QUECTEL®
Build a Smarter World

EC2X&AG35-QuecOpen WiFi API User Guide

LTE Module Series

Rev. Quectel_EC2X&AG35-QuecOpen_WiFi_API_User_Guide_V1.0

Date: 2018-04-07

Status: Preliminary

www.quectel.com

Our aim is to provide customers with timely and comprehensive service. For any assistance, please contact our company headquarters:

Quectel Wireless Solutions Co., Ltd.
7th Floor, Hongye Building, No.1801 Hongmei Road, Xuhui District, Shanghai 200233, China
Tel: +86 21 5108 6236
Email: info@quectel.com

Or our local office. For more information, please visit:

http://quectel.com/support/sales.htm

For technical support, or to report documentation errors, please visit:

http://quectel.com/support/technical.htm
Or email to: support@quectel.com

GENERAL NOTES

QUECTEL OFFERS THE INFORMATION AS A SERVICE TO ITS CUSTOMERS. THE INFORMATION PROVIDED IS BASED UPON CUSTOMERS' REQUIREMENTS. QUECTEL MAKES EVERY EFFORT TO ENSURE THE QUALITY OF THE INFORMATION IT MAKES AVAILABLE. QUECTEL DOES NOT MAKE ANY WARRANTY AS TO THE INFORMATION CONTAINED HEREIN, AND DOES NOT ACCEPT ANY LIABILITY FOR ANY INJURY, LOSS OR DAMAGE OF ANY KIND INCURRED BY USE OF OR RELIANCE UPON THE INFORMATION. ALL INFORMATION SUPPLIED HEREIN IS SUBJECT TO CHANGE WITHOUT PRIOR NOTICE.

COPYRIGHT

About the Document

History

Revision	Date	Author	Description
1.0	2018-04-07	Mike ZHOU	Initial

Contents

1 Introduction

This document introduces the use and introduction of WiFi related API.

2 Introduction of WiFi Related Functions

This document applies to FC20/AF20, about WiFi API specification for OpenLinux project.

FC20/AF20 can work in 4 ways, which are:

(1) **AP**
Only open one hot spot, and support 2.4GHz and 5GHz frequency band, which can only start one frequency band at one time.

(2) **STA**
Used as a wireless card, can connect the 2.4GHz and 5GHz band hot spots.

(3) **AP and STA Coexist**
Single Hotspot mode and STA mode can run simultaneously.

(4) **Dual AP**
Can run two hot spots at the same time, the most commonly using scenario: set the same SSID, one cover 2.4GHz band and another cover 5GHz band.

3 Interface Descriptions

Linux does not start WiFi function by default, interfaces provided are as followings

(1) int ql_wifi_enable(void)
Enable WiFi function, this function will load WiFi corresponding driver file, and run WiFi daemon.

(2) int ql_wifi_disable(void)
Disable WIFI function, will stop WIFI daemon and uninstall WIFI corresponding driver files.

(3) int ql_wifi_work_mode_set(ql_wifi_work_mode_e mode)
int ql_wifi_work_mode_get(ql_wifi_work_mode_e *mode)
Set up and get the WiFi mode, the mode refers to: AP, STA mode, AP and STA coexistence mode and dual AP. The macro definition is as follows:

```
typedef enum {
    QL_WIFI_WORK_MODE_STA = 0,        /* WiFi is in STA Mode */
    QL_WIFI_WORK_MODE_AP0,            /* WiFi is in AP Mode */
    QL_WIFI_WORK_MODE_AP0_STA,        /* WiFi is in AP/STA Mode */
    QL_WIFI_WORK_MODE_AP0_AP1         /* WiFi is in AP/AP Mode */
} ql_wifi_work_mode_e;
```

NOTE

The later chapters will describe this function in detail.

(4) int ql_wifi_ap_ssid_set(ql_wifi_ap_index_e idx, char *ssid)
int ql_wifi_ap_ssid_get(ql_wifi_ap_index_e idx, char *ssid)
Set up and get AP name (SSID), the maximum length is 32 bytes (Chinese encoding is not supported for the time being).

