Realtek WiFi concurrent mode Introduction.

2012/10/29	Aries Lee	Ver 0.1
2012/11/06	Aries Lee	Ver 0.9
2016/08/19	Gary Hsieh	Ver 1.0
2016/10/04	Gary Hsieh	Ver 1.1

1.WHAT'S CONCURRENT MODE

This feature registers 2 wireless network devices in OS (wlan0, wlan1) and those two interfaces share the same hardware device.
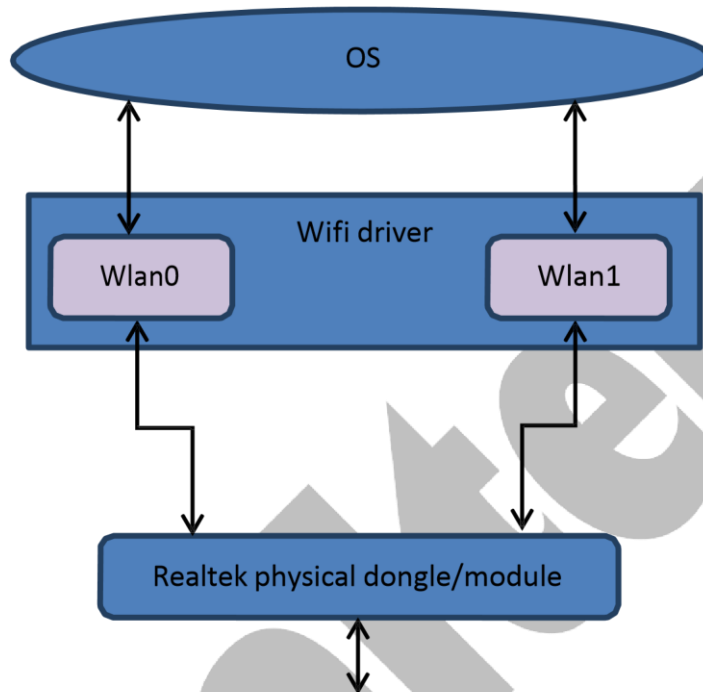
This feature allows performing 2 separate wireless tasks at the same time with single hardware device.
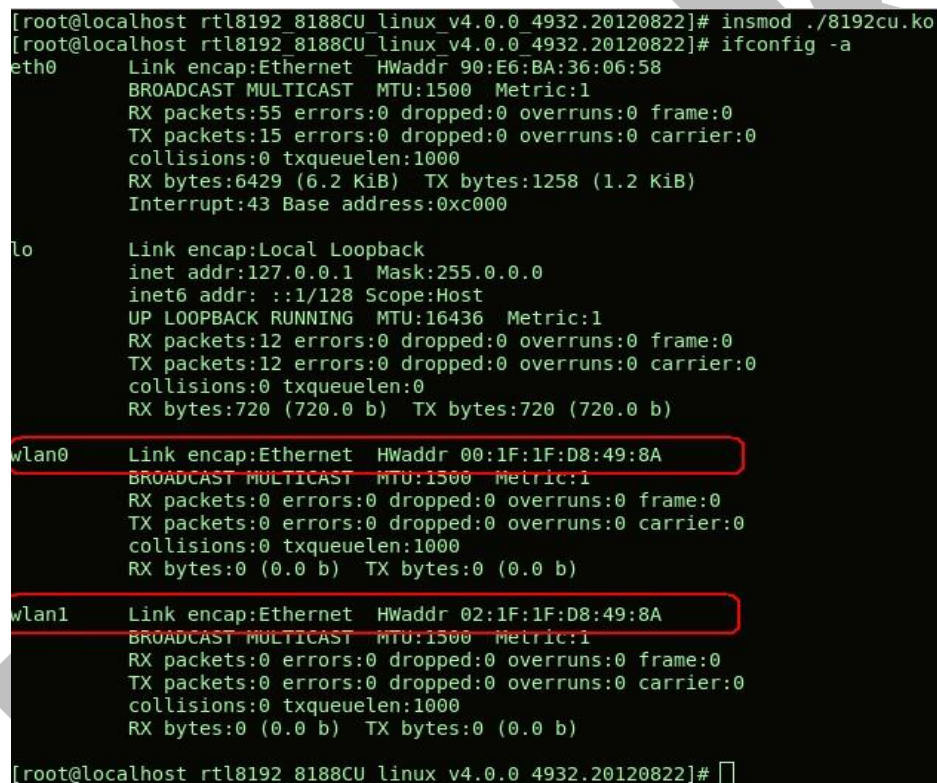
For example:

The system can perform station mode to connect with an AP router and access internet, at the same time, it also perform the p2p connection with another p2p device and get the resource of this p2p device.

Note: Realtek wifi only support 3 concurrent combination

1. **Station mode + Station mode**
2. **Station mode + AP mode**
3. **Station mode + P2P mode**

2.HOW TO ENABLE CONCURRENT MODE

In Realtek wifi driver source code folder, edit "Makefile", change "CONFIG_CONCURRENT_MODE = n" to "CONFIG_CONCURRENT_MODE = y" then rebuild and insmod the driver, done. You can verify if it works by typing "ifconfig – a" command, It is supposed to show two wireless interfaces, and MAC address of

```
[root@localhost rtl8192_8188CU_linux_v4.0.0_4932.20120822]# insmod ./8192cu.ko
[root@localhost rtl8192_8188CU_linux_v4.0.0_4932.20120822]# ifconfig -a
eth0      Link encap:Ethernet  HWaddr 90:E6:BA:36:06:58
          BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:55 errors:0 dropped:0 overruns:0 frame:0
          TX packets:15 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:6429 (6.2 KiB)  TX bytes:1258 (1.2 KiB)
          Interrupt:43 Base address:0xc000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:12 errors:0 dropped:0 overruns:0 frame:0
          TX packets:12 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:720 (720.0 b)  TX bytes:720 (720.0 b)

wlan0     Link encap:Ethernet  HWaddr 00:1F:1F:D8:49:8A
          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:1000
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)

wlan1     Link encap:Ethernet  HWaddr 02:1F:1F:D8:49:8A
          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:1000
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)

[root@localhost rtl8192_8188CU_linux_v4.0.0_4932.20120822]# []
```

secondary interface is nearly the same except the bit1 of byte0 was assigned as 1.

