does not fall within the Wi-Fi frequency band. However, it also interferes with the Wi-Fi signals to some extent as frequency isolation is poor.

As the soldering point of the HDMI connector cannot be shielded, if the space between the HDMI interface on the board and the Wi-Fi module is less than 5 cm, the HDMI output will interfere with Wi-Fi signals, which results in Wi-Fi connection failures and throughput decrease.

Therefore, the Wi-Fi module needs to be placed far away from the HDMI interface to avoid interference. See Figure 4-1. USB2, instead of USB1, is recommended for connecting the Wi-Fi module.

Figure 4-1 Hardware design for the HDMI and Wi-Fi module



4.2 Interference from the System Clock

If the system clock works at about 600 MHz, its multiplied frequency may fall between 2.4 GHz and 2.5 GHz, which interferes with Wi-Fi signals and results in Wi-Fi connection failures and throughput decrease.

The solution is described as follows:

- Check whether the interference affects the Wi-Fi module through the power or GND on the PCB. Put the contact probe (cannot receive space radiation signals) of the spectrum analyzer near the power and GND of the Wi-Fi module. Compare the noise strength with that in the nearby non-frequency-multiplication area. If the difference is less than 5 dB, the interference can be ignored. Otherwise, isolate the noises by connecting electromagnetic interface (EMI) beads in series on the power and GND traces.
- Check whether the interference affects the Wi-Fi module and antenna through space radiation. Connect the spectrum analyzer to a near-field probe (for detecting near-field radiation) and scan the area around the Wi-Fi module. If there are noises at the multiplied frequency, noises exist. If the Wi-Fi module is affected by radiation interference, you are advised to shield the Wi-Fi module. If the Wi-Fi antenna is affected by radiation interference, put the Wi-Fi antenna to a position inside the STB with less interference. If the STB is not big enough to house the antenna, an external antenna is recommended.
- Adjust the system clock frequency (for example, from 600 MHz to 590 MHz) so that its multiplied frequencies do not fall within the Wi-Fi frequency band.

5 Precautions for Software Design

5.1 Device Detection

5.1.1 Impact on Device Detection

Compared with the wired network, the Wi-Fi network has higher packet loss rate, which varies according to the Wi-Fi signal strength, Wi-Fi module performance, and environment interference. The interference is most severe with very high packet loss rate when the adjacent frequency (channel) has a large number of data services. In this case, a device may fail to be detected due to the loss of multicast packets.

5.1.2 Workaround

- Analyze the Wi-Fi performance.
- Shorten the interval for sending multicast packets, whereas prolong the timeout period.
- Report the connection and disconnection notification messages by using TCP short links, and increase the timeout period to 20 to 30 seconds by taking the loss of TCP handshake packets into consideration.
- Minimize the Wi-Fi application scope. Considering the impact on services exerted by IP layer packet loss, use the wired network in priority during device detection in the implementation of the software solution.

5.2 UDP Services

5.2.1 Main UDP Services

The UDP protocol is used in real-time streaming services, such as mirror, mircast, and video phone. These services have high requirements on real-time data. Long delay affects service experience. However, for the media playback that is insusceptible to delay and buffering, the UDP is not appropriate for data transmission.

5.2.2 Impact on UDP Services

Unpredictable packet loss easily occurs in the Wi-Fi network due to interference. The UDP protocol has no retransmission mechanism, and therefore lost packets are not retransmitted at the transmit end. In addition, the disorder issue of received UDP packets at the receive end is

more severe than that in the wired network. These two factors affect user experience of UDP services.

Take the Miracast service as an example. The packet loss is described as follows according to statistics:

- The packet loss rate at the network adapter is more than the RTP packet loss rate at the application layer, indicating that not only UDP packets carried with RTP packets are discarded, other packets are also discarded, such as the P2P connection keep-alive packet.
- There is no linear relationship between the packet loss rate counted by the network adapter and the RTP packet loss rate at the application layer. Therefore, the packet loss rate of the network adapter cannot indicate the packet loss rate of the RTP packets, or the Miracast performance.
- The display effect is excellent when the RTP packet loss rate is lower than 1%, and it is acceptable when the packet loss rate falls between 1% and 2%. The display effect is poor when the packet loss rate exceeds 6%.