(5) int ql_wifi_ap_mode_set(ql_wifi_ap_index_e idx, ql_wifi_mode_type_e mode)
int ql_wifi_ap_mode_get(ql_wifi_ap_index_e idx, ql_wifi_mode_type_e *mode)
Set up and get the AP mode, macro definition is as following:

```
typedef enum {
    QL_WIFI_MODE_80211B = 0,          /* IEEE 802.11b (2.4 GHz) */
    QL_WIFI_MODE_80211BG,             /* IEEE 802.11bg (2.4 GHz) */
    QL_WIFI_MODE_80211BGN,            /* IEEE 802.11bgn (2.4 GHz) */
    QL_WIFI_MODE_80211A,              /* IEEE 802.11a (5 GHz) */
    QL_WIFI_MODE_80211AN,              /* IEEE 802.11an (5 GHz) */
    QL_WIFI_MODE_80211AC               /* IEEE 802.11ac (5 GHz) */
} ql_wifi_mode_type_e;
```

NOTE

This working way requires WiFi chip support. For details, you can check the related documents of the WiFi chip. FC20 currently supports 11b / g / n / a / ac.

(6) int ql_wifi_ap_bandwidth_set(ql_wifi_ap_index_e idx, ql_wifi_bandwidth_type_e bandwidth)
int ql_wifi_ap_bandwidth_get(ql_wifi_ap_index_e idx, ql_wifi_bandwidth_type_e *bandwidth)

Set up and get the AP bandwidth, macro definition is as following:

```
typedef enum {
    QL_WIFI_BANDWIDTH_HT20 = 0,
    QL_WIFI_BANDWIDTH_HT40,
    QL_WIFI_BANDWIDTH_HT80         /* only support 11AC */
} ql_wifi_bandwidth_type_e;
```

NOTE

This bandwidth requires WiFi chip support. For details, you can check the related documents of the WiFi chip. 11b / g / don’t support bandwidth configuration.

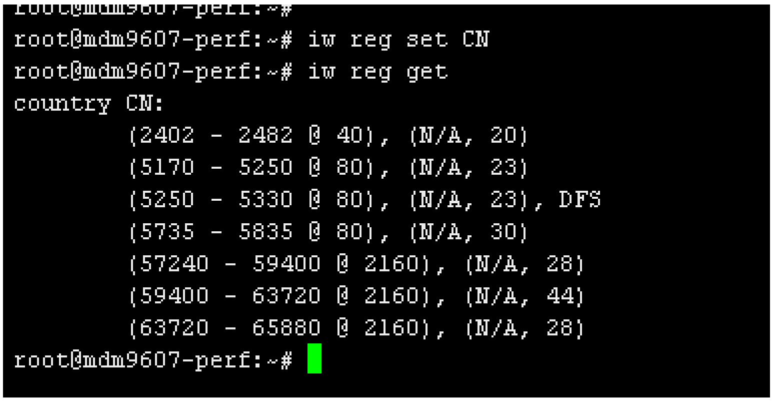
(7) int ql_wifi_ap_channel_set(ql_wifi_ap_index_e idx, int channel)
int ql_wifi_ap_channel_get(ql_wifi_ap_index_e idx, int *channel)

Set and get the current AP channel.

1) Value 0 is automatic channel selection, only supports FC20 / AF20 chip.
2) 2.4GHz band, support 1 ~ 13.
3) 5GHz band, support setting:
36/40/44/48/52/56/60/64/100/104/108/112/116/120/124/128/132/136/140/144/149/153/157/161/165/175/181

NOTE

The available value of hotspot channel is by definition of each country(currently only support China Country Code), iw command can be used to query the channel setting of current country. The following picture is taking FC20 as example.



```
root@mdm9607-perf:~#
root@mdm9607-perf:~# iw reg set CN
root@mdm9607-perf:~# iw reg get
country CN:
        (2402 - 2482 @ 40), (N/A, 20)
        (5170 - 5250 @ 80), (N/A, 23)
        (5250 - 5330 @ 80), (N/A, 23), DFS
        (5735 - 5835 @ 80), (N/A, 30)
        (57240 - 59400 @ 2160), (N/A, 28)
        (59400 - 63720 @ 2160), (N/A, 44)
        (63720 - 65880 @ 2160), (N/A, 28)
root@mdm9607-perf:~#
```

(8) int ql_wifi_ap_auth_set(ql_wifi_ap_index_e idx, ql_wifi_ap_auth_s *auth)
int ql_wifi_ap_auth_get(ql_wifi_ap_index_e idx, ql_wifi_ap_auth_s *auth)
Set and get encryption way of AP, support setting OPEN and WPA-PSK, the macro definition is as following:

```
typedef enum {
    QL_WIFI_AUTH_OPEN = 0,
    QL_WIFI_AUTH_WPA_PSK
} ql_wifi_auth_e;
```

(9) int ql_wifi_ap_max_sta_set(ql_wifi_ap_index_e idx, int max_sta_num)
int ql_wifi_ap_max_sta_get(ql_wifi_ap_index_e idx, int *max_sta_num)
Set and get the number of current maximum terminal connections. FC20 / AF20 maximum support 16.

