Fibocom
PERFECT WIRELESS EXPERIENCE

RIL Log Analysis Guide

V1.3

Disclaimer

Customers shall design and develop their products according to the information provided in the document. Fibocom shall not be liable for any damages caused by failure to comply with the relevant operations, specifications or rules. Due to product version update or other reasons, Fibocom reserves the rights to modify any information in this document at any time without prior notice and without any responsibility. Unless otherwise specified, all the statements, information and suggestions contained in the document do not constitute any explicit or implicit guarantee.

Copyright Statement

Unless specifically authorized by Fibocom, the recipient of the document shall keep the received documents and information confidential and shall not use it for any purpose other than the implementation and development of this project. Without the prior written permission of the copyright holder, any company or individual is prohibited to excerpt, copy any part of or the entire document, or distribute the document in any form. For any violation of confidentiality obligations, unauthorized use or malicious use of the document and information in other illegal forms, Fibocom has the rights to pursue legal responsibility.

Trademark Statement

Fibocom The trademark is registered and owned by Fibocom Wireless Inc.

Other trademarks, product names, service names, and company names appearing in this document are the properties of their respective owners.

Contact

Website: https://www.fibocom.com

Address: Floor 10-14, Block A, Building 6, Shenzhen International Innovation Valley, Dashiyi Road, Xili Community, Xili Street, Nanshan District, Shenzhen

Tel: 0755-26733555

Safety Instruction

Do not operate wireless communication products in areas where use of radio is not recommended without proper equipment certification. These areas include environment where radio interference may occur, such as flammable and explosive environment, medical equipment, aircraft, or any other equipment that may be subject to any form of radio interference.

Any driver of a vehicle must not operate a wireless communication product while controlling the vehicle. Doing so will reduce the driver's control and operation of the vehicle, posing a safety risk.

The wireless communication product does not guarantee a valid connection under any circumstances, for example, when the (U)SIM is in arrears or invalid. In case of emergency, use the emergency call function in power-on state, and make sure the equipment is in an area with sufficient signal strength.

Contents

Applicable Models

No.	Applicable Model	Description
1	L61x&LC61x&LG61x&MC61x&MG61x series	L610, LC610N, LG610, MC610, MC615, MC617, MC619
2	MC11x&MG110 series	MC116, MG110
3	NL668&NL668T series	NL668, NL668T
4	L71x series	L716, L718
5	MC919 series	MC919
6	MC66x&MG66x series	MC660, MC661, MC665, MC667, MC669, MG661, MG662
7	FG132 series	FG132

Change History

V1.3 (2024-03-05)	Add "Applicable Models"
V1.2 (2023-11-15)	Removed Chinese characters from figures.
V1.1 (2023-09-06)	Changed the time in copyright statement from 2022 to 2023. Updated the log capture methods and LOG TAG.
V1.0 (2021-06-24)	Initial version

1 Overview

If the customer's host computer uses an Android system and there are problems such as failure to access the Internet, failure to make phone calls, or failure to send SMS messages, you need to capture the RIL logs for analysis. The following section describes the log capture and analysis method.

1.1 Log Capture Method

The RIL log capture method is as follows:

```
adb logcat -b all > xx
```

1.2 LOG TAG (Log label)

Logs output by different APPs are distinguished by LOG TAG. In the RIL log, you can search for "GHT_RIL" to get the information output by the RIL, or search for "GHT_AT" to get the AT commands sent and received by the RIL.

You can also search for the regular expression "GHT_RIL|GHT_AT" to get all the information of the RIL and AT at the same time. See Figure 1.

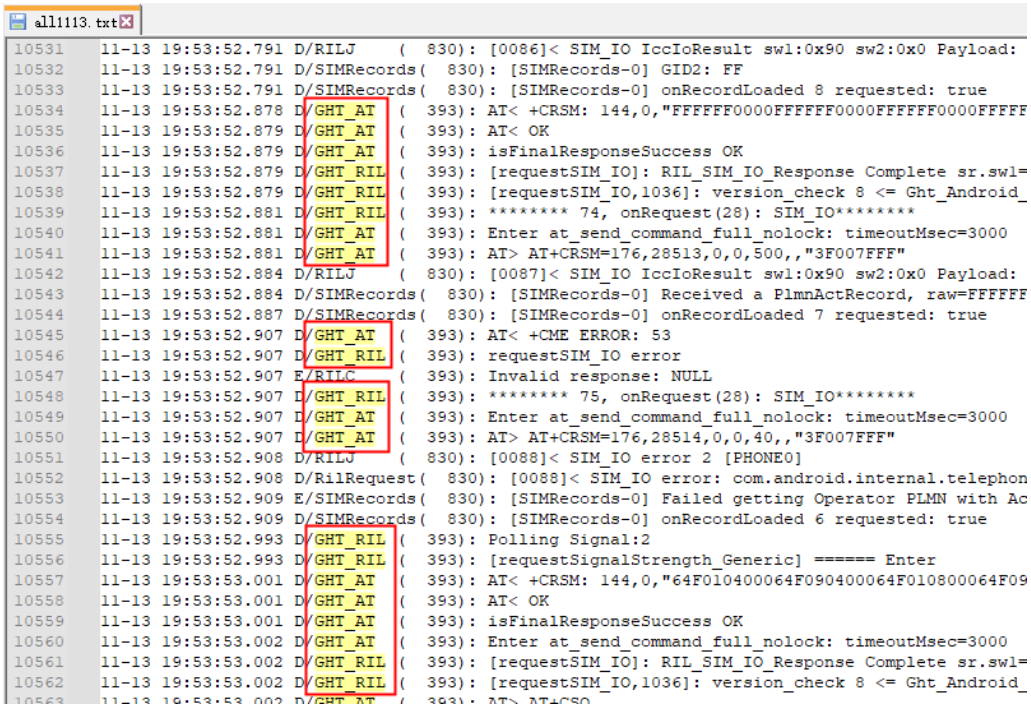

Figure 1. "GHT_RIL|GHT_AT" search result

Very few RIL libraries have older versions, and the LOG TAG is "RLOG-RIL" and "RLOG-AT" (as shown in Figure 2), or "RIL" and "AT" (as shown in Figure 3), which is expected.