5.2.3 Workaround

The solutions to the packet loss issue are described as follows:

- During frame data processing, erratic display is worse than intermittence. Assume that a data frame can be displayed as a complete image, and a data frame transmitted to the peer over the UDP can be multiple UDP packets. When a large number of packets in a data frame are lost, the priority is to discard the data frame completely instead of displaying it. This is because compared with the process of reducing the frame rate by discarding the data frame, user experience is poorer when many error packets are generated or mosaic occurs as the data frame with high packet loss rate is decoded.
- Statistics on the packet loss rate is almost meaningless, because when there is interference to Wi-Fi signals, packets may not be sent or received. Therefore, to ensure the display of the last media data frame when packet loss occurs is the most reliable method.
- Wi-Fi packet loss can be reduced by decreasing the load of IP packets, which can be implemented by reducing the video definition, such as reducing the frame rate, image quality, or bit rate.

In a word, when data transmission is implemented over the UDP, the policies for data packet loss must be fully considered. You need to take comprehensive consideration based on different software application scenarios, no matter the policy is to retransmit the packets or reduce the image quality.

5.3 TCP Services

5.3.1 Main TCP Services

The TCP protocol typically applies to services such as multimedia playback and network browse. Its main application layer protocol is HTTP, and the services include NFS, FTP, SAMBA, and DLNA.

5.3.2 Impact on TCP Services

The TCP protocol has the retransmission mechanism, that is, packets are retransmitted after they are lost. When there is interference to Wi-Fi signals and the packet loss rate is high, the

TCP handshake time increases, data transmission or reception duration increases, and TCP one-way transmission (read only or write only) may occur as well. Based on tests of the Mirror and Miracast services, the TCP one-way transmission may last for five seconds, and the timeout period may be 20 to 30 seconds when the interference is serious. Long timeout duration may cause the change of the application layer policy, and you cannot identify whether the network is disconnected.

The following are the features of TCP services in the Wi-Fi network:

- When the Wi-Fi working environment is complex, the packet handshake time and confirmation time may be very long.
- The packet size is irrelevant to the issue.
- The TCP short link commands are not easily lost.
- You cannot determine whether the network of the peer end is unavailable if you exit the TCP connection by using timeout.

5.3.3 Workaround

- Use short links if possible because asynchronization of state machines does not occur when short links are used.
- The timeout period for the Wi-Fi module cannot be configured with that for other modules because the Wi-Fi module timeout period is longer, which cannot be accepted if it is used for other modules.
- Do not use TCP packets for services that have high requirements on real-time data.
- Use standard protocols, software, and components if possible.
- Review the design of modules that are exited in timeout mode because the timeout does not indicate that the network is actually disconnected. The timeout period can be prolonged as required.

5.4 Comparison Between the Wi-Fi Direct Mode and Station Mode

NOTE

The Wi-Fi network has various types: Wi-Fi Direct, station-AP, and ad-hoc. In the Wi-Fi Direct network, the two peers are equivalent and can be directly connected. Station-AP is commonly used and one peer must be the AP. Ad-hoc is rarely used.

During service design, data channels can use different types of Wi-Fi networks. You need to choose the most appropriate network type based on the requirements on service functions and performance.

The interference issues of the Wi-Fi Direct and station-AP networks are the same. Therefore, the Wi-Fi network stability is poor due to interference. In addition, the Wi-Fi Direct network takes 3 seconds more to establish the connection than the station-AP network. The station-AP network attempts to reestablish the connection when it is disconnected, and services continue to run when the network is successfully reconnected. However, the Wi-Fi Direct network does not reconnect after it is disconnected. You are advised not to use the Wi-Fi network in scenarios that have high requirements on user experience. If there is no other choice, minimize interference from the adjacent frequency.

5.5 Comparison Between the 5 GHz Frequency Band and 2.4 GHz Frequency Band

There are 13 channels (China) on the 2.4 GHz frequency band, and the channel center frequency interval is 5 MHz. There are more than 20 channels on the 5 GHz frequency band, and the channel center frequency interval is 20 MHz. The interference to the 2.4 GHz frequency band is more serious than that to the 5 GHz frequency band because ZigBee, Bluetooth, and the microwave oven are working on the 2.4 GHz frequency band, which adds more non-Wi-Fi interference to the 2.4 GHz frequency band. Therefore, the 5 GHz frequency band is a better choice for services when interference is taken into consideration. However, if through-wall is considered, the 2.4 GHz frequency band is recommended because it has better through-wall capability.