(10) int ql_wifi_ap_start(ql_wifi_ap_index_e idx)
int ql_wifi_ap_stop(ql_wifi_ap_index_e idx)
int ql_wifi_ap_restart(ql_wifi_ap_index_e idx)
Start / stop / restart AP control process, this function will not operate WiFi driver files.

(11) int ql_wifi_sta_ssid_set(char *ssid)
int ql_wifi_sta_ssid_get(char *ssid)
Connect and disconnect the hotspot name which STA mode required.

(12) int ql_wifi_sta_auth_set(ql_wifi_sta_auth_s *auth)
int ql_wifi_sta_auth_get(ql_wifi_sta_auth_s *auth)

Set and get the hotspot encryption way which STA mode connection required.

(13) int ql_wifi_sta_connect(void)
int ql_wifi_sta_disconnect(void)
Connect and disconnect the hotspot which STA mode connection required.

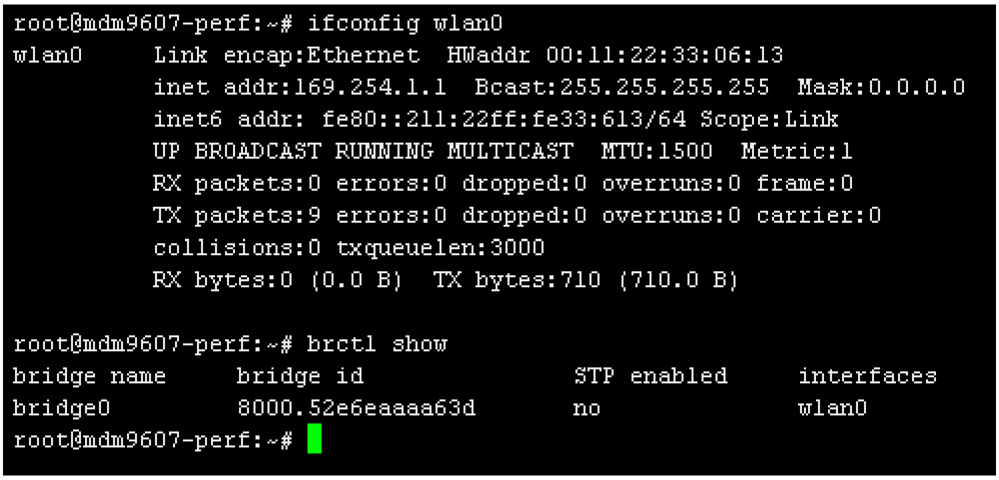
(14) int ql_wifi_sta_status(ql_wifi_station_status_e *status)
Check whether STA mode connect hotspot.

4 Example

Please refer to **example/sgmii/example_wifi.c.**

4.1. AP Mode

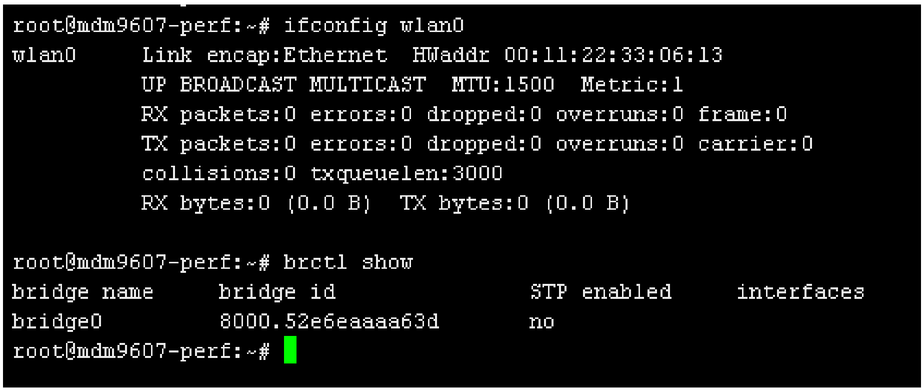
When enable this mode, wlan0 network interface will be created and added onto bridge0 (provide dhcpserver function and data forwarding).



```
root@mdm9607-perf:~# ifconfig wlan0
wlan0     Link encap:Ethernet  HWaddr 00:11:22:33:06:13
          inet addr:169.254.1.1  Bcast:255.255.255.255  Mask:0.0.0.0
          inet6 addr: fe80::211:22ff:fe33:613/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:9 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:3000
          RX bytes:0 (0.0 B)  TX bytes:710 (710.0 B)

root@mdm9607-perf:~# brctl show
bridge name     bridge id               STP enabled     interfaces
bridge0         8000.52e6eaaaa63d       no              wlan0
root@mdm9607-perf:~#
```