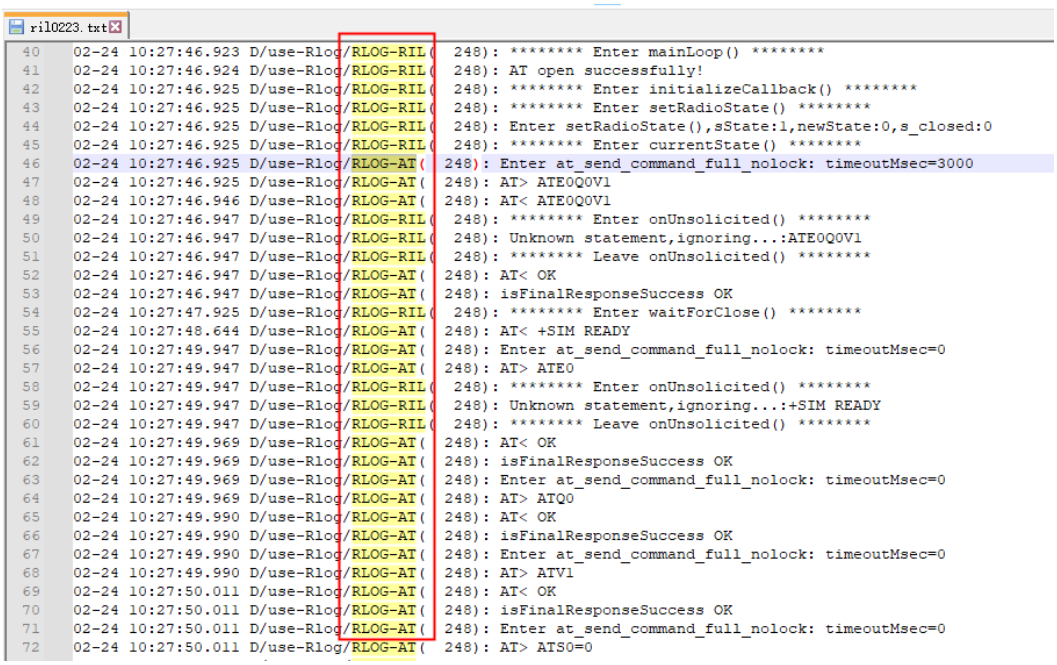

Figure 2. "RLOG-RIL|RLOG-AT" search result

ril0609.txt

```
06-10 15:13:06.456 D/AT       (  243): AT< +SIM READY
06-10 15:13:07.127 D/TelephonyManager(  504): /proc/cmdline=earlycon=uart8250,mmio32,0xffla0000 swiotl
06-10 15:13:08.250 D/AT       (  243): End sleep for 3 seconds.
06-10 15:13:08.250 D/AT       (  243): ********Leave at_handshake********
06-10 15:13:08.250 D/AT       (  243): Enter at_send_command_full_nolock: timeoutMsec=10000
06-10 15:13:08.250 D/AT       (  243): AT> ATE0
06-10 15:13:08.250 D/RIL      (  243): ******** Enter onUnsolicited() ********
06-10 15:13:08.250 D/RIL      (  243): [onUnsolicited,2846]Report SIM Inserted, sState:0(RADIO_OFF)
06-10 15:13:08.251 D/RIL      (  243): ******** Leave onUnsolicited() ********
06-10 15:13:08.271 D/AT       (  243): AT< OK
06-10 15:13:08.272 D/AT       (  243): isFinalResponseSuccess OK
06-10 15:13:08.272 D/AT       (  243): Enter at_send_command_full_nolock: timeoutMsec=10000
06-10 15:13:08.272 D/AT       (  243): AT> ATQ0
06-10 15:13:08.292 D/AT       (  243): AT< OK
06-10 15:13:08.293 D/AT       (  243): isFinalResponseSuccess OK
06-10 15:13:08.293 D/AT       (  243): Enter at_send_command_full_nolock: timeoutMsec=10000
06-10 15:13:08.293 D/AT       (  243): AT> ATV1
06-10 15:13:08.314 D/AT       (  243): AT< OK
06-10 15:13:08.314 D/AT       (  243): isFinalResponseSuccess OK
06-10 15:13:08.314 D/AT       (  243): Enter at_send_command_full_nolock: timeoutMsec=10000
06-10 15:13:08.314 D/AT       (  243): AT> ATS0=0
06-10 15:13:08.335 D/AT       (  243): AT< OK
06-10 15:13:08.335 D/AT       (  243): isFinalResponseSuccess OK
06-10 15:13:08.335 D/AT       (  243): Enter at_send_command_full_nolock: timeoutMsec=10000
06-10 15:13:08.335 D/AT       (  243): AT> AT+CMEE=1
06-10 15:13:08.356 D/AT       (  243): AT< OK
06-10 15:13:08.356 D/AT       (  243): isFinalResponseSuccess OK
06-10 15:13:08.356 D/AT       (  243): Enter at_send_command_full_nolock: timeoutMsec=10000
```

Figure 3. "RIL|AT" search result

1.3 Querying the RIL Version

You can query the current RIL version from the RIL LOG, which is usually located at the beginning of the RIL LOG, under the "getVersion()" information. The following figure shows the log in the version of "L61x&MC61x_RIL_V7X.06.V1.0.0".

ril0223.txt

```
02-24 10:27:44.720 D/RILD     (  248): **RIL Daemon Started**
02-24 10:27:44.720 D/RILD     (  248): **RILd param count=6**
02-24 10:27:44.740 W/RILD     (  248): RIL_SAP_Init not defined or exported in /system/lib64/libreference
02-24 10:27:44.740 D/RILD     (  248): RIL_Init argc = 5 clientId = 0
02-24 10:27:44.740 D/use-Rlog/RLOG-RIL(  248): ******** Enter RIL_Init() ********
02-24 10:27:44.740 D/use-Rlog/RLOG-RIL(  248): setenforce 0
02-24 10:27:44.814 D/use-Rlog/RLOG-RIL(  248): ******** Enter getVersion() ********
02-24 10:27:44.815 D/use-Rlog/RLOG-RIL(  248): L61x&MC61x_RIL_V7X.06.V1.0.0
02-24 10:27:44.815 D/use-Rlog/RLOG-RIL(  248): uname -r > /data/ODM_KERNEL_VERSION
02-24 10:27:44.895 D/use-Rlog/RLOG-RIL(  248): ODM_KERNEL_VERSION :: 4.4.103
02-24 10:27:44.896 D/use-Rlog/RLOG-RIL(  248): 4.4.103
02-24 10:27:45.897 D/RIL      (  248): find fibocom module idVendor:[1782] idProduct:[4d11]
02-24 10:27:45.897 D/RIL      (  248): usb_device_info.usbdevice_path:[/sys/bus/usb/devices/3-1]
02-24 10:27:45.897 D/RIL      (  248): find usb serial option driver, but do not cantain fibocom vid&pid
02-24 10:27:46.879 D/TelephonyManager(  200): No /proc/cmdline exception=java.io.FileNotFoundException:
02-24 10:27:46.879 D/TelephonyManager(  200): /proc/cmdline=
02-24 10:27:46.880 D/TelephonyManager(  201): No /proc/cmdline exception=java.io.FileNotFoundException:
02-24 10:27:46.880 D/TelephonyManager(  201): /proc/cmdline=
```

Figure 4. RIL version

2 Module Identification Anomalies

This chapter describes several methods to check whether the module identification is abnormal.

- Check whether VID&PID is identified.
- Check whether the ttyUSB interface is identified.
- Check whether the upper layer delivers requests.
- Check whether the AT commands are successfully sent.

If there are no problems in the above checks, the module is successfully identified.

2.1 Checking Whether VID&PID Is Identified

If the RIL successfully identifies the VID&PID of the module, it outputs the information in the RIL LOG.

For example, if the VID&PID of the current module is "1782 4d11", and the RIL successfully identifies the module, the following information will be output in the RIL log:

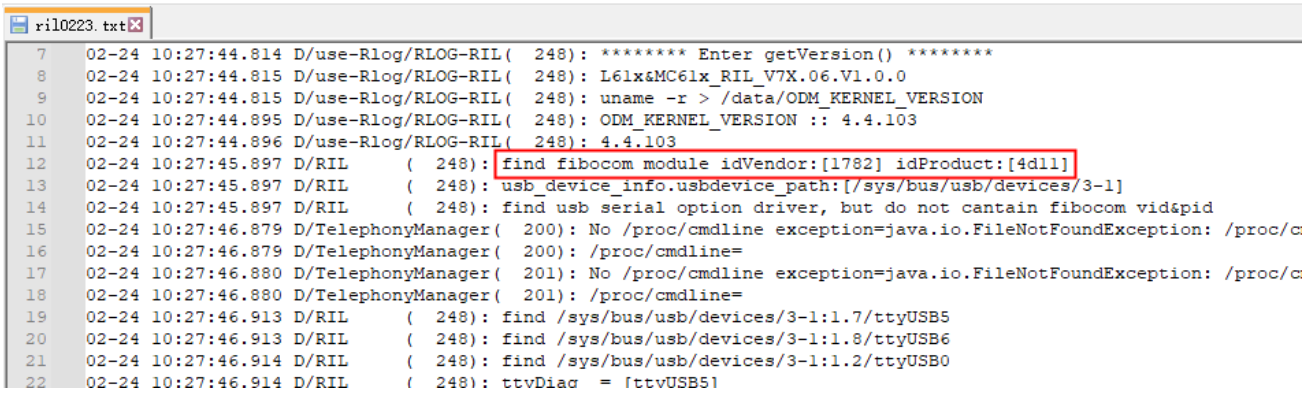

Figure 5. RIL successfully identified the VID&PID

If the module is connected to the Android device, but the RIL does not successfully identify the VID&PID, the RIL will keep trying to find the available device and output the following log:

ril0624.txt

```
06-24 12:06:36.991 D/RILD    (  287): **RIL Daemon Started**
06-24 12:06:36.991 D/RILD    (  287): **RILd param count=3**
06-24 12:06:37.029 W/RILD    (  287): RIL_SAP_Init not defined or exported in /vendor/lib64/libquectel-ril.so: undefined sy
06-24 12:06:37.029 D/RILD    (  287): RIL_Init argc = 3 clientId = 0
06-24 12:06:37.029 D/use-Rlog/RLOG-RIL(  287): ******** Enter RIL_Init() ********
06-24 12:06:37.029 D/use-Rlog/RLOG-RIL(  287): setenforce 0
06-24 12:06:37.180 D/use-Rlog/RLOG-RIL(  287): ******** Enter getVersion() ********
06-24 12:06:37.181 D/use-Rlog/RLOG-RIL(  287): L61x&MC61x_RIL_V10X.06.V1.0.3
06-24 12:06:37.181 D/use-Rlog/RLOG-RIL(  287): uname -r > /data/ODM_KERNEL_VERSION
06-24 12:06:37.247 D/use-Rlog/RLOG-RIL(  287): ODM_KERNEL_VERSION ::
06-24 12:06:37.250 D/use-Rlog/RLOG-RIL(  287):
06-24 12:06:38.251 E/RIL     (  287): getUsbPortInfo error errnum:[-1]
06-24 12:06:38.251 D/RIL     (  287): please check usb port state,RIL do not detech the usb devices
06-24 12:06:38.403 D/TelephonyManager(  232): No /proc/cmdline exception=java.io.FileNotFoundException: /proc/cmdline: oper
06-24 12:06:38.403 D/TelephonyManager(  232): /proc/cmdline=
06-24 12:06:39.253 E/RIL     (  287): getUsbPortInfo error errnum:[-1]
06-24 12:06:39.253 D/RIL     (  287): please check usb port state,RIL do not detech the usb devices
06-24 12:06:40.255 E/RIL     (  287): getUsbPortInfo error errnum:[-1]
06-24 12:06:40.255 D/RIL     (  287): please check usb port state,RIL do not detech the usb devices
06-24 12:06:41.014 E/TelephonyManager(  445): IMSI error: Subscriber Info is null
06-24 12:06:41.139 D/TelephonyManager(  233): No /proc/cmdline exception=java.io.FileNotFoundException: /proc/cmdline: oper
06-24 12:06:41.139 D/TelephonyManager(  233): /proc/cmdline=
06-24 12:06:41.256 E/RIL     (  287): getUsbPortInfo error errnum:[-1]
06-24 12:06:41.257 D/RIL     (  287): please check usb port state,RIL do not detech the usb devices
06-24 12:06:42.258 E/RIL     (  287): getUsbPortInfo error errnum:[-1]
06-24 12:06:42.258 D/RIL     (  287): please check usb port state,RIL do not detech the usb devices
06-24 12:06:43.260 E/RIL     (  287): getUsbPortInfo error errnum:[-1]
06-24 12:06:43.260 D/RIL     (  287): please check usb port state,RIL do not detech the usb devices
06-24 12:06:44.262 E/RIL     (  287): getUsbPortInfo error errnum:[-1]
06-24 12:06:44.262 D/RIL     (  287): please check usb port state,RIL do not detech the usb devices
06-24 12:06:45.263 E/RIL     (  287): getUsbPortInfo error errnum:[-1]
06-24 12:06:45.264 D/RIL     (  287): please check usb port state,RIL do not detech the usb devices
06-24 12:06:46.157 D/TelephonyRegistry(  445): listen oscl: mHasNotifySubscriptionInfoChangedOccurred==false no callback
```

Figure 6. RIL unsuccessfully identified the VID&PID

In such a case, check whether the module is normally connected to the Android device. Run the following command to confirm:

```
adb shell

lsusb
```

```
rk3399_roc_pc_plus:/ $ exit

D:\wu\cmder $ adb shell
rk3399_roc_pc_plus:/ $ lsusb
Bus 005 Device 001: ID 1d6b:0002
Bus 003 Device 001: ID 1d6b:0002
Bus 001 Device 001: ID 1d6b:0002
Bus 006 Device 001: ID 1d6b:0001
Bus 001 Device 002: ID 0bda:d723
Bus 004 Device 001: ID 1d6b:0003
Bus 002 Device 001: ID 1d6b:0001
rk3399_roc_pc_plus:/ $
```

Figure 7. Checking whether the module is connected

If the result does not contain the module VID and PID, check whether the module is powered on and started normally. If necessary, seek assistant from hardware personnel.