5.6 Enabling/Disabling the Wi-Fi

Enabling and disabling the Wi-Fi both take a long time. Enabling the Wi-Fi typically requires 1 to 3 seconds. The Wi-Fi is enabled/disabled only when the state change notification is received, not when the API is returned. For example, on the Android platform, after WifiManager.setWifiEnabled() is called, you need to wait for the WifiManager.WIFI_STATE_CHANGED_ACTION broadcast.

```
   private final BroadcastReceiver mReceiver = new BroadcastReceiver() {
      @Override
      public void onReceive(Context context, Intent intent) {
         String action = intent.getAction();
         if (WifiManager.WIFI_STATE_CHANGED_ACTION.equals(action)) {
            handleWifiStateChanged(intent.getIntExtra(
                  WifiManager.EXTRA_WIFI_STATE,
WifiManager.WIFI_STATE_UNKNOWN));
         }
      }
   }
   private void handleWifiStateChanged(int state) {
      switch (state) {
         case WifiManager.WIFI_STATE_ENABLING:
            break;
         case WifiManager.WIFI_STATE_ENABLED:
            break;
         case WifiManager.WIFI_STATE_DISABLING:
            break;
         case WifiManager.WIFI_STATE_DISABLED:
            break;
         default:
            break;
      }
```

The enable or disable process is complete when the state is WIFI_STATE_ENABLED or WIFI_STATE_DISABLED. Before that, you cannot operate the Wi-Fi.

6 Solutions to Common Issues

6.1 Fault Location Tools

6.1.1 iw Tools

The iw tools include iwconfig, iwlist, and iwpriv, which are used on Linux for configuring the Wi-Fi network and checking the network status.

- iwconfig
  - Check whether the Wi-Fi driver is successfully initialized.

    ```
    # iwconfig
    lo       no wireless extensions.

    eth0     no wireless extensions.

    eth1     no wireless extensions.

    wlan0    unassociated  Nickname:"<WIFI@REALTEK>"
             Mode:Auto  Frequency=2.412 GHz  Access Point: Not-
    Associated
             Sensitivity:0/0
             Retry:off   RTS thr:off   Fragment thr:off
             Encryption key:off
             Power Management:off
             Link Quality=0/100  Signal level=0 dBm  Noise level=0 dBm
             Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
             Tx excessive retries:0  Invalid misc:0   Missed beacon:0
    ```

    If **wlan0** is displayed, the driver is initialized successfully; otherwise, the initialization fails.
  - Check the status of the Wi-Fi network.

    ```
    # iwconfig wlan0
    wlan0    IEEE 802.11bgn  ESSID:"B21"  Nickname:"<WIFI@REALTEK>"
    ```

```
        Mode:Managed  Frequency:2.437 GHz  Access Point:
8C:21:0A:A5:CD:B2
        Bit Rate:150 Mb/s   Sensitivity:0/0
        Retry:off   RTS thr:off   Fragment thr:off
        Encryption key:****-****-****-****-****-****-****-****
Security mode:open
        Power Management:off
        Link Quality=88/100  Signal level=-45 dBm  Noise level=0
dBm
        Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
        Tx excessive retries:0  Invalid misc:0   Missed beacon:0
```

The information includes the supported 80211 protocol, AP SSID, STA/ad-hoc/AP mode, AP frequency, AP MAC address, current rate, encryption mode, and signal quality.

- iwlist

  Scan for the AP.

```
# iwlist wlan0 scan
wlan0     Scan completed :
        Cell 01 -  Address: 8C:21:0A:A5:CD:B2
                ESSID:"B21"
                Protocol:IEEE 802.11bgn
                Mode:Master
                Frequency:2.437 GHz (Channel 6)
                Encryption key:on
                Bit Rates:300 Mb/s

Extra:wpa_ie=dd1a0050f20101000050f20202000050f2040050f20201000050f202
                IE: WPA Version 1
                   Group Cipher : TKIP
                   Pairwise Ciphers (2) : CCMP TKIP
                   Authentication Suites (1) : PSK

Extra:rsn_ie=30180100000fac020200000fac04000fac020100000fac020000
                IE: IEEE 802.11i/WPA2 Version 1
                   Group Cipher : TKIP
                   Pairwise Ciphers (2) : CCMP TKIP
                   Authentication Suites (1) : PSK
                Quality=0/100  Signal level=-47 dBm
```

  For details about the scan result, see section 2.1.3 "Scanning for APs."

