RALINK TECHNOLOGY, CORP.

RALINK RT61 PCI/miniPCI/CardBus Wireless Card

SOFTWARE DRIVER RELEASE NOTE

Contents

1. Release Note

1.1. CHANGE HISTORY

1. Add support for fragmentation of EAP packet.

2. Add Countermeasure support.
   The older version only support countermeasure in fragmentation case.

3. Bug fix - register net device using error MAC address
   This is one of differences between MSSID and MBSSID

4. Bug fix - MAC address report error when using iwconfig command
   This will occur when multiple BSSID.

5. Bug fix - 'own_addr' needs to save all BSSID's MAC address (802.1x)
   This is one of differences between MSSID and MBSSID

6. Unify the usage of pci_map_single and pci_map_page
   The difference of these two routines are not utilized by this module

7. Add initialize the MBSSID interface in skb buffer.

8. Adjust member's sequence of structure MLME_QUEUE_ELEM to solve alignment issue in some platform.

9. Modify the message3 of WPA 4-way handshake to solve windows zero config can not link to WPA2PSK-TKIPAES case

10. Support non-copy when frame aggregation

11. Add firmware loading error check.

12. **Support RT2561S,**
    1.) **RT2561S (DeviceID=0x0301) : high throughput version**
    2.) **RT2561 (DeviceID=0x0302) : cost down version**
    3.) **RT2661 (DeviceID=0x0401) : smart antenna version**
    Please remember to update firmware.

13. Add "DisableOLBC" control, turn it on to eliminate the B/G AP co-channel throughput interference issue. However, it needs to be disable when doing Wi-Fi test.

14. v1.0.1.0. changed:
    1.) Interface support and bugs fix for WMM (Under testing).
    2.) DFS support.
    3.) Support WPA over WDS.
    4.) Bug fix for two WPAPSK-STAs cause AP crash.
    5.) Bug fix for BG-STAs will link up with B-only-AP.
    6.) Fix compatiblility issue in 802.11d.

15. v1.0.2.0 changed:
- **1.)** Support Tx power range from 36 to -6
- **2.)** Fix WDS bug
- **3.)** Fine tune BBP tuning to fix long distance issue
- **4.)** Bypass RATE_6 and RATE_9
- **5.)** Update DFS to support Japan's spec.
- **6.)** Update firmware version as v0.9
- **7.)** Fix compatibility issue in Centrino chipset when WPAPSK
- **8.)** Fix TxBurst issue in Centrino chipset when B mode

16. v1.0.3.0 changed:
- **1.)** Support Tx power range from 36 to -6
- **2.)** Fine tune BBP tuning for long distance.
- **3.)** Bypass RATE_6 and RATE_9
- **4.)** Update DFS to support Japan's spec.

17. v1.0.4.0. changed:
- **1.)** Bug fix - small allocated buffer in ACL(Access Control List) causes driver crashed.
- **2.)** Bug fix - small allocated buffer in reading MAC, BBP EEPROM causes driver crashed.
- **3.)** Bug fix - mixed encryption algorithm can not be link up in WPA2PSK with AES
- **4.)** Bug fix - turn on antenna diversity in SoftAP will cause low performance

18. v1.0.5.0. changed:
- **1.)** Add radio on/off function
- **2.)** Fix PMK key cache timeout issue
- **3.)** Replace strtok with rstrtok, beacuse strtok is obsolote in linux kernel 2.6
- **4.)** Replace verify_area with access_ok, beacuse verify_area is obsolote
- **5.)** make RT61AP to run in linux 2.4 and linux 2.6
- **6.)** Modify rtmp_task.c to comply with linux 2.6
- **7.)** When **THREAD_ISR** was defined, caller must disable IRQ (RTMP_IRQ_LOCK()) before using
- **8.)** Fix WPAPSK TKIP Group Key
- **9.)** Fix system crash while running ATE command(TXFRAME) twice.

19. v1.0.6.0. changed:
- **1.)** Fix the issue sta can't connect after ifconfig ra0 down and ifconfig ra0 up
- **2.)** Update for pass WMM certification
- **3.)** Fix OPEN-WEP-802.1x-MD5 authentication fails issue

20. v1.0.7.0. changed:
- **1.)** Add WMM-APSD support **[Valid on WmmCapable=1]**.
- **2.)** Add proprietary DLS protocol **[Valid on WmmCapable=1]**.
- **3.)** Add ATE PER(packet error rate) display per second.
- **4.)** Support RTL865x Tx Fast Path.
- **5.)** Fix ATE TxPower issue.
- **6.)** Fix Macintosh issue.
- **7.)** Fix Two Antenna average RSSI issue.
- **8.)** **Support Japan filter**.
- **9.)** Fix pci_map_single, pci_unmap_single unsync problem.
- **10.)** Fix bug : When NoForwarding is used in WDS mode, the WDS packets will be discard.
- **11.)** Implement auto select channel for A band.
- **12.)** Fix LED Mode Doesn't Match with EEPROM Setting.