4.2. STA Mode

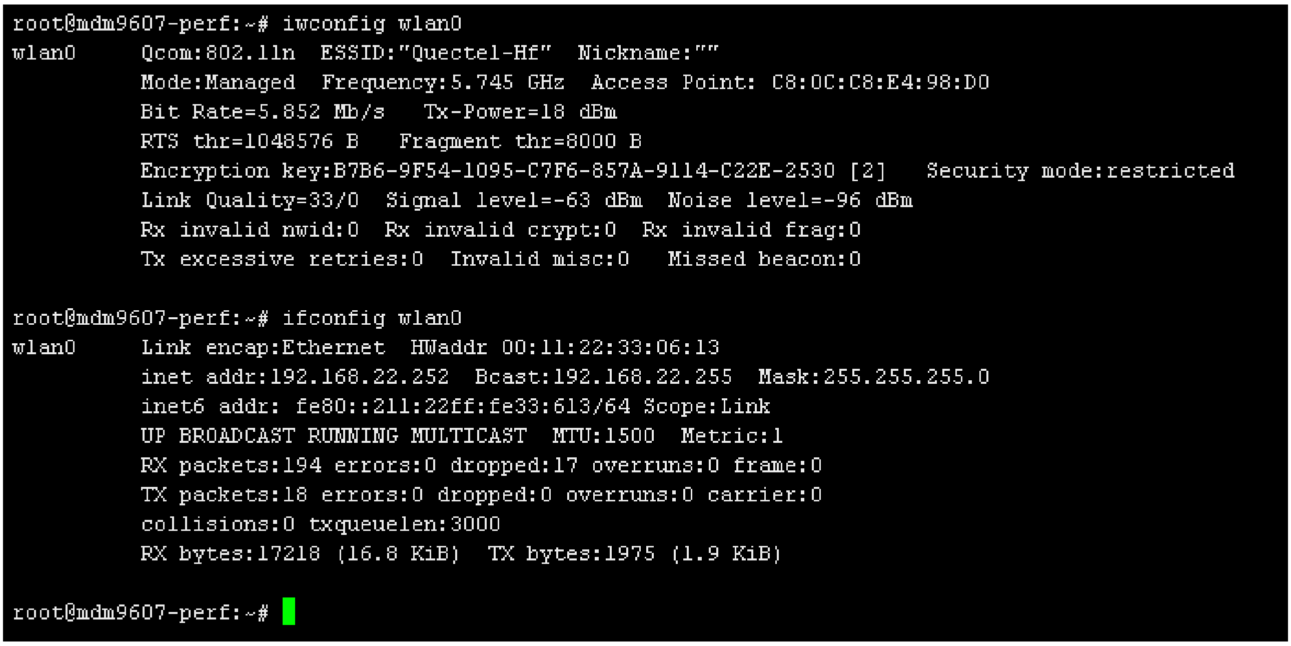
When enable this mode, wlan0 network interface will be created and won’t be added onto bridge0.



```
root@mdm9607-perf:~# ifconfig wlan0
wlan0     Link encap:Ethernet  HWaddr 00:11:22:33:06:13
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:3000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

root@mdm9607-perf:~# brctl show
bridge name     bridge id               STP enabled     interfaces
bridge0         8000.52e6eaaaa63d       no
root@mdm9607-perf:~#
```

Search hotspot API has not yet been achieved (**iwlist wlan0 scanning** command can scan the surrounding hot spots).