- iwpriv

  iwpriv directly reads or configures parameters from the driver. The supported parameters vary according to the Wi-Fi device. For details about the meaning of each parameter, consult the vendor.

  Check the parameters supported by the Wi-Fi:

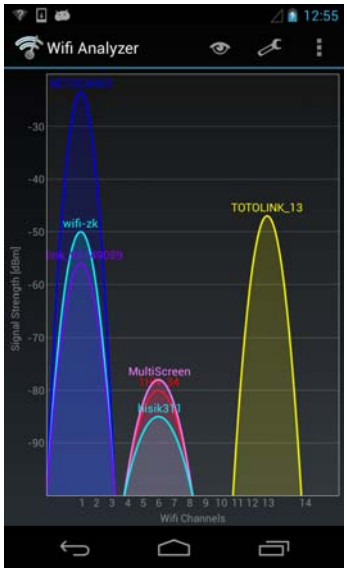
```
# iwpriv wlan0
```

```
wlan0     Available private ioctls :
          write           (8BE0) : set 2047 char  & get   0
          read            (8BE1) : set 2047 char  & get  16 char
          driver_ext       (8BE2) : set   0      & get   0
          mp_ioctl         (8BE3) : set   0      & get   0
          apinfo           (8BE4) : set   1 int   & get   0
          setpid           (8BE5) : set   2 int   & get   0
          wps_start        (8BE6) : set   1 int   & get   0
          get_sensitivity  (8BE7) : set   1 int   & get   0
          wps_prob_req_ie  (8BE8) : set   1 int   & get   0
          wps_assoc_req_ie (8BE9) : set   1 int   & get   0
          channel_plan      (8BEA) : set   1 int   & get   0
          dbg             (8BEB) : set   2 int   & get   0
          rfw             (8BEC) : set   3 int   & get   0
          rfr             (8BED) : set   2 int   & get  16 char
          p2p_set          (8BF0) : set  64 char  & get   0
          p2p_get          (8BF1) : set  64 char  & get  64 char
          p2p_get2         (8BF2) : set  64 char  & get  16 char
          NULL            (8BF3) : set 128 char  & get   0
          tdls            (8BF4) : set  64 char  & get   0
          tdls_get         (8BF5) : set  64 char  & get  64 char
          pm_set           (8BF6) : set  64 char  & get   0
          rereg_nd_name     (8BF8) : set  16 char  & get   0
          efuse_set        (8BFA) : set 1024 char  & get   0
          efuse_get        (8BFB) : set 128 char  & get 2047 char
```

6.1.2 Wi-Fi Analyzer

The Wi-Fi analyzer is an Android application that collects statistics on the number of APs around, AP SSID, channel, and signal strength. It can be used to analyze the interference, as shown in Figure 6-1.