21. v1.0.8.0. changed:
- **1.)** Support DFS function for Japan.
- **2.)** Support A band channel for Japan.
- **3.)** Fix bug: Centrino clients in B/G mode can't associate with RT61AP when country code is

set.
4.) Update .bin file (**version 1.1**).
5.) Modify LED mode.
6.) Fix bug: site survey display wrong issue.
7.) Enhance 802.1x dynamic wep keying.
8.) **AP Client** support (include BIG-ENDIAN platform).
9.) **SNMP** support.

22. v1.0.8.1. changed:
1.) For DFS, prevent to set previous channel by iwpriv command.
2.) Add a 300ms-timer to enqueue EAPoL-Start for WPAPSK, not RTMPusecDelay.
3.) The Fragment issue in WPAPSK-TKIP mode.
4.) Error handle for empty RSN_IS in association-request for RSNA security.
5.) Support new CE and FCC radar detection capability.
6.) The Traffic Indicator bit is set to 1 in TIM elements with a value of 0 in the DTIM Count field when one or more broadcast or multicast frames are buffered at the AP.
7.) Make some modification for AP-Client.

23. v1.0.9.0. changed:
1.) Create a sofeware queue for NULL frame transmission
2.) Support **IGMP-Snooping and Multicast filter**.
3.) Add new iwpriv command "CountryString" support countrycode map countryregion.
4.) Fix bug: AP-Client interface information display error by iwconfig.
5.) Correct R3 value, R3 value should be in range 0 ~ 31.
6.) Fix bug: In power-saving mode, M/Bcast frame should be sent when DTIM count is zero.
7.) Solve ATE issue and add ASIC model number check.
8.) Fix bug: It can't check multi-unicast cipher for STA's RSN_IE.
9.) Fix bug: WDS with lazy capability can't work in WPAPSK mode.
10.) Support **Multiple RADIUS server for MBSS**
11.) Fix bug: AP will freeze after radar detection miss-trigger.
12.) Reduce the radar detection tolerance.
13.) Update .bin file **(version 1.2**)

24. v1.1.0.0 changed:
1.) Add Carrier-Detection function.
2.) Fixed some bugs relative to DFS:
(1) AP freeze while enable 80211H in B/G band.
(2) AP sending too much switch-Announcements.
(3) Switch-Announcement didn't carry PS-Count.
3.) Fixed bug unable to handle IGMP v2 membership report message.
4.) Fix bug: DTIM of MBSS beacon is not sync.
5.) Fix bug: Beacon of MBSS can be started or stopped independently.
6.) Disable a function used in Adhoc mode. It will influence the beacon start transmission time in root AP mode.
7.) Exchange TIM element and WMM element in beacon frame.
8.) Bug fix for More data bit in legacy PS mode. More data bit = 0 in last dnlink PS packet from AP.
9.) Add new feature (EOSP length, it is optional in WMM spec.).
10.) WPS support
11.) Add some LLTD related ioctl functions.
12.) Fix bug: redundant virtual interfaces will be created after ifconfig down, up, down, up...
13.) Let AP forward all IGMP packets to Station.
14.) Remove IgmpGroupTxRate parameter.
15.) Remoe all iwpriv command relative to IgmpGroupTxRate.
16.) Reset "NoDataIdleCount" when AP successfully sending packet to STA.

25. v1.1.1.0 changed:

1.) Add RTSThreshold=0 support; Fix RTSThreshold bug.
2.) Fix TxPower adjust problem.
3.) Fix TxRate input bug in 4 BSSID case.
4.) Improve connectivity in noisy environment.
5.) Fix the compilation error for WIRELESS_EXT >= 21.
6.) Fix WDS bugs.
7.) Seperate the security settings of Main BSSID's and ApClient's. Now they can set different AuthMode and EncrypType.
8.) Change ApClient Keep Alive mechanism.
9.) ApClient dynamically switch the channel specified in channel-switch-announcement IE of beacon.
10.) Fix ApClient retry connection bug.
11.) Add the LED behaviors.
12.) Fix the bug of performance drop after switching mode to B then back to G or B/G mixed.
13.) Fix the bug of AutoChannelSelect out of channel range for specified region.
14.) Support per-device statistics counters.
15.) Fix WPS interoperability problem with other WPS devices.
16.) Add WPS patches to pass MS WCN.
17.) Support 64-bit platform.
18.) CountryRegion supports regions 0-7. CountryRegionABand supports regions 0-10.
19.) Update countrycode_vs_channel.txt and codes for the most updated country/channel lists.
20.) Improve IGMP snooping and fix its memory leakage bug.
21.) Fix the system hang problem in timer.