If the result contains the module VID and PID, the module is normally connected to the Android device. Request the RIL personnel to check whether the RIL has been adapted to this module.

2.2 Checking Whether the ttyUSB Interface Is Identified

If the RIL identifies the ttyUSB interface correctly, the following information will be output in the RIL log to indicate which ttyUSB interface is used by Diag, Modem, and AT:

ril0223.txt

```
02-24 10:27:45.897 D/RIL      (  248): usb_device_info.usbdevice_path:[/sys/bus/usb/devices/3-1
02-24 10:27:45.897 D/RIL      (  248): find usb serial option driver, but do not cantain fibocc
02-24 10:27:46.879 D/TelephonyManager(  200): No /proc/cmdline exception=java.io.FileNotFoundE
02-24 10:27:46.879 D/TelephonyManager(  200): /proc/cmdline=
02-24 10:27:46.880 D/TelephonyManager(  201): No /proc/cmdline exception=java.io.FileNotFoundE
02-24 10:27:46.880 D/TelephonyManager(  201): /proc/cmdline=
02-24 10:27:46.913 D/RIL      (  248): find /sys/bus/usb/devices/3-1:1.7/ttyUSB5
02-24 10:27:46.913 D/RIL      (  248): find /sys/bus/usb/devices/3-1:1.8/ttyUSB6
02-24 10:27:46.914 D/RIL      (  248): find /sys/bus/usb/devices/3-1:1.2/ttyUSB0
02-24 10:27:46.914 D/RIL      (  248): ttyDiag  = [ttyUSB5]
02-24 10:27:46.914 D/RIL      (  248): ttyModem = [ttyUSB6]
02-24 10:27:46.914 D/RIL      (  248): ttyAT    = [ttyUSB0]
02-24 10:27:46.914 D/RIL      (  248): getUsbPortInfo diagchannel   :[/dev/ttyUSB5]
02-24 10:27:46.914 D/RIL      (  248): getUsbPortInfo atchannel   :[/dev/ttyUSB0]
02-24 10:27:46.914 D/RIL      (  248): getUsbPortInfo datachannel :[/dev/ttyUSB6]
02-24 10:27:46.914 D/use-Rlog/RLOG-RIL(  248): fibocom get atchannel = /dev/ttyUSB0
02-24 10:27:46.914 D/use-Rlog/RLOG-RIL(  248): fibocom get diagchannel = /dev/ttyUSB5
02-24 10:27:46.914 D/use-Rlog/RLOG-RIL(  248): fibocom get datachannel = /dev/ttyUSB6
02-24 10:27:46.914 D/use-Rlog/RLOG-RIL(  248): Opening dialmode 1
02-24 10:27:46.914 D/RILD     (  248): RIL_Init rilInit completed
02-24 10:27:46.914 I/RILC     (  248): SIM_COUNT: 1
```

Figure 8. RIL successfully identified the ttyUSB interface

If the RIL does not identify the ttyUSB interface properly, it will report the following error information:

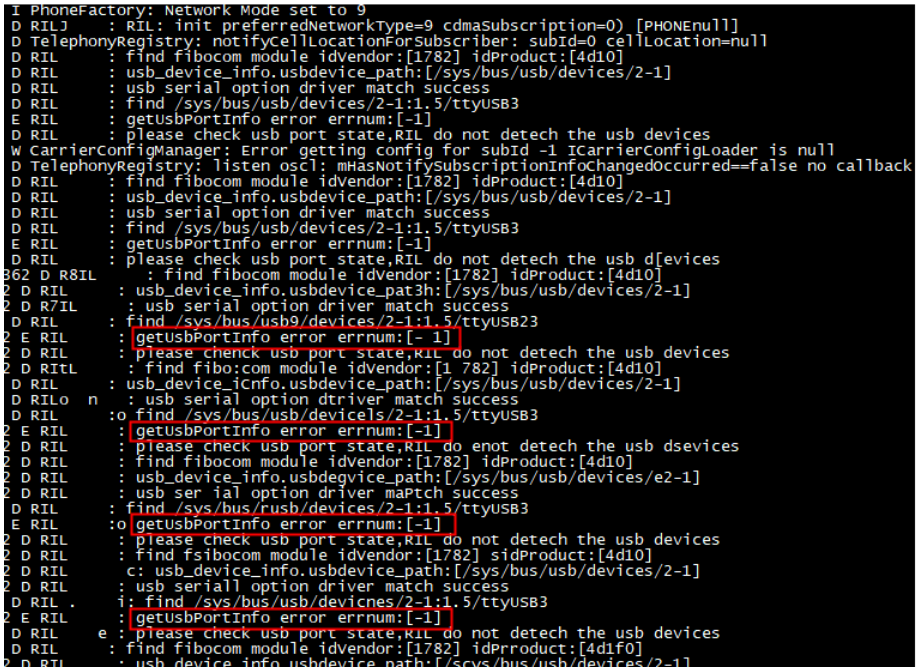

Figure 9. RIL unsuccessfully identified the ttyUSB interface

This is a rare situation and may be due to differences between the customer's device and the equipment used by Fibocom for self-test. In this case, you can switch to a different USB mode and try again (for example, from 31 to 32, or vice versa), or request the RIL personnel to implement adaptation.

2.3 Checking Whether the Upper Layer Delivers Requests

If the RIL communicates with the upper layer (such as RILC and RILJ) normally, the upper layer will send various requests. The log displays the following information:

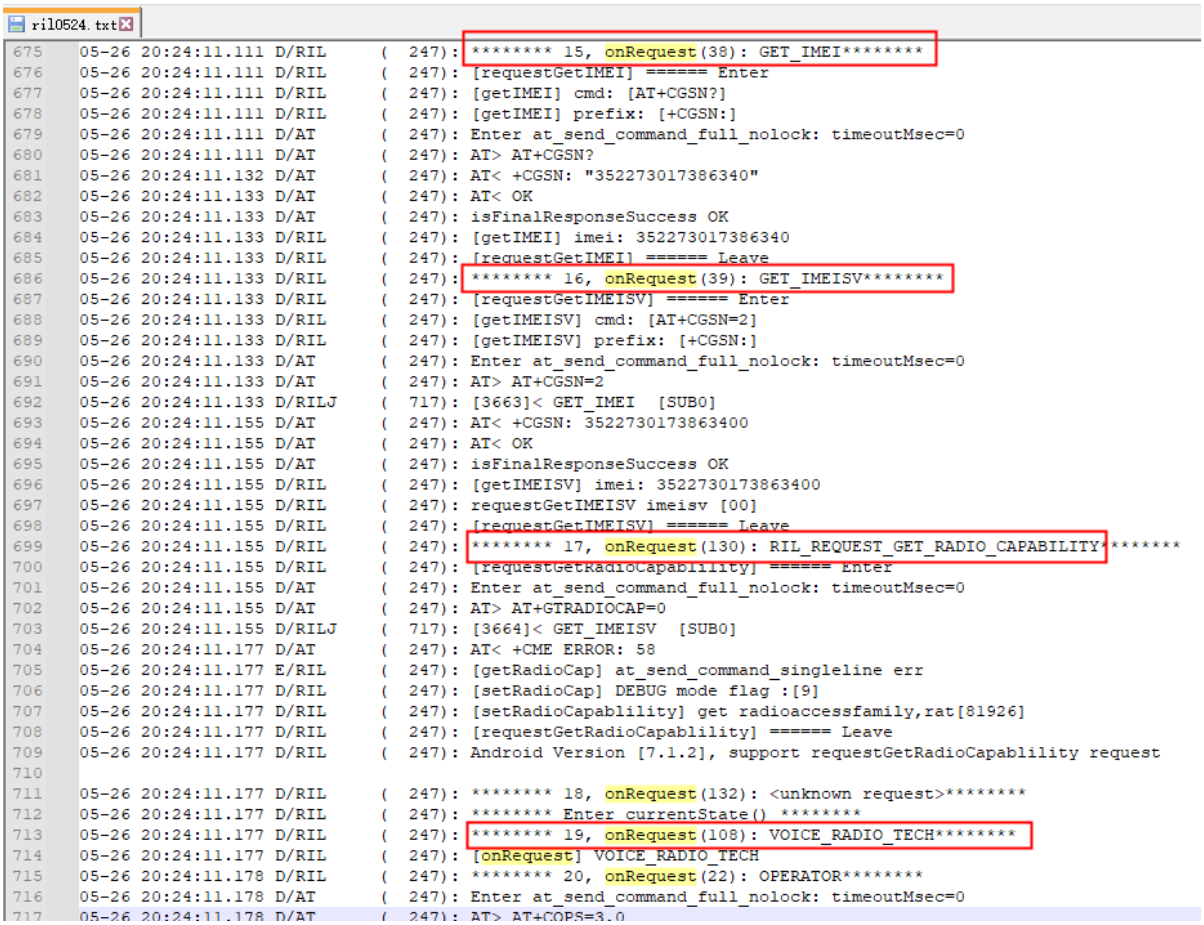

Figure 10. "onRequest" in RIL log

If there is no result returned when you search for "onRequest", the upper-layer Radio service is not started normally. In this situation, capture the RIL log according to section 1.1 and search for "radio" to confirm.