Figure 6-1 Wi-Fi analyzer



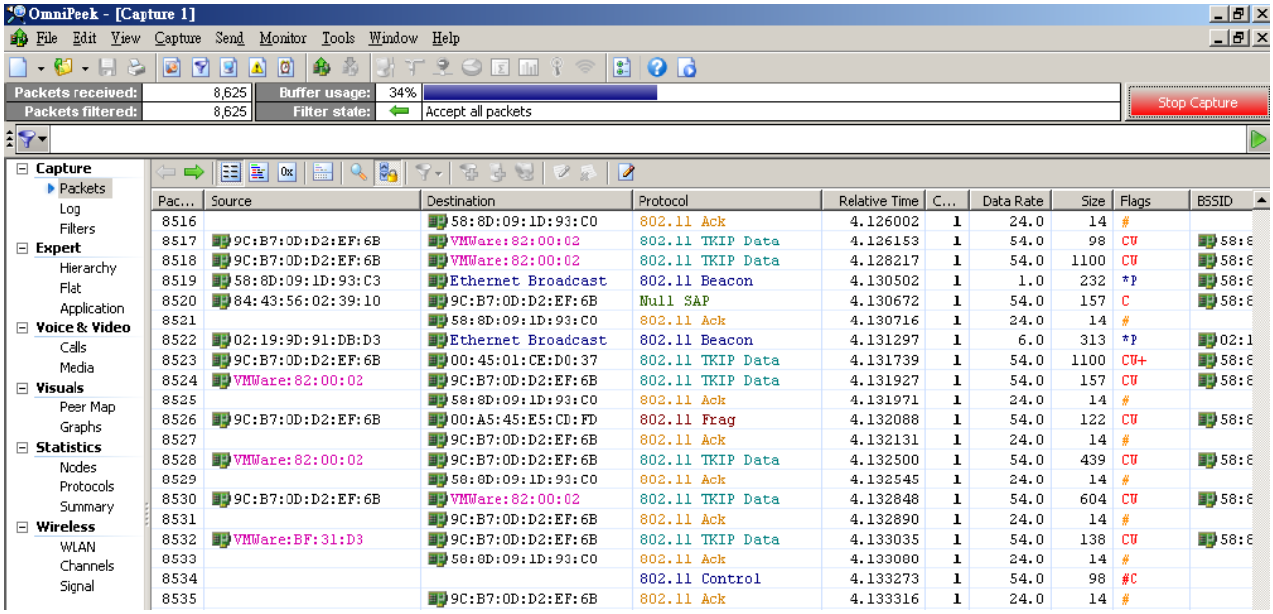
In Figure 6-1, there are three APs on channel 1, three APs on channel 6, and one AP on channel 13. A greater AP peak value indicates stronger signal strength. The span indicates the frequency range. For example, APs on channel 1 interfere with channel 2 and channel 3.

The Wi-Fi analyzer cannot obtain the amount of data transmitted in each channel, and therefore it cannot show the actual interference but only analysis based on the probability. Typically the more APs in a channel, the greater the interference.

6.1.3 OmniPeek

The OmniPeek is a network packet-capture tool which works with the DWA160 wireless network adapter. It captures packets transmitted over the Wi-Fi network and is used for analyzing the Wi-Fi protocols.

Figure 6-2 OmniPeek capturing packets



In the captured data, you can find the content of each 80211 packet.

6.1.4 Logcat

The logcat is a tool for obtaining logs on the Android platform. The Wi-Fi settings, framework layer, HAL, wpa_supplicant, and hostapd all need to display logs by using the logcat when a fault occurs.

6.2 FAQs

6.2.1 What Do I Do If the Wi-Fi Driver Fails to Be Loaded?

Problem Description

When the driver is being loaded, a message is displayed indicating that the file format or magic number is incorrect.

Cause Analysis

The Linux kernel version is incorrect, or the cross compilation environment is incorrectly configured in the **Makefile** of the driver so that the Wi-Fi module is compiled into a module for the PC platform.

Solution

Modify the driver **Makefile**, set **ARCH**, **CROSS_COMPILE**, and the Linux kernel path correctly, and then recompile the driver.

Take RTL8188EUS as an example:

```
ifeq ($(CONFIG_PLATFORM_HISILICON), y)
EXTRA_CFLAGS += -DCONFIG_LITTLE_ENDIAN -DCONFIG_PLATFORM_ANDROID -
DCONFIG_PLATFORM_SHUTTLE
ARCH := arm
ifeq ($(CROSS_COMPILE),)
CROSS_COMPILE = arm-hisiv200-linux-
endif
MODULE_NAME := rtl8188eu
ifeq ($(KSRC),)
KSRC := ../../../../../../kernel/linux-3.4.y
endif
endif
```

6.2.2 What Do I Do If a Message Similar to "usb 1-2.1: USB disconnect, address 3" Is Displayed?

Problem Description

When the Wi-Fi is used, the serial port outputs a message similar to "usb 1-2.1: USB disconnect, address 3", and the Wi-Fi cannot be used.

Cause Analysis

The USB port cannot identify the Wi-Fi device. This log is displayed by the USB driver. This issue occurs because the Wi-Fi device is damaged or the power supply of the USB port is insufficient.