26. 1.1.2.0 changed:

1.) Modify DFS parameters.
2.) Improve UAPSD with new algorithm.
3.) Add WPS support to ApClient.
4.) Fix crash problem on Realtek 865x platform.
5.) Fix AP WPAPSK bug when APCLI_SUPPORT compiled in, but ApClient not enabled together with the last digit of the MAC address of the IC is odd number.
6.) WPS re-generate PIN-code modifications.
7.) Fix the bug that affects ApClient settings when WPS writes the settings to the profile.
8.) Fix IGMP snooping problem with IGMP v3.

2. README

RT61 a/b/g SoftAp driver for RT61 a/b/g, Ralink Tech Corp.

2.1. FEATURES

This RT61 a/b/g SoftAp driver implements wireless Access Point(AP) function and support four BSSID concurrently .

Through bridge service in linux, AP can access internet through some other interface, i.e. ethernet or others.

For authentication, this driver provides OPEN, SHARED, WPAPSK, 802.1x(called WPA/WPA2). and also support encryption methods like WEP, TKIP, AES and of course NONE.

If OPEN or SHARED, use NONE or WEP. If using WPA/WPA2 or WPAPSK/WPA2PSK and their combinations, use TKIP or AES.

Other combinations are not supported by this driver yet.

2.2. USAGE

This source code package can be use in Linux version after RedHat Linux 7.3.

2.2.1. Scripts

load	**load module to kernel**
unload	**unload module from kernel**
Configure	**config build linux version**
bridge_setup	**script for bridge setup**

2.2.2. Setup Sequence

1. Use 'chmod' command to change access right of following script files 'load', 'unload', 'Configure', 'bridge_setup'
2. Turn on or patch linux bridge package
3. $make config # config build linux os version
4. $make # compile driver source code
5. $cp RT2561.bin /etc/Wireless/RT61AP/ # copy firmware
6. $load # load/insmod module(rt61ap.o)
7. $bridge_setup # configure bridge

2.2.3. bridge_setup

```
/usr/sbin/brctl addbr br0
/usr/sbin/brctl addif br0 eth0
/usr/sbin/brctl addif br0 ra0
/sbin/ifconfig eth0 0.0.0.0
/sbin/ifconfig ra0 0.0.0.0
/sbin/ip link set br0 up
/sbin/ip addr add 192.168.5.234/24 brd + dev br0
/sbin/ip route add default via 192.168.5.254
```

2.2.4. load

```
/sbin/insmod rt61ap.o
/sbin/ifconfig ra0 inet 192.168.5.234 up
/sbin/route add default gw 192.168.5.254
```

2.2.5. unload

```
/sbin/ifconfig ra0 down
/sbin/rmmod rt61ap
```

3. Configuration

1. RT61 SoftAp driver can be configured via two interfaces,
   i.e. 1**) configuration file**, 2**). "iwpriv"** command

   1). RT61AP.dat is an example of configuration file.
   2). iwpriv usage, please refer to iwpriv_usage.txt.

2. Please put RT61AP.dat in **/etc/Wireless/RT61AP/RT61AP.dat**.

3. To change the file path, please change the definition in rt_config.h
   #define PROFILE_PATH **"/etc/Wireless/RT61AP/RT61AP.dat"**

4. To edit configuration file, please follow the rules below:
   1). add # at head for comment line
   2). syntax is 'Param'='Value'