Figure 11 shows the normal main log. For the android.hardware.radio service, search for the device from the latest supported version (as shown in Figure 11 1.5) to the earliest version, and stop the search when an available service is found. As shown in the following log, the Radio services in versions 1.5-1.2 are not found, indicating that the Radio service in version 1.1 has been found (if it is not found, the log will output the corresponding information). If the log shows that the Radio services in versions 1.5-1.0 are not found, it indicates that the Radio service (the earliest version is 1.0) is not found. In this case, request the customer to modify the upper-layer code of the Android device to enable the Radio service.

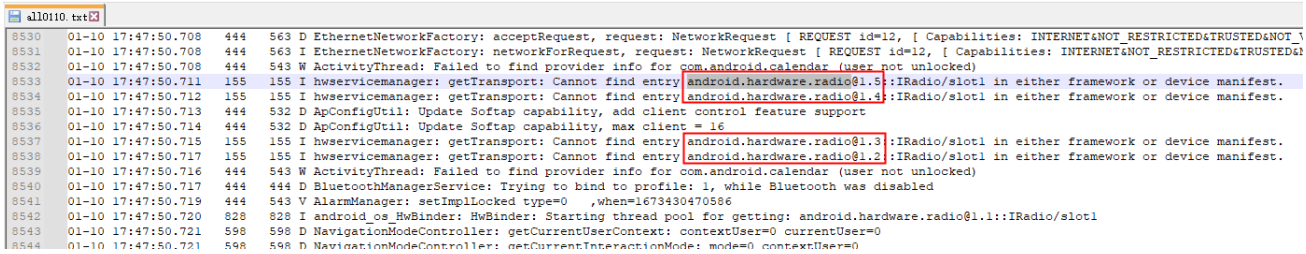

Figure 11. Normal main log

2.4 Checking Whether the AT Commands Are Sent

Throughout the operating process, the RIL continuously sends AT commands to the module and receives responses and other messages from the module. Generally, the first AT command sent by RIL to the module is "ATE0Q0V1" and RIL is expected to receive an "OK" response:

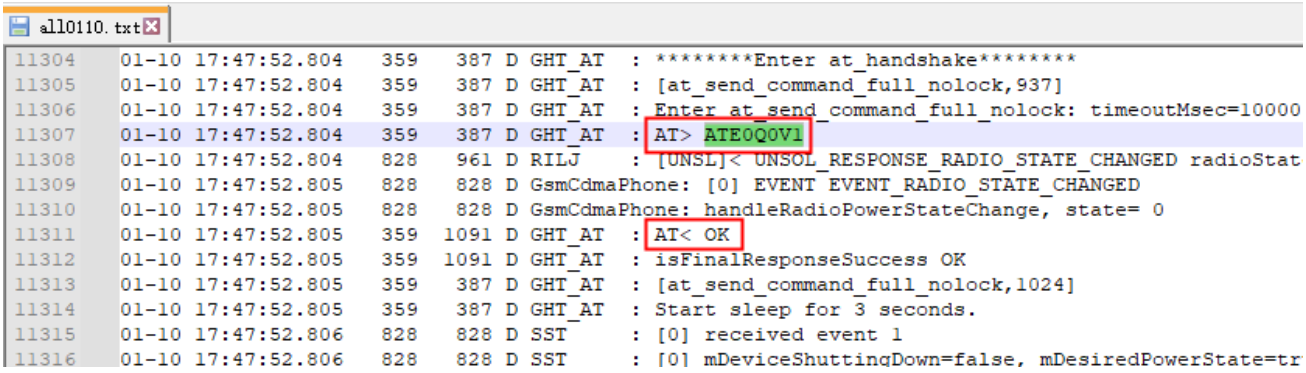

Figure 12. First AT command sent by RIL

If the above command is not successfully sent, or if an "OK" response is not received, it is an anomaly. In this situation, search for "ttyAT" in the RIL log and check the AT interface used by the RIL currently. The following figure shows that the RIL is currently using the "/dev/ttyUSB0" as the AT interface.

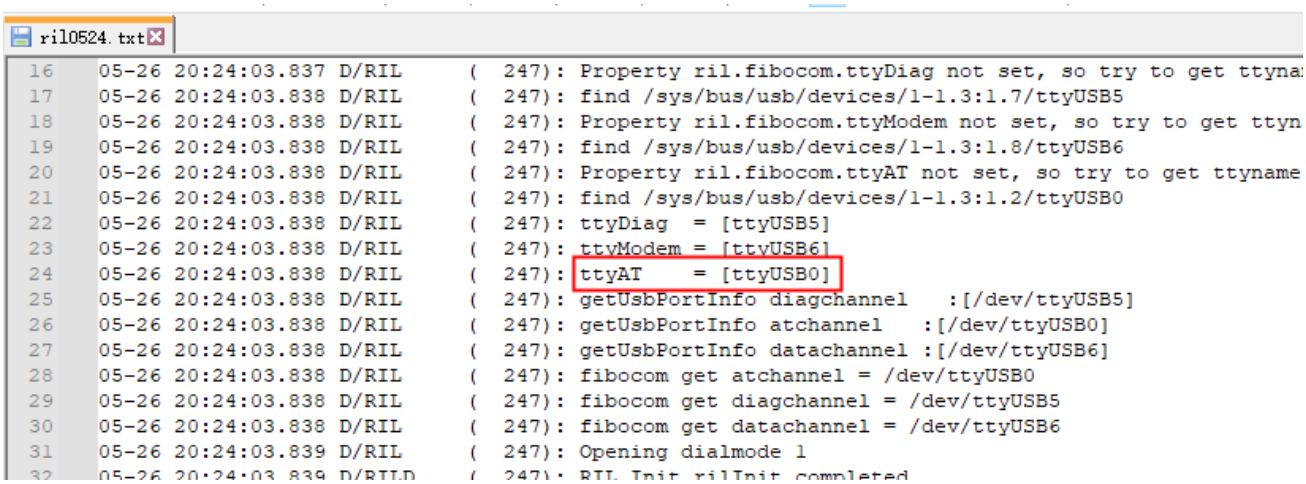

Figure 13. AT interface used by the RIL

Run the following commands to manually send an AT command on the Android device.

```
adb shell

su

stop ril-daemon

echo –e –n “at\r” > /dev/ttyUSBx | cat /dev/ttyUSBx      //Response from ttyUSBx according to the previous query result

//The interface is /dev/ttyUSB0, as shown in the figure above.
```

If the AT response cannot be received on the Android device, the AT interface of the module may be abnormal. Restart the device and try again, or contact the RIL personnel to modify the AT interface used.

3 Abnormal Calling Service

If a call cannot be initiated, it may be caused by the following reasons:

1. The module does not support the calling function.

2. The calling APP of the upper layer is abnormal, for example, insufficient permissions.

The following describes these situations in detail.

3.1 Checking Whether the Module Supports the Calling Service

Search for "DIAL" in the RIL log to find the content of "onRequest(xx): DIAL". The "ATDxxxxxxx" command sent in the request is the AT command for calling:

ril0627.txt

```
4916  06-27 16:29:35.162 D/GsmCdmaCallTracker(  717): [GsmCdmaCallTracker] update phone state, old=IDLE new=OFFHOOK
4917  06-27 16:29:35.162 D/RILJ    (  717): [3983]< SET_MUTE  [SUB0]
4918  06-27 16:29:35.162 D/RILC    (  241): dispatchDial
4919  06-27 16:29:35.163 D/RIL     (  241): ********335, onRequest(10): DIAL*******
4920  06-27 16:29:35.163 D/AT      (  241): Enter at_send_command_full_nolock: timeoutMsec=10000
4921  06-27 16:29:35.163 D/AT      (  241): AT> AT+COPS=3,2
4922  06-27 16:29:35.165 D/RILJ    (  717): [3985]> SET_SUPP_SVC_NOTIFICATION [SUB0]
4923  06-27 16:29:35.167 D/PhoneNumberUtils(  717): slotId:0 subId:1 country:null emergencyNumbers:
4924  06-27 16:29:35.168 D/PhoneNumberUtils(  717): slotId:0 subId:1 country:null emergencyNumbers:
4925  06-27 16:29:35.175 D/DCT      (  717): [0]onVoiceCallStarted
4926  06-27 16:29:35.184 D/AT       (  241): AT< OK
4927  06-27 16:29:35.185 D/AT       (  241): isFinalResponseSuccess OK
4928  06-27 16:29:35.185 D/AT       (  241): Enter at_send_command_full_nolock: timeoutMsec=10000
4929  06-27 16:29:35.185 D/AT       (  241): AT> AT+COPS?
4930  06-27 16:29:35.208 D/AT       (  241): AT< +COPS: 0,2,"46000",7
4931  06-27 16:29:35.208 D/AT       (  241): AT< OK
4932  06-27 16:29:35.208 D/AT       (  241): isFinalResponseSuccess OK
4933  06-27 16:29:35.208 D/RIL      (  241): network type 7,   g_cops_lte 7 ,mode_flag: 9
4934  06-27 16:29:35.208 D/RIL      (  241): network type E-UTRAN
4935  06-27 16:29:35.208 D/RIL      (  241): get current network22222 E-UTRAN
4936
4937  06-27 16:29:35.208 D/RIL      (  241): network type 14
4938  06-27 16:29:35.208 D/RIL      (  241): voice dial begin
4939  06-27 16:29:35.208 D/AT       (  241): Enter at_send_command_full_nolock: timeoutMsec=10000
4940  06-27 16:29:35.208 D/AT       (  241): AT> ATD18706761476;
4941  06-27 16:29:35.235 D/AT       (  241): AT< OK
4942  06-27 16:29:35.235 D/AT       (  241): isFinalResponseSuccess OK
4943  06-27 16:29:35.237 D/RIL      (  241): ********336, onRequest(62): <unknown request>********
4944  06-27 16:29:35.239 D/RILJ     (  717): [3984]< DIAL  [SUB0]
```