Solution

Replace the Wi-Fi device to check whether the module is faulty. Then check whether the USB voltage and current meet the requirement of the Wi-Fi module.

6.2.3 What Do I Do If the Wi-Fi Throughput Is Low?

Problem Description

The Wi-Fi throughput is low when the distance between the board and the AP is less than 1 m.

Solution

Low Wi-Fi throughput can be caused for many reasons. Perform the following steps to locate the fault:

Step 1 Check whether the 11n mode is set on the AP configuration page.

Step 2 Check whether the WEP encryption mode is set on the AP configuration page. Note that WEP encryption does not support the 11n mode.

Step 3 Check whether the protocol is set to WPA-PSK/WPA2-PSK on the AP configuration page, and whether the encryption algorithm is set to TKIP.

Step 4 When testing the throughput, run the shell command **iwconfig wlan0** and check whether the bit rate is greater than or equal to 150 Mbit/s. If the MTK Wi-Fi device is used and the bit rate is 72 Mbit/s, set **HT_BW** to **1** in the **RT2870STA.dat** file.

Step 5 Check whether the AP and the board are on the same plane. If no, the Wi-Fi performance is affected.

Step 6 If the AP has two antennas, check whether the two antennas and the antenna of the board form a straight line. If yes, the Wi-Fi performance is affected.

Step 7 Check whether there are many APs around that cause interference to the Wi-Fi device by using the Wi-Fi analyzer.

Step 8 Check whether the Wi-Fi antenna is properly installed.

Step 9 Check whether the distance between the Wi-Fi device and the HDMI interface is less than 5 cm. If yes, the HDMI interface may cause great interference to the Wi-Fi device.

Step 10 Check whether the GND trace of the Wi-Fi module is clean.

Step 11 Connect the board and the PC using the wired network, test the throughput, and check whether the PC speed rate is limited.

Step 12 Restore the AP to factory settings.

----End

6.2.4 What Do I Do If the AP Cannot Be Detected?

Problem Description

The AP is located near the board but cannot be detected by the board.

Cause Analysis

The cause may be either of the following:

- The country code in the driver is incorrect. As a result, the AP channel is not supported.
- The GND trace of the Wi-Fi module is not clean or interference from the HDMI interface exists.

Solution

The solution is described as follows based on the cause:

- Set the country code correctly and recompile and burn the driver.
- Ask the module vendor to check the Wi-Fi hardware design.

6.2.5 What Do I Do If the AP Cannot Be Connected?

Problem Description

The AP can be detected but cannot be connected.

Solution

This issue can be caused by many reasons. Perform the following steps to locate the fault:

Step 1 Check whether the password is set correctly.

Step 2 Check whether the antenna of the board is properly installed.

Step 3 Check whether the Wi-Fi is close to the HDMI interface.

Step 4 Capture packets by using the OmniPeek, and analyze the data packets to check whether the packets do not comply with the protocols.

----End

6.2.6 What Do I Do If the Wi-Fi Cannot Be Enabled?

Problem Description

The Wi-Fi cannot be enabled on the UI on the Android platform.

Solution

Enable the logcat and analyze the logcat display information and driver logs.

- "Cannot find supported device" indicates that the Wi-Fi module is damaged or the Wi-Fi device is not supported by the current version.
- "Failed to load driver!" indicates that the driver is not found, the driver fails to be initialized, or the driver has been loaded but not unloaded.
- The driver has been loaded, but error information (the specific error information varies according to the Wi-Fi chip) is displayed during initialization, and wlan0 cannot be found when **iwconfig** is executed. This issue arises possibly because the driver is mismatched, the memory fails to be allocated, or the Wi-Fi module is faulty.
- "Supplicant not running, cannot connect" indicates that the wpa_supplicant process fails to be started. In this case, run the **ps** command to check whether the wpa_supplicant process is running. If no, the wpa_supplicant service parameter in **init.rc** is incorrect. Modify the parameter and then compile and burn the kernel image.
- "Unable to open connection to supplicant on xxx" indicates that the HAL and the wpa_supplicant process fail to establish a socket connection. This issue arises possibly because the wpa_supplicant service parameter in **init.rc** or the **ctrl_interface** parameter in **wpa_supplicant.conf** is incorrect. **ctrl_interface** needs to be set to **wlan0**.
- The Wi-Fi is automatically disabled and enabled repeatedly on the UI. This issue arises because the driver does not support some commands of the wpa_supplicant process. Check whether the driver supports the following commands: SCAN-ACTIVE, **SCAN-PASSIVE**, **RSSI, LINKSPEED**, **BTCOEXSCAN-START**, **BTCOEXSCAN-STOP**, **BTCOEXMODE**, **MACADDR**, **GETBAND**, and **SETBAND**.