5. Detail description and usage os each parameters as following sections.

3.1. RT61AP.dat Parameter List

#The word of "Default" must not be removed
Default

1. Basic Parameters:

CountryRegion=5
CountryRegionABand=7
CountryCode=
BssidNum=1
SSID=AP1
WirelessMode=0
TxRate=0
Channel=6
BasicRate=15
BeaconPeriod=100
DtimPeriod=1
TxPower=100
DisableOLBC=0
BGProtection=0
TxAntenna=
RxAntenna=
TxPreamble=0
RTSThreshold=2347
FragThreshold=2346
TxBurst=1
PktAggregate=0
TurboRate=0
NoForwarding=0
NoForwardingBTNBSSID=0
HideSSID=0
ShortSlot=1
AutoChannelSelect=0
AccessPolicy0=0
AccessControlList0=
AccessPolicy1=0
AccessControlList1=
AccessPolicy2=0
AccessControlList2=

AccessPolicy3=0
AccessControlList3=
HSCounter=0

2. WMM Parameters:

WmmCapable=0
APAifsn=3;7;1;1
APCwmin=4;4;3;2
APCwmax=6;10;4;3
APTxop=0;0;94;47
APACM=0;0;0;0
BSSAifsn=3;7;2;2
BSSCwmin=4;4;3;2
BSSCwmax=10;10;4;3
BSSTxop=0;0;94;47
BSSACM=0;0;0;0
AckPolicy=0;0;0;0
APSDCapable=0
DLSCapable=0

3. IEEE802.1h+d, Spectrum Management

MaxTxPowerLevel=16
IEEE80211H=0
CSPeriod=10

4. Security Policy Parameters

AuthMode=OPEN
EncrypType=NONE
WPAPSK=
PreAuth=0
RekeyMethod=DISABLE
RekeyInterval=0
PMKCachePeriod=10
DefaultKeyID=1
Key1Type=0
Key1Str=
Key2Type=0
Key2Str=
Key3Type=0
Key3Str=
Key4Type=0
Key4Str=

5. WDS Parameters

WdsEnable=0
WdsEncrypType=NONE
WdsList=
WdsKey=

6. 802.1X Authenticator

IEEE8021X=0
RADIUS_Server=192.168.2.3
RADIUS_Port=1812
RADIUS_Key=ralink
own_ip_addr=192.168.5.234
Ethifname=eth0

7. AP Client Parameters

ApCliEnable
ApCliSsid
ApCliBssid
ApCliAuthMode
ApCliEncrypType
ApCliWPAPSK
ApCliDefaultKeyID
ApCliKey1Type
ApCliKey1Str
ApCliKey2Type
ApCliKey2Str
ApCliKey3Type
ApCliKey3Str
ApCliKey4Type
ApCliKey4Str

8. WPS Parameters

WscConfMode=0
WscConfStatus=1

3.2. Iwpriv Command List

1. Basic Parameters:

DriverVersion
CountryRegion
CountryRegionABand
SSID
HideSSID
WirelessMode
Channel
TxRate
BasicRate
BeaconPeriod
DtimPeriod
TxPower
RadioOn
BGProtection
DisableOLBC
TxAntenna
RxAntenna
TxPreamble
ShortSlot
TxBurst
PktAggregate
TurboRate
RetryLimit
TxQueueSize
RTSThreshold
FragThreshold
AccessPolicy
AccessControlList
NoForwarding
NoForwardingBTNBSSID
Debug
HSCounter
ResetCounter
stat
SiteSurvey
get_site_survey
get_mac_table
chk_asic_ver

2. WMM Parameters:

WmmCapable

3. 802.1X Authenticator

IEEE8021X

4. IEEE802.1d, Regular Domain

CountryCode
CountryString

5. IEEE802.1h, Spectrum Management

IEEE80211H
RDDurRegion
CSPeriod

4. Basic Parameters

These parameters are basic parameters and have to set, otherwise default value used.

4.1. Supported Parameters in RT61AP.dat

4.1.1. CountryRegion=value

value

0: channel 1-11
1: channel 1-13
2: channel 10-11
3: channel 10-13
4: channel 14
5: channel 1-14
6: channel 3-9
7: channel 5-13

4.1.2. CountryRegionABand=value

value

0: channel 36, 40, 44, 48, 52, 56, 60, 64, 149, 153, 157, 161, 165
1: channel 36, 40, 44, 48, 52, 56, 60, 64, 100, 104, 108, 112, 116, 120, 124, 128, 132, 136, 140
2: channel 36, 40, 44, 48, 52, 56, 60, 64
3: channel 52, 56, 60, 64, 149, 153, 157, 161
4: channel 149, 153, 157, 161, 165
5: channel 149, 153, 157, 161
6: channel 36, 40, 44, 48
7: channel 36, 40, 44, 48, 52, 56, 60, 64, 100, 104, 108, 112, 116, 120, 124, 128, 132, 136, 140, 149, 153, 157, 161, 165
8: channel 52, 56, 60, 64
9: channel 34, 38, 42, 46
10: channel 34, 36, 38, 40, 42, 44, 46, 48, 52, 56, 60, 64

4.1.3. CountryCode=value

Value

2 characters, like TW for Taiwan.