Figure 14. "DIAL" request

If you cannot find the "DIAL" request, the upper layer has not sent a calling request, and you need to check the upper layer according to the section 3.2.

If the "DIAL" request is found, but the ATD command in the request returns ERROR, check whether the currently used module or SIM card supports the calling function.

3.2 Checking the Calling APP Status

Generally, the upper-layer calling APP has two abnormal situations: There is no calling permission or the Android device does not support the calling function.

3.2.1 Checking Whether There Is a Calling Permission

Taking the Android 7 as an example, enable the calling permission in Settings > Apps > Phone > Permissions.

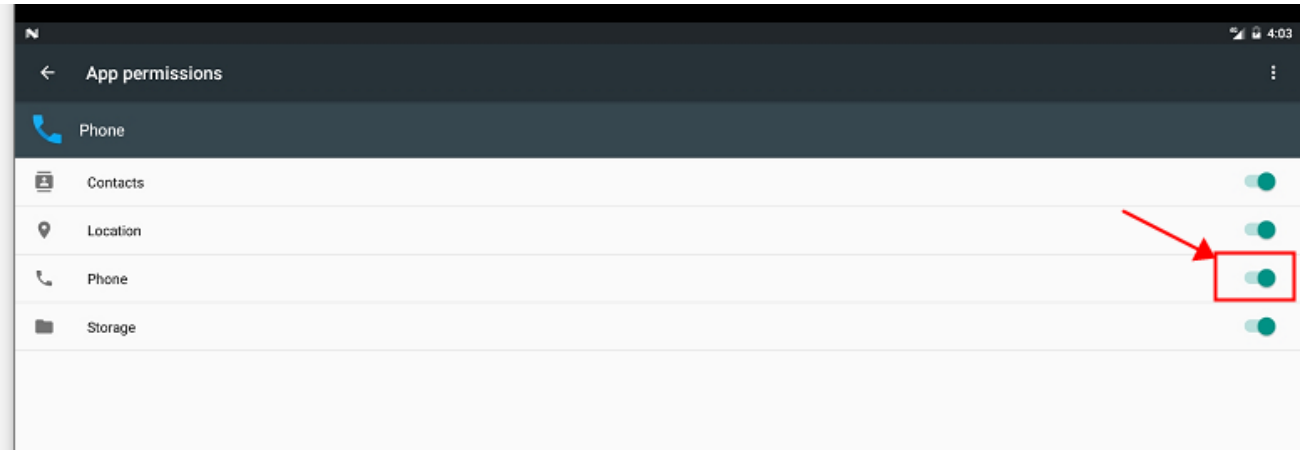

Figure 15. Calling permission of the APP

3.2.2 Checking Whether the Calling Function Is Enabled on the Device

If the customer uses the calling function for the first time and the calling fails, the calling function may not be enabled. If the customer has previously used the calling function, skip this step.

If the customer does not enable the calling function, enable the function according to section 6.1.

4 Abnormal SMS Service

Similar to the calling function, there are two abnormal situations of the SMS function.

1. The module does not support the SMS function.

2. The upper-layer SMS APP is abnormal.

The following describes these situations in detail.

4.1 Checking Whether the Module Supports the SMS Service

Search for "SEND_SMS" in the RIL log. If the log contains "onRequest(xx): SEND_SMS", it indicates that the RIL sends an SMS message. AT+CMGS is the AT command used for sending the SMS message. The details are shown in the following figure:

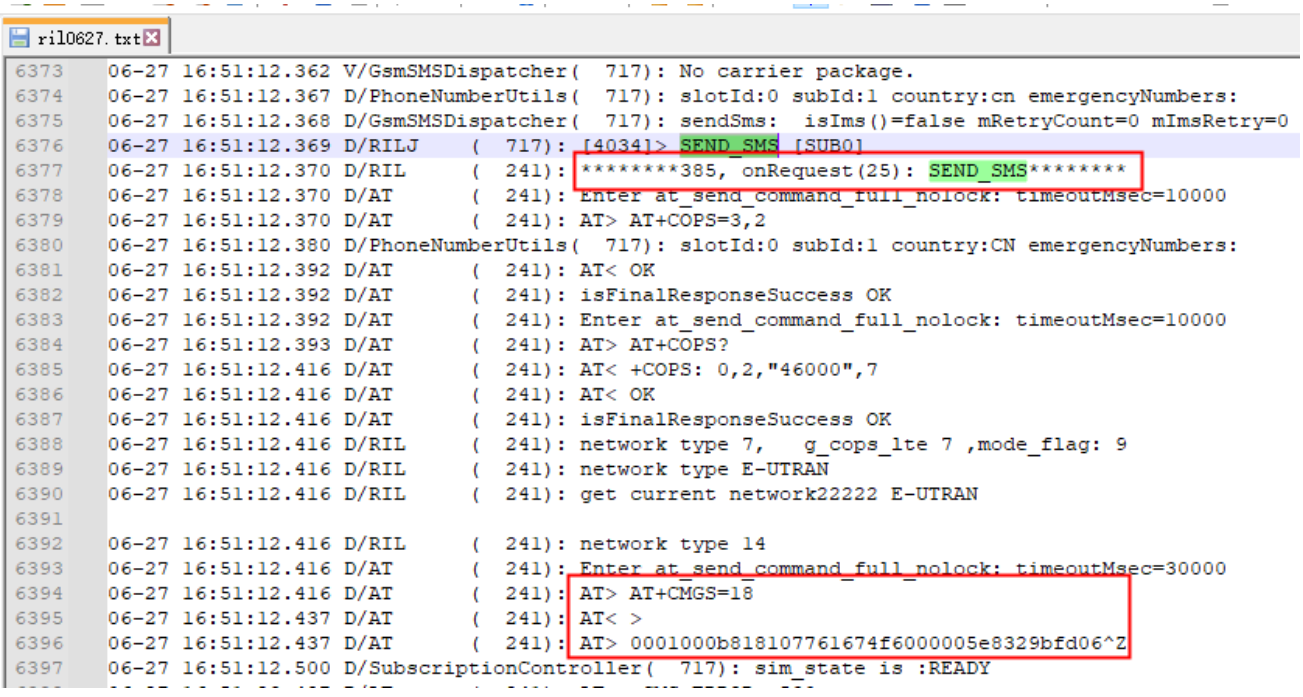

Figure 16. "SEND_SMS" request

If you cannot find the "SEND_SMS" request, the upper layer has not sent an SMS request, and you need to check the upper layer according to the section 4.2.

If the "SEND_SMS" request is found, but AT+CMGS command in the request returns ERROR, check whether the currently used module or SIM card supports the SMS function.

4.2 Checking the SMS APP Status

Generally, the upper-layer SMS APP has two abnormal situations: There is no calling permission or the Android device does not support the SMS function.

4.2.1 Checking Whether There Is a SMS Sending Permission

Taking the Android 7 as an example, enable the SMS permission in Settings > Apps > Messaging > Permissions.

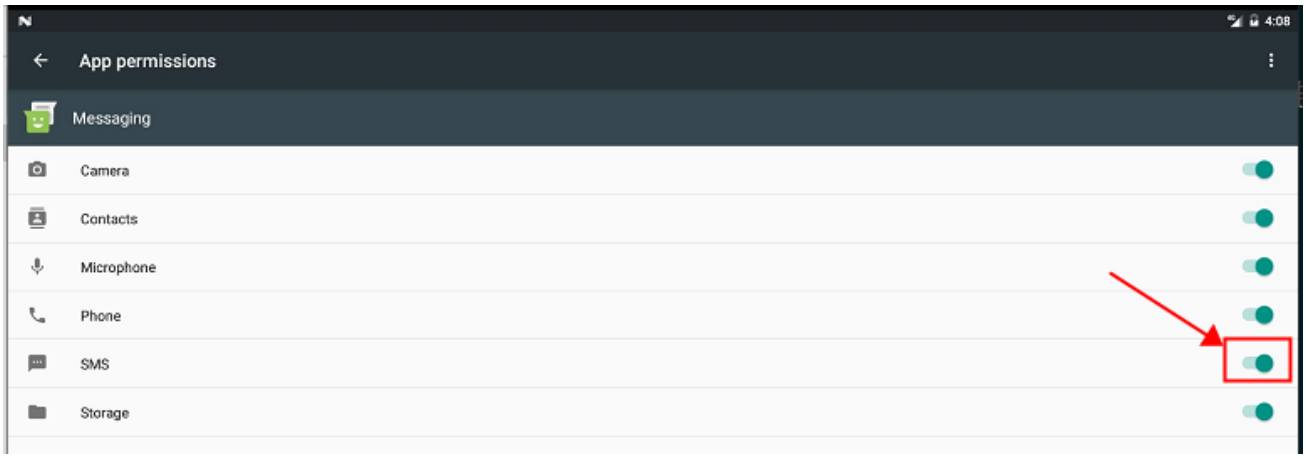

Figure 17. SMS permission of the APP

4.2.2 Checking Whether the SMS Function Is Enabled

If the SMS function is not enabled on the customer's device, enable this function according to section 6.1.