Base on the past experiences, we recommend our clients to verify whole system steps by steps – first check station mode, then check AP mode, finally station + AP mode (concurrent mode).

- Please reference "*wpa_cli_with_wpa_supplicant.pdf*" to understand how to run in station mode, "*Quick_Start_Guide_for_Station_Mode.pdf*" is also a good choice.

- Please reference "*Quick_Start_Guide_for_SoftAP.pdf*" to understand how to run in AP mode.

3. PORTING CONCURRENT MODE IN ANDROID FRAMEWORK.

You can read the "readme.txt" in Realtek software SDK folder for more porting/testing information.

For example:

If the target system is Android JB, it will indicate you to follow the steps of "*Realtek_WiFi_SDK_for_Android_JB.pdf*", you can easily accomplish the porting task.

Attention: The concurrent architecture starts at Android JB. For older version, you need to modify the Android framework by yourself.

4. HOW TO ENABLE MCC MODE

In Realtek wifi driver source code folder, edit "Makefile", change following lines:

CONFIG_MCC_MODE = n => CONFIG_MCC_MODE = y

CONFIG_CONCURRENT_MODE = n => CONFIG_CONCURRENT_MODE = y

The rest of steps are the same as part 2 "HOW TO ENABLE CONCURRENT MODE".

MCC mode is based on concurrent mode, but the two wireless interfaces can work under two different channels.

Two wireless interfaces actually share the same physical hardware device, and therefore the MCC mechanism should switch channel according to following mcc parameters:

rtw_mcc_policy_table_idx   = default using 3

rtw_mcc_interval   = 100   (Beacon interval)

rtw_mcc_duration   = duration for STA   default 50

rtw_mcc_interval - rtw_mcc_duration   = duration for softAP   default 50

rtw_mcc_start_time_offset   = (rtw_mcc_interval - rtw_mcc_duration) / 2

rtw_mcc_tsf_sync_offset   = rtw_mcc_start_time_offset + 10

If you want to adjust the durations for two interface, Insmod with following command (STA/softAP = 80/20): sudo insmod ./8821cu.ko rtw_mcc_policy_table_idx=0 rtw_mcc_duration=80 rtw_mcc_tsf_sync_offset=20 rtw_mcc_start_time_offset=10

4.1 Dynamic MCC Duration

 If support dynamic MCC duration, it will be default enable when enter MCC mode.
Reference commands:

- **"cat /proc/net/rtl8821cu/wlan1/mcc/mcc_dynamic_duration"** => show dynamic mcc duration enable(1)/disable(0)

```
gary@gary-ubuntu:~/bluez-5.41$ cat /proc/net/rtl8821cu/wlan1/mcc/mcc_dynamic_duration
mcc dynamic duration = 1
```

- **"echo 0 > /proc/net/rtl8821cu/wlan1/mcc/mcc_dynamic_duration"** => 1: enable 0: disable

- **"cat /proc/net/rtl8821cu/wlan1/mcc/mcc_pattern"** => show current pattern & pattern table

```
gary@gary-ubuntu:~/bluez-5.41$ cat /proc/net/rtl8821cu/wlan1/mcc/mcc_pattern
Current pattern table index = 3
duration      ,tsf sync offset      ,start time offset      ,interval      ,guard offset0  ,guard offset1
20            ,50                   ,40                     ,100          ,0              ,0              ⇑  < 3  GO prefered
30            ,45                   ,35                     ,100          ,0              ,0
40            ,40                   ,30                     ,100          ,0              ,0              ⇐ default index 3 (50/50)
50            ,35                   ,25                     ,100          ,0              ,0
60            ,30                   ,20                     ,100          ,0              ,0
70            ,25                   ,15                     ,100          ,0              ,0              ⇓  > 3 STA prefered
80            ,20                   ,10                     ,100          ,0              ,0
```

- **"echo 6 > /proc/net/rtl8821cu/wlan1/mcc/mcc_pattern"** => control pattern when dynamic mcc duration is disable

5. Q&A

Q1: Why does the system still has only one interface after insmod the rebuilding module?

A: Please make sure you edit the correct autoconf.h and insmod the correct module, and also provide your autoconf.h and kernel message log to us.

Q2: (In concurrent mode)Everything is fine when I only start hostapd, but when I start running a station mode in another interface in the same time, the hostapd will disconnect for a second then be reconnected again, however ,the channel is differ from before setting. Is anything wrong?

A: Don't worry, it is fine! As I mention before, those two wireless interfaces actually share the same physical hardware device. **That means those 2 wireless interfaces must work under the same channel.** Our rule is AP/P2P interface should follow the channel of station interface, if both interface are running station mode, the connected APs MUST be the same channel.

Q3: How is the throughput in 2 wireless interfaces in concurrent mode?

A: Because there is only one physical hardware device, the two wireless interface (wlan0, wlan1) will share the transmit bandwidth,

for example:

Assume the throughput limitation of current environment is 85Mb/s, then the throughput of wlan0 + the throughput of wlan1 is basically equal or smaller than 85Mb/s.

Q4: Which wireless interface can run in station mode? Which wireless interface can run in AP mode?

A: We recommend that run station mode in wlan0, run AP or P2P mode in wlan1.