Please refer to ISO3166 code list for other countries and can be found at http://www.iso.org/iso/en/prods-services/iso3166ma/02iso-3166-code-lists/list-en1.html#sz

4.1.4. BssidNum=value

Value

1~4: multiple BSSID number

Note:

1. MAC Address alignment on MBSSID.
   - **a.** Main BSSID have to insure MAC address is multiple of 2s on 2-BSSIDs' application.
   - **b.** Main BSSID have to insure MAC address is multiple of 4s on 4-BSSIDs' application.
2. Example 4 BSSIDs:

Align	1st	2nd	3rd	4th
0x00	AA-BB-CC-DD-EE-F0	AA-BB-CC-DD-EE-F1	AA-BB-CC-DD-EE-F2	AA-BB-CC-DD-EE-F3
0x04	AA-BB-CC-DD-EE-F4	AA-BB-CC-DD-EE-F5	AA-BB-CC-DD-EE-F6	AA-BB-CC-DD-EE-F7
0x08	AA-BB-CC-DD-EE-F8	AA-BB-CC-DD-EE-F9	AA-BB-CC-DD-EE-FA	AA-BB-CC-DD-EE-FB

0x0C	AA-BB-CC-DD-EE-FC	AA-BB-CC-DD-EE-FD	AA-BB-CC-DD-EE-FE	AA-BB-CC-DD-EE-FF

3. Refer to data sheet for detail.
   a. MAC_CSR3, MAC_CSR4 and MAC_CSR5.
   b. SEC_CSR4.
   c. Security Key Table Layout.

4.1.5. SSID=value

value

1~32 ascii characters.

4.1.6. WirelessMode=value

value

0: 802.11 B/G mixed
1: 802.11 B only
2: 802.11 G only
3: 802.11 A only

4.1.7. TxRate=value

Value
A mode,

0: Auto
1: 6 Mbps
2: 9 Mbps
3: 12 Mbps
4: 18 Mbps
5: 24 Mbps
6: 36 Mbps
7: 48 Mbps
8: 54 Mbps
9: 6 Mbps
10: 9 Mbps // Auto tune TxRate between 6 and 9
11: 12 Mbps // Auto tune TxRate between 6, 9 and 12
12: 18 Mbps // Auto tune TxRate between 6, 9, 12 and 18
13: 24 Mbps // Auto tune TxRate between 6, 9, 12, 18 and 24
14: 36 Mbps // Auto tune TxRate between 6, 9, 12, 18, 24 and 36
15: 48 Mbps // Auto tune TxRate between 6, 9, 12, 18, 24, 36 and 48
16: 54 Mbps // Auto tune TxRate between 6, 9, 12, 18, 24, 36, 48 and 54

B/G mode

0: Auto
1: 1 Mbps
2: 2 Mbps
3: 5.5 Mbps
4: 11 Mbps
5: 6 Mbps //WirelessMode must be 0
6: 9 Mbps //WirelessMode must be 0
7: 12 Mbps //WirelessMode must be 0
8: 18 Mbps //WirelessMode must be 0
9: 24 Mbps //WirelessMode must be 0
10: 36 Mbps //WirelessMode must be 0
11: 48 Mbps //WirelessMode must be 0
12: 54 Mbps //WirelessMode must be 0
13: 1 Mbps
14: 2 Mbps // Auto tune TxRate between 1 and 2
15: 5.5 Mbps // Auto tune TxRate between 1 and 2 and 5.5
16: 11 Mbps // Auto tune TxRate between 1,2,5.5,11
17: 6 Mbps // Auto tune TxRate between 1,2,5.5,11,6
18: 9 Mbps // Auto tune TxRate between 1,2,5.5,11,6,9
19: 12 Mbps // Auto tune TxRate between 1,2,5.5,11,6,9,12
20: 18 Mbps // Auto tune TxRate between 1,2,5.5,11,6,9,12,18
21: 24 Mbps // Auto tune TxRate between 1,2,5.5,11,6,9,12,18,24
22: 36 Mbps // Auto tune TxRate between 1,2,5.5,11,6,9,12,18,24,36
23: 48 Mbps // Auto tune TxRate between 1,2,5.5,11,6,9,12,18,24,36,48
24: 54 Mbps // Auto tune TxRate between 1,2,5.5,11,6,9,12,18,24,36,48,54