5 Abnormal Data Dial-up

There are many reasons for the data dial-up failure. This chapter describes the troubleshooting methods.

5.1 Checking Whether the Dial-up Request Is Successful

Check whether the upper layer initiates the dial-up and whether the dial-up is successful. In the RIL log, search for "SETUP_DATA_CALL". "onRequest(xx): SETUP_DATA_CALL" in the log indicates the RIL dial-up process. The log with the LOG TAG of RILJ and the content of "< SETUP_DATA_CALL" records the RIL dial-up result.

If the RIL log contains the RIL dial-up process and the result includes the IP and gateway addresses, the dial-up is successful. See Figure 18.

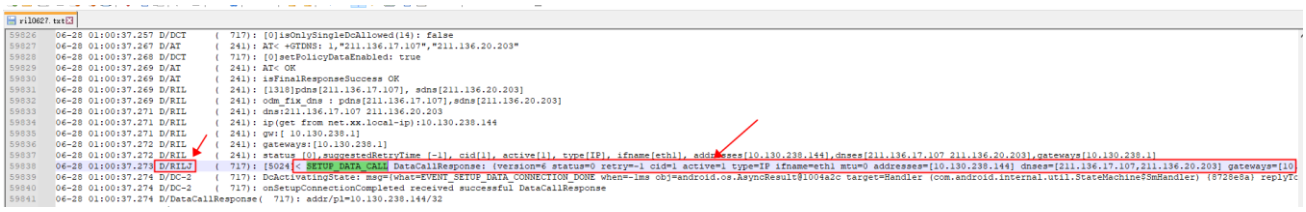

Figure 18. RIL dial-up result

If there is no RIL dial-up process, check whether the APN is normal according to section 5.2. If there is a dial-up process but the dial-up fails, proceed with the following steps.

5.1.1 ECM Dial-up

1. For the ECM dial-up, check whether OK is returned in response to the GTRNDIS command.

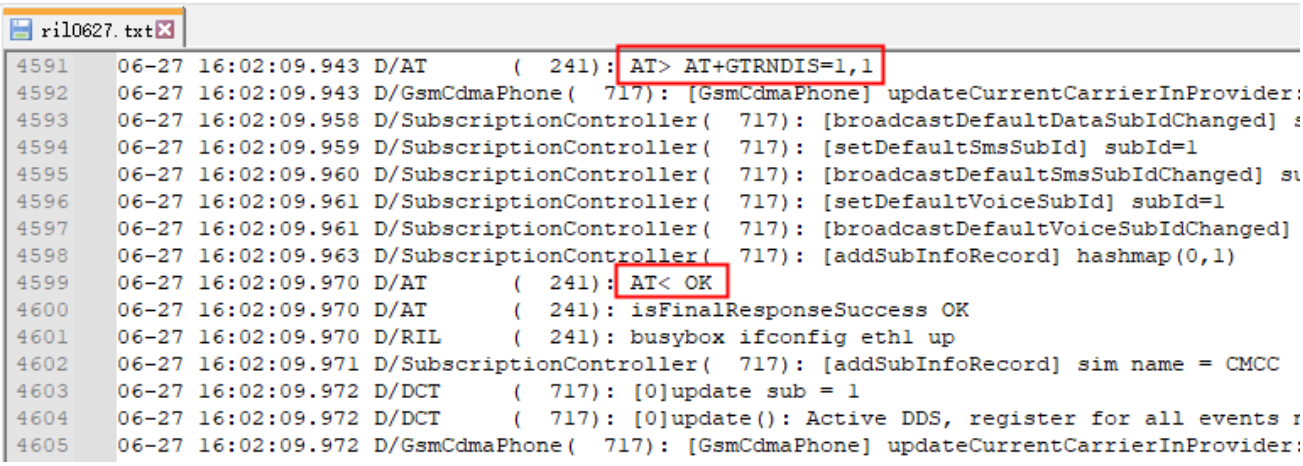

Figure 19. GTRNDIS command

If the command returns ERROR, check whether the USBMODE is correct. If USBMODE is correct, capture the log of the module and request the personnel responsible for the module to analyze.

5.1.2 PPP Dial-up

For the PPP dial-up, a common error is that the IP address is not obtained. The details are shown in the following figure:

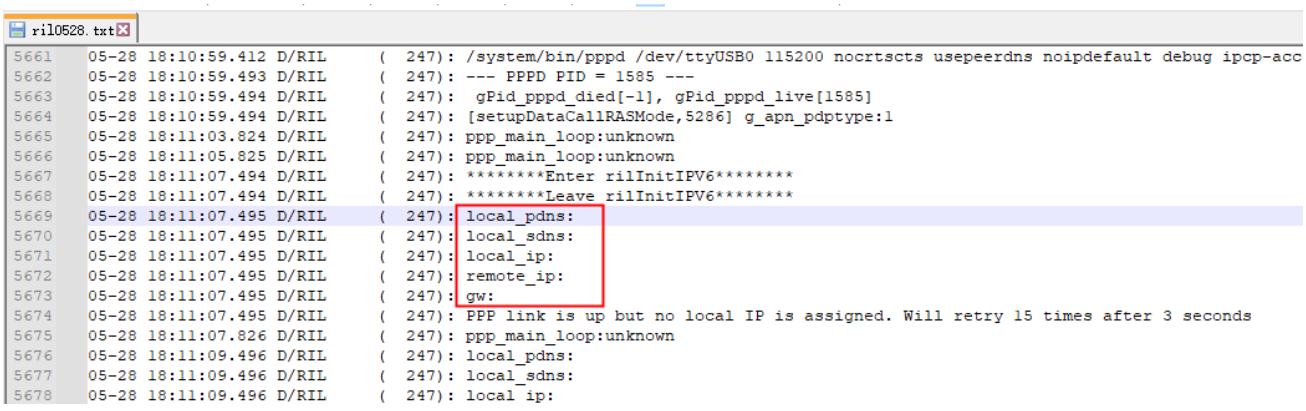

Figure 20. PPP dial-up without IP address

Analyze the RIL log (see section 1.1 for the capture method). If necessary, request the personnel responsible for the module to analyze. For example, identify the cause for the peer to deny the authentication, as shown in the following figure.

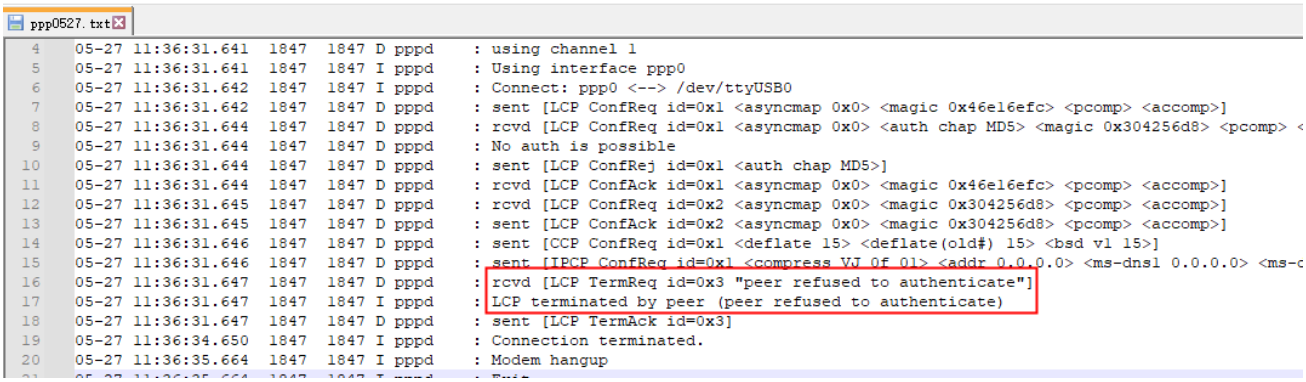

Figure 21. Example PPP dial-up error

5.2 Checking Whether the APN Is Normal

Search for "ATTACH_APN" in the RIL log. If there is a "RIL_REQUEST_SET_INITIAL_ATTACH_APN" request and the "AT+CGDCONT" command in this request does not return ERROR, the APN is normal. The details are shown in the following figure:

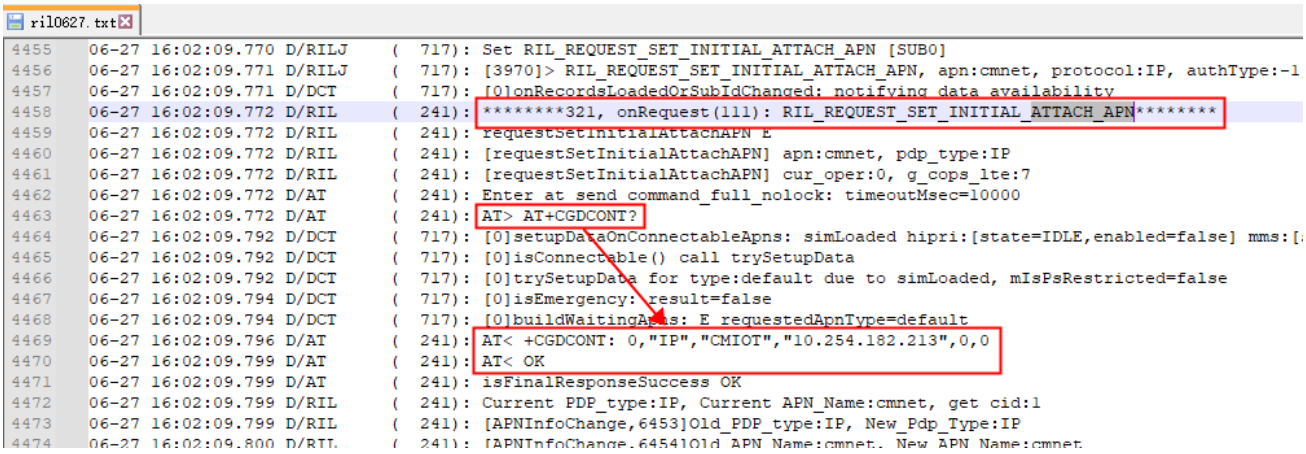

Figure 22. APN setting in RIL

If there is no "RIL_REQUEST_SET_INITIAL_ATTACH_APN" request, check whether the module is registered with the PS domain according to section 5.3. If this request exists but the AT+CGDCONT command returns ERROR, request the personnel responsible for the module to identify the cause for the error returned by the AT command.

5.3 Checking Whether the Module Registers with the PS Domain

The module cannot initiate the data dial-up if it is not registered with the PS domain.

In the RIL log, search for "DATA_REGISTRAION" and find the DATA_REGISTRATION_STATE request. Check whether the AT+CEREG? command in the request returns the correct information (as shown in the following figure).

ril0627.txt

```
06-27 16:02:06.433 D/RIL     (  241): ********249, onRequest(21): DATA_REGISTRATION_STATE********
06-27 16:02:06.433 D/RIL     (  241): ********enter requestRegistrationState********
06-27 16:02:06.433 D/AT      (  241): Enter at_send_command_full_nolock: timeoutMsec=10000
06-27 16:02:06.433 D/AT      (  241): AT> AT+COPS=3,2
06-27 16:02:06.434 D/RILJ    (  717): [3897]< OPERATOR {CHINA MOBILE, CMCC, 46000} [SUB0]
06-27 16:02:06.454 D/AT      (  241): AT< OK
06-27 16:02:06.455 D/AT      (  241): isFinalResponseSuccess OK
06-27 16:02:06.455 D/AT      (  241): Enter at_send_command_full_nolock: timeoutMsec=10000
06-27 16:02:06.455 D/AT      (  241): AT> AT+COPS?
06-27 16:02:06.477 D/AT      (  241): AT< +COPS: 0,2,"46000",7
06-27 16:02:06.477 D/AT      (  241): AT< OK
06-27 16:02:06.477 D/AT      (  241): isFinalResponseSuccess OK
06-27 16:02:06.477 D/RIL     (  241): network type 7,   g_cops_lte 7 ,mode_flag: 9
06-27 16:02:06.477 D/RIL     (  241): network type E-UTRAN
06-27 16:02:06.477 D/RIL     (  241): get current network22222 E-UTRAN

06-27 16:02:06.478 D/RIL     (  241): network type 14
06-27 16:02:06.478 D/AT      (  241): Enter at_send_command_full_nolock: timeoutMsec=10000
06-27 16:02:06.478 D/AT      (  241): AT> AT+CREG=2
06-27 16:02:06.499 D/AT      (  241): AT< OK
06-27 16:02:06.499 D/AT      (  241): isFinalResponseSuccess OK
06-27 16:02:06.499 D/AT      (  241): Enter at_send_command_full_nolock: timeoutMsec=10000
06-27 16:02:06.499 D/AT      (  241): AT> AT+CGREG=2
06-27 16:02:06.521 D/AT      (  241): AT< OK
06-27 16:02:06.521 D/AT      (  241): isFinalResponseSuccess OK
06-27 16:02:06.521 D/AT      (  241): Enter at_send_command_full_nolock: timeoutMsec=10000
06-27 16:02:06.521 D/AT      (  241): AT> AT+CEREG=2
06-27 16:02:06.542 D/AT      (  241): AT< OK
06-27 16:02:06.542 D/AT      (  241): isFinalResponseSuccess OK
06-27 16:02:06.543 D/AT      (  241): Enter at_send_command_full_nolock: timeoutMsec=10000
06-27 16:02:06.543 D/AT      (  241): AT> AT+CEREG?
06-27 16:02:06.568 D/AT      (  241): AT< +CEREG: 2,1,"90F3","07828A01",7
06-27 16:02:06.568 D/AT      (  241): AT< OK
06-27 16:02:06.569 D/AT      (  241): isFinalResponseSuccess OK
06-27 16:02:06.569 D/RIL     (  241): parseRegistrationState. Parsing: +CEREG: 2,1,"90F3","07828A01",7
06-27 16:02:06.569 D/RIL     (  241): commas = 4
06-27 16:02:06.569 D/RIL     (  241): resp[] = 1, 37107, 125995521, 7
06-27 16:02:06.569 D/RIL     (  241): resp[] = 1, 37107, 125995521, 7
06-27 16:02:06.569 D/RIL     (  241): [3837]registration[0][1][2][3] == 1,37107,125995521,7

06-27 16:02:06.569 D/RIL     (  241): registration[3] == 14

06-27 16:02:06.569 D/RIL     (  241): registration[0] == 1

06-27 16:02:06.569 D/RIL     (  241): ********250, onRequest(20): VOICE_REGISTRATION_STATE********
06-27 16:02:06.570 D/RIL     (  241): ********enter requestRegistrationState********
06-27 16:02:06.570 D/AT      (  241): Enter at_send_command_full_nolock: timeoutMsec=10000
06-27 16:02:06.570 D/AT      (  241): AT> AT+COPS=3,2
06-27 16:02:06.570 D/RILJ    (  717): [3898]< DATA_REGISTRATION_STATE {1, 90f3, 7828a01, 14} [SUB0]
06-27 16:02:06.591 D/AT      (  241): AT< OK
```

Figure 23. DATA_REGISTRATION_STATE request

If the AT+CEREG? command does not return the available cell information (for example, the return value is 2, 0), check whether the module is correctly registered with the network according to section 5.4.

5.4 Checking Whether the Network Registration Is Successful

In the RIL log, run COPS to find the AT+COPS command. Then, you can identify whether the module is successfully registered with the network. The following figure shows the log when the module is successfully registered with the network.

ril0627.txt

```
3138  06-27 16:02:06.433 D/AT      (  241): AT> AT+COPS=3,2
3139  06-27 16:02:06.434 D/RILJ    (  717): [3897]< OPERATOR {CHINA MOBILE, CMCC, 46000} [SUB0]
3140  06-27 16:02:06.454 D/AT      (  241): AT< OK
3141  06-27 16:02:06.455 D/AT      (  241): isFinalResponseSuccess OK
3142  06-27 16:02:06.455 D/AT      (  241): Enter at_send_command_full_nolock: timeoutMsec=10000
3143  06-27 16:02:06.455 D/AT      (  241): AT> AT+COPS?
3144  06-27 16:02:06.477 D/AT      (  241): AT< +COPS: 0,2,"46000",7
3145  06-27 16:02:06.477 D/AT      (  241): AT< OK
3146  06-27 16:02:06.477 D/AT      (  241): isFinalResponseSuccess OK
3147  06-27 16:02:06.477 D/RIL     (  241): network type 7,   g_cops_lte 7 ,mode_flag: 9
3148  06-27 16:02:06.477 D/RIL     (  241): network type E-UTRAN
3149  06-27 16:02:06.477 D/RIL     (  241): get current network22222 E-UTRAN
3150
```

Figure 24. AT+COPS command

Unsuccessful network registration may be caused by the following reasons, listed based on the order of probability from high to low:

1. The signal strength is poor.

2. The SIM card is not detected.

3. The network system is incorrect.

The following section respectively describes these situations in detail.

5.4.1 Signal Strength

In the RIL log, search for "CSQ" to obtain the current signal strength. The details are shown in the following figure:

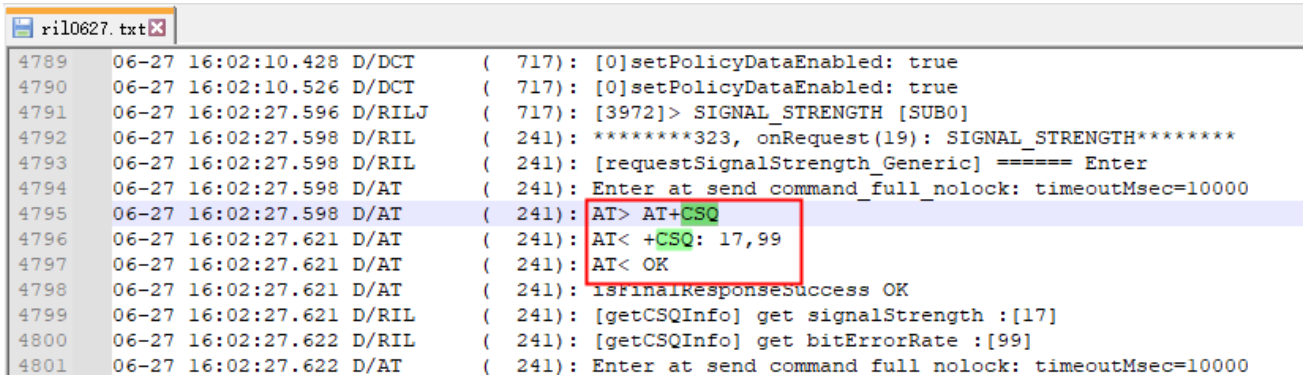

Figure 25. AT+CSQ

If the signal strength value is low, check the antenna and the signal coverage in the corresponding area.