After this mode is enabled, it will create udchpd process to obtain the IPv4 address and set it onto wlan0 interface.

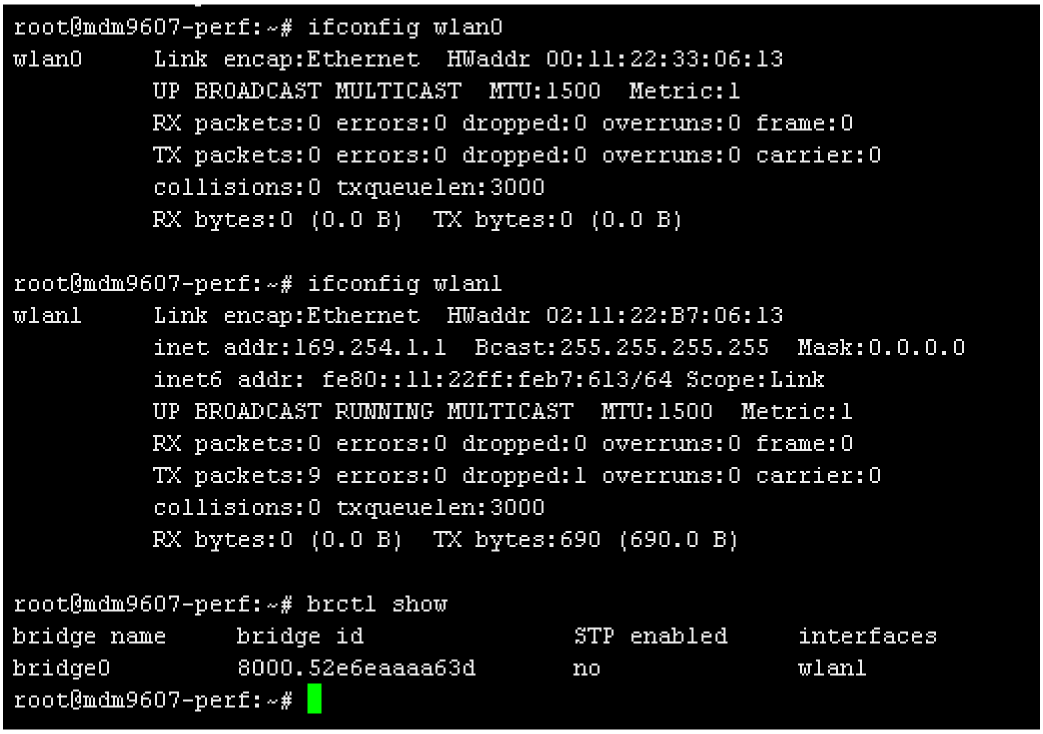


```
root@mdm9607-perf:~# iwconfig wlan0
wlan0     Qcom:802.11n  ESSID:"Quectel-Hf"  Nickname:""
          Mode:Managed  Frequency:5.745 GHz  Access Point: C8:0C:C8:E4:98:D0
          Bit Rate=5.852 Mb/s   Tx-Power=18 dBm
          RTS thr=1048576 B   Fragment thr=8000 B
          Encryption key:B7B6-9F54-1095-C7F6-857A-9114-C22E-2530 [2]   Security mode:restricted
          Link Quality=33/0  Signal level=-63 dBm  Noise level=-96 dBm
          Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
          Tx excessive retries:0  Invalid misc:0   Missed beacon:0

root@mdm9607-perf:~# ifconfig wlan0
wlan0     Link encap:Ethernet  HWaddr 00:11:22:33:06:13
          inet addr:192.168.22.252  Bcast:192.168.22.255  Mask:255.255.255.0
          inet6 addr: fe80::211:22ff:fe33:613/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:194 errors:0 dropped:17 overruns:0 frame:0
          TX packets:18 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:3000
          RX bytes:17218 (16.8 KiB)  TX bytes:1975 (1.9 KiB)

root@mdm9607-perf:~#
```

Use **ql_wifi_sta_status** API or **iwconfig** command to check whether the connection is succeed.

4.3. AP and STA Coexistence Mode

When this mode enabled, will create wlan0 and wlan1 network interface, which wlan0 is as STA mode, wlan1 is for AP mode (wlan1 will be added into bridge0), as shown in the following picture.