6.2.7 What Do I Do If a Specific AP Cannot Be Connected?

Problem Description

The Wi-Fi device cannot connect to a specific AP, or it is very hard to connect the Wi-Fi device to the AP, or the Wi-Fi device is easily disconnected from the AP after being connected.

The Wi-Fi device can connect to other APs properly, and other Wi-Fi devices and boards can connect to the specific AP properly.

Cause Analysis

This issue arises possibly because the Wi-Fi module is faulty. Frequency offset occurs when the AP is frequently used. If frequency offset also occurs on the Wi-Fi module or the RF quality of some modules is poor, this issue is caused.

Solution

Restore the AP to factory settings. If the problem persists, replace the Wi-Fi module on the board.

6.2.8 How Do I Resolve Issues Related to Standby and Wakeup?

Problem Description

After the Wi-Fi is enabled, the following issues may arise during standby and wakeup:

- The kernel crashes when the Wi-Fi device enters the standby mode.
- The Wi-Fi device cannot enter the standby mode.
- The kernel crashes when the Wi-Fi device is woken up from the standby mode.
- The Wi-Fi device cannot be used after wakeup.
- The Wi-Fi device cannot enter the standby mode immediately after it connects to an AP.

Cause Analysis

The Android platform is designed for mobile phones and tablets. The Wi-Fi standby mechanism does not apply to the STB. In addition, in the original Android design, the Wi-Fi device connects to the SDIO interface and is not powered off during standby. However, the Wi-Fi device connected to the USB port is powered off during standby.

Solution

It is recommended that the Wi-Fi be disabled before standby and enabled after wakeup on both the Linux and Android platforms.

CAUTION

On the Android platform, if the Wi-Fi device has connected to an AP, Android applies for a 60-second wakelock when the Wi-Fi is disabled so that the device has time to connect to the mobile WAN. This ensures that network data packets can still be received during standby. Therefore, the Wi-Fi device enters standby mode 60 seconds after the Wi-Fi is disabled. In this case, you need to comment out **mCm.requestNetworkTransitionWakelock(TAG)** in **WifiService**.

6.2.9 What Do I Do If Other Wi-Fi Modules Cannot Connect to the Wi-Fi Direct Network After MT7601U Connects to the Wi-Fi Direct Network?

Problem Description

After the MT7601U module is inserted to the board and connect to the mobile phone using the Wi-Fi Direct or Miracast service, the Wi-Fi Direct and Miracast connections always fail when the MT761U is replaced with another Wi-Fi module, even if the board and mobile phone are restarted.

Cause Analysis

When Wi-Fi Direct (Miracast also uses the Wi-Fi Direct) is enabled on MT7601U, a network port is created and saved on the board after the network is successfully connected. The subsequent connections require this network port, which is not supported by other Wi-Fi modules.

Solution

Do not use other Wi-Fi modules after the MT7601U connects to the Wi-Fi Direct network.

6.2.10 What Do I Do If the Wi-Fi Compilation Fails?

Problem Description

After the Wi-Fi compilation configuration is enabled and the entire project is compiled, the compilation fails and the following error information is displayed:

```
cannot find -lnl-genl
```

or:

```
bison -y -d  -o route/pktloc_syntax.c route/pktloc_syntax.y
route/pktloc_syntax.y:11.9-16: syntax error, unexpected identifier,
expecting string
```

Cause Analysis

The Wi-Fi module needs to use libnl, and Bison and Flex must be installed on the server for compiling libnl. If the second error information is displayed, the Bison version is too early.

Solution

Install Bison 2.4.1 and Flex 2.5.35 or later on the compilation server.