5.4.2 SIM Card Detection

In the RIL log, search for "CPIN" to check whether the SIM card is correctly identified. The details are shown in the following figure:

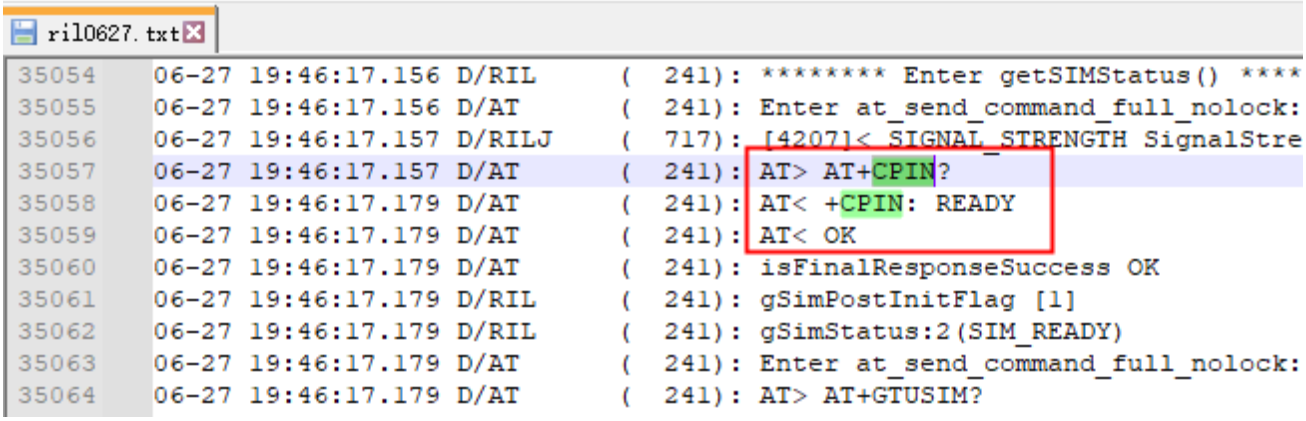

Figure 26. AT+CPIN

If the SIM card is not correctly loaded, check whether the SIM card is inserted properly or whether the hardware circuit is normal.

5.4.3 Network System

If the module uses the 4G+2G or 4G system and the current network system is 2G, the module may slowly register with the network or fail to register with the network.

In the RIL log, search for "GTRAT" to obtain the current network system. The details are shown in the following figure:

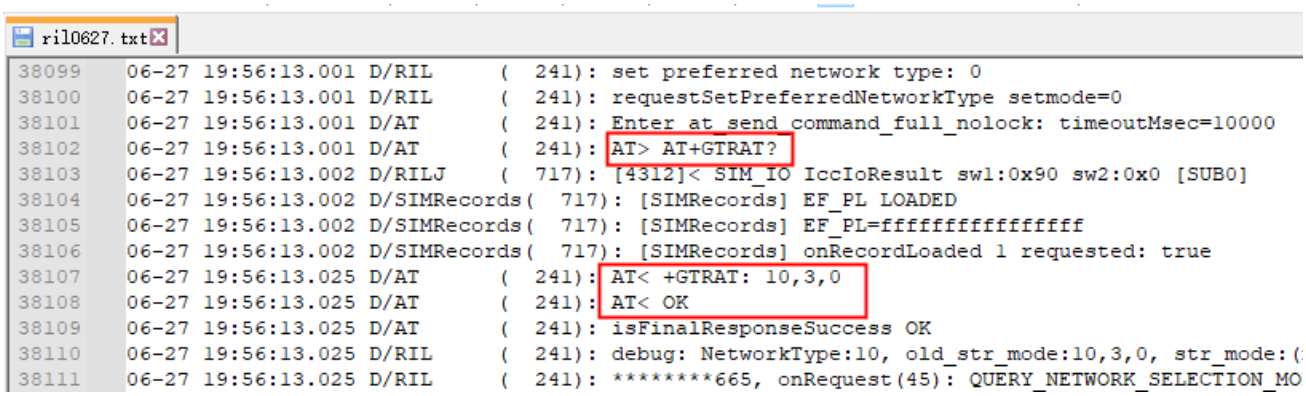

Figure 27. AT+GTRAT

If the module adopts the 2G network system, change the network system to 4G in Settings > More > Cellular networks > Preferred network type (taking Android 7 as an example).

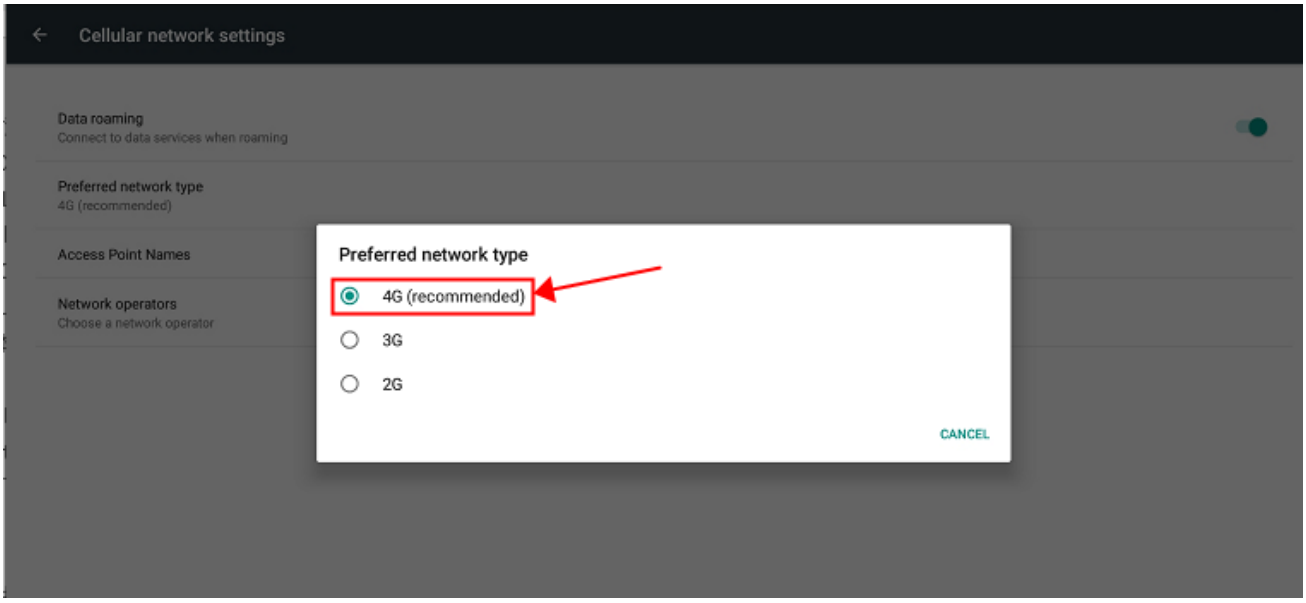

Figure 28. Setting the preferred network

6 Accessories

6.1 Modification on the AP Side

```
./device/rockchip/common/overlay/frameworks/base/core/res/res/values/config.xml
diff --git
a/device/rockchip/common/overlay/frameworks/base/core/res/res/values/config.xml
b/device/rockchip/common/overlay/frameworks/base/core/res/res/values/config.xml
index 5663d84..e88088b 100644
--- a/device/rockchip/common/overlay/frameworks/base/core/res/res/values/config.xml
+++ b/device/rockchip/common/overlay/frameworks/base/core/res/res/values/config.xml
@@ -20,10 +20,10 @@
 <resources>

     <!-- This device is not "voice capable"; it's data-only. -->
-    <bool name="config_voice_capable">false</bool>
+    <bool name="config_voice_capable">true</bool>

     <!-- This device does not allow sms service. -->
-    <bool name="config_sms_capable">false</bool>
+    <bool name="config_sms_capable">true</bool>

     <!-- If this is true, the screen will come on when you unplug usb/power/whatever.
-->
     <bool name="config_unplugTurnsOnScreen">true</bool>

./device/rockchip/common/tv/overlay/frameworks/base/core/res/res/values/config.xml
diff --git
a/device/rockchip/common/tv/overlay/frameworks/base/core/res/res/values/config.xml
b/device/rockchip/common/tv/overlay/frameworks/base/core/res/res/values/config.xml
index cbd8423..67d4dda 100755
```

```
---
a/device/rockchip/common/tv/overlay/frameworks/base/core/res/res/values/config.xml
+++
b/device/rockchip/common/tv/overlay/frameworks/base/core/res/res/values/config.xml
@@ -24,10 +24,10 @@
     <bool name="config_sendAudioBecomingNoisy">false</bool>

     <!-- This device is data-only. -->
-    <bool name="config_voice_capable">false</bool>
+    <bool name="config_voice_capable">true</bool>

     <!-- This device does not allow sms service. -->
-    <bool name="config_sms_capable">false</bool>
+    <bool name="config_sms_capable">true</bool>

     <!-- Control the default UI mode type to use when there is no other type override
          happening.  One of the following values (See Configuration.java):
```