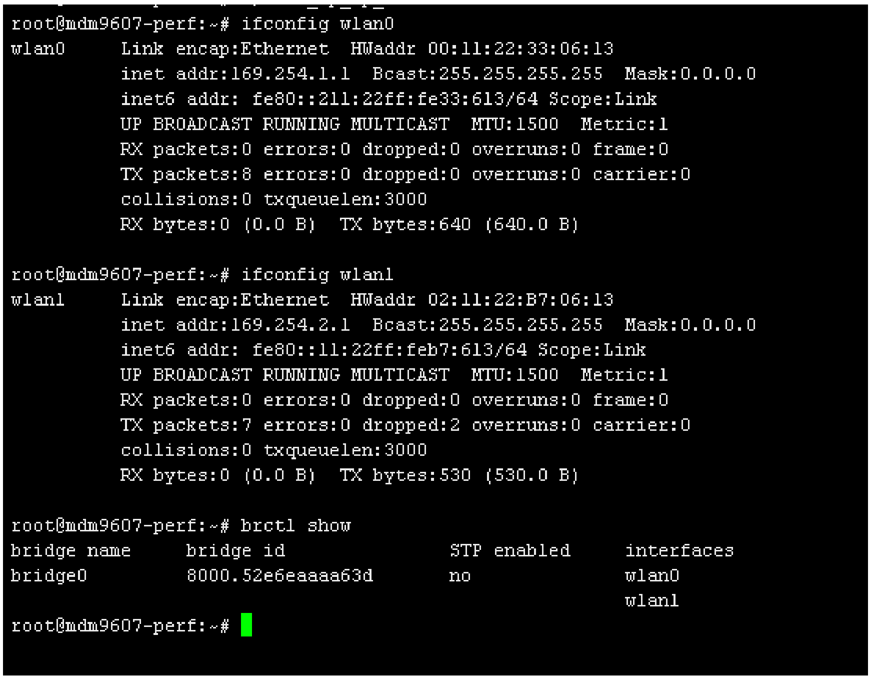
```
root@mdm9607-perf:~# ifconfig wlan0
wlan0     Link encap:Ethernet  HWaddr 00:11:22:33:06:13
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:3000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

root@mdm9607-perf:~# ifconfig wlan1
wlan1     Link encap:Ethernet  HWaddr 02:11:22:B7:06:13
          inet addr:169.254.1.1  Bcast:255.255.255.255  Mask:0.0.0.0
          inet6 addr: fe80::11:22ff:feb7:613/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:9 errors:0 dropped:1 overruns:0 carrier:0
          collisions:0 txqueuelen:3000
          RX bytes:0 (0.0 B)  TX bytes:690 (690.0 B)

root@mdm9607-perf:~# brctl show
bridge name     bridge id               STP enabled     interfaces
bridge0         8000.52e6eaaaa63d       no              wlan1
root@mdm9607-perf:~#
```

NOTE

Under this mode, dial-up via QCMAP and default exit is directly to STA port.

4.4. Dual AP Mode

When this mode enabled, will create wlan0 and wlan1 network interface, which wlan0 is as AP0 mode, wlan1 is as AP1 mode hotspot (wlan0 and wlan1 will both be added into bridge0), as shown in the following picture.



```
root@mdm9607-perf:~# ifconfig wlan0
wlan0     Link encap:Ethernet  HWaddr 00:11:22:33:06:13
          inet addr:169.254.1.1  Bcast:255.255.255.255  Mask:0.0.0.0
          inet6 addr: fe80::211:22ff:fe33:613/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:8 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:3000
          RX bytes:0 (0.0 B)  TX bytes:640 (640.0 B)

root@mdm9607-perf:~# ifconfig wlan1
wlan1     Link encap:Ethernet  HWaddr 02:11:22:B7:06:13
          inet addr:169.254.2.1  Bcast:255.255.255.255  Mask:0.0.0.0
          inet6 addr: fe80::11:22ff:feb7:613/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:7 errors:0 dropped:2 overruns:0 carrier:0
          collisions:0 txqueuelen:3000
          RX bytes:0 (0.0 B)  TX bytes:530 (530.0 B)

root@mdm9607-perf:~# brctl show
bridge name     bridge id               STP enabled     interfaces
bridge0         8000.52e6eaaaa63d       no              wlan0
                                                        wlan1
root@mdm9607-perf:~#
```