6.2.11 What Do I Do If the Wi-Fi Is Enabled and Disabled Repeatedly on the Android Platform?

Problem Description

On the Android platform, the Wi-Fi driver is loaded by using the CLI and the scan is normal. However, when the Wi-Fi is enabled on the setting UI, it is enabled and disabled repeatedly.

Cause Analysis

When the socket connection between the framework and wpa_supplicant fails to be set up, the framework restarts the wpa_supplicant process and sets up a socket connection again, and this process is repeated for multiple times. This may be caused by the following two reasons:

- The parameters for starting wpa_supplicant in **init.xxx.rc**, especially **-D**, are incorrect. Incorrect parameter values or sequence may cause failures in creating the socket node or initializing wpa_supplicant.
- The Wi-Fi driver does not support some Android commands, which results in the wpa_supplicant initialization failure. The Android commands include but are not limited to the following:
  - SCAN-ACTIVE
  - SCAN-PASSIVE
  - RSSI, LINKSPEED
  - BTCOEXMODE
  - MACADDR

Solution

- If the issue occurs due to the first cause, modify the parameters of wpa_supplicant in **init.xxx.rc** as required.
- If the issue occurs due to the second cause, ask the vendor to modify the code for supporting those commands in the driver.

6.2.12 What Do I Do If Data Is Interrupted After the iperf or Ping Test Has Been Performed for Some Time?

Problem Description

When the board connects to the AP over the Wi-Fi network, iperf data is interrupted or the network cannot be pinged after the iperf or ping test has been performed for several hours.

Cause Analysis

When the Dynamic Host Configuration Protocol (DHCP) of the AP is updated, WifiStateMachine receives the update event and transmits **BTCOEXMODE** to the driver. However, the driver does not support this command, and wpa_supplicant returns the CTRL-EVENT-DRIVER-STATE HUANG event. After receiving this event, WifiStateMachine disables the Wi-Fi and enables it again. Therefore data is interrupted.

Solution

Modify the code for supporting the **BTCOEXMODE** command in the driver. When this command is received, a success code can be returned without processing.

6.2.13 What Do I Do If the SoftAP Throughput Is Low?

Problem Description

There are few APs in the environment. The SoftAP throughput is only 5–6 Mbit/s or a little more than 20 Mbit/s.

Cause Analysis

This issue occurs because the 11n transmission rate is not used. If the throughput is only 5–6 Mbit/s, the 11b rate is used. If the throughput is a little more than 20 Mbit/s, the 11g rate is used. If the 11n rate is used, the throughput can be more than 40 Mbit/s when the bandwidth is 20 MHz and 80 Mbit/s when the bandwidth is 40 MHz. The Wi-Fi SoftAP of Realteck and Atheros is configured by using hostapd, and 11n is not correctly configured in **hostapd.conf**.

Solution

Add the following statement in **hostapd.conf**:

- When there are seven channels or less:

```
hw_mode=g
ieee80211n=1
ht_capab=[SHORT-GT-20][SHORT-GI-40][HT40+]
```

- When there are more than seven channels:

```
hw_mode=g
ieee80211n=1
ht_capab=[SHORT-GT-20][SHORT-GI-40][HT40-]
```

6.2.14 What Do I Do If the Wi-Fi Advanced Settings of the Android Version Do Not Contain the Option for Switching the Frequency Band?

Problem Description

For the Android version, the Wi-Fi chip supports the 2.4 GHz and 5 GHz frequency bands. However, the **Advanced Wi-Fi** menu of Wi-Fi settings does not contain the **Wi-Fi frequency band** option, and therefore the frequency band cannot be switched. Similar option exists in other mobile phones.

Cause Analysis

Whether the **Wi-Fi frequency band** option is available depends on the **config_wifi_dual_band_support** parameter in **frameworks/base/core/res/res/values/config.xml**. This parameter is set to **false** by default because this version is used in multiple Wi-Fi chips that support only the 2.4 GHz frequency

band. If the Wi-Fi chip supports both 2.4 GHz and 5 GHz frequency bands, set the parameter to **true**.

Solution

Set **config_wifi_dual_band_support** in **frameworks/base/core/res/res/values/config.xml** to **true** as follows:

<bool translatable="false" name="config_wifi_dual_band_support">true</bool>