TxRate			
Mode	**Value**		**Rate (unit: Mbps)**
802.11A		0.	Auto
	Fixed Rate	1.	6
		2.	9
		3.	12
		4.	18
		5.	24
		6.	36
		7.	48
		8.	54
	Rate Pool for Rate Switching	9.	6
		10.	6, 9
		11.	6, 9, 12
		12.	6, 9, 12, 18
		13.	6, 9, 12, 18, 24
		14.	6, 9, 12, 18, 24, 36
		15.	6, 9, 12, 18, 24, 36, 48
		16.	6, 9, 12, 18, 24, 36, 48, 54
Mode	**Value**		**Rate (unit: Mbps)**
802.11B/G		0.	Auto
	Fixed Rate	1.	1
		2.	2
		3.	5.5
		4.	11
		5.	6
		6.	9
		7.	12
		8.	18
		9.	24
		10.	36
		11.	48
		12.	54
	Rate Pool for Rate Switching	13.	1
		14.	1, 2
		15.	1, 2, 5.5
		16.	1, 2, 5.5, 11
		17.	1, 2, 5.5, 11, 6
		18.	1, 2, 5.5, 11, 6, 9
		19.	1, 2, 5.5, 11, 6, 9, 12
		20.	1, 2, 5.5, 11, 6, 9, 12, 18
		21.	1, 2, 5.5, 11, 6, 9, 12, 18, 24
		22.	1, 2, 5.5, 11, 6, 9, 12, 18, 24, 36
		23.	1, 2, 5.5, 11, 6, 9, 12, 18, 24, 36, 48
		24.	1, 2, 5.5, 11, 6, 9, 12, 18, 24, 36, 48, 54

4.1.8. Channel=value

value

802.11b/g: 1~14 depends on CountryRegion setting
802.11a : 36~165 depends on CountryRegion setting

4.1.9. BasicRate=value

Value

0 ~4095

Note:

A bitmap represent basic support rate(A mode not support)

1: Basic rate-1Mbps
2: Basic rate-2Mbps
3: Basic rate-1Mbps, 2Mbps
4: Basic rate-5.5Mbps
15: Basic rate-1Mbps, 2Mbps, 5.5Mbps, 11Mbps

Examples:

Basic Rate Bit Map (max. 12-bit, represent max. 12 basic rates)												
Bit	11	10	9	8	7	6	5	4	3	2	1	0
Rate	54	48	36	24	18	12	9	6	11	5.5	2	1
Set	0	1	0	1	0	1	0	1	1	1	1	1
Hex	5				5				F			
Decimal	1375											

Notes:

Set correct basic rates set before changing wireless mode.

11B/G Mixed:

iwpriv ra0 set BasicRate=15 ➔ (0x0F: 1, 2, 5.5, 11 Mbps)
iwpriv ra0 set WirelessMode=0

11B:

iwpriv ra0 set BasicRate=3 ➔ (0x03: 1, 2 Mbps)
iwpriv ra0 set WirelessMode=1

11G-Only:

iwpriv ra0 set BasicRate=351 ➔ (0x15F : 1, 2, 5.5, 11, 6, 12, 24 Mbps)
iwpriv ra0 set WirelessMode=2

4.1.10. BeaconPeriod=value

value

20 ~ 999

4.1.11. DtimPeriod=value

value

1 ~ 255

4.1.12. TxPower=value

Vaule

100 ~ 90 use value in E2PROM as default
90 ~ 60 default value -2
60 ~ 30 default value -6
30 ~ 15 default value -12
15 ~ 9 default value -18
9 ~ 0 default value -24

Note:

1. Range: 1 ~ 100 (unit in percentage)
2. This value restricted by HW characteristic.

4.1.13. BGProtection=value

value
- 0: Auto
- 1: Always On
- 2: Always Off

4.1.14. DisableOLBC=value

value
- 0: Enable
- 1: Disable

4.1.15. TxAntenna=value

value
- 1: Antenna-A
- 2: Antenna-B

4.1.16. RxAntenna=value

value
- 0:
- 1:
- 2:

Note: not support yet

4.1.17. TxPreamble=value

value
- 0: Long Preamble
- 1: Short Preamble

4.1.18. RTSThreshold=value

value

1 ~ 2347

4.1.19. FragThreshold=value

value

256 ~ 2346

4.1.20. TxBurst=value

value
- 0: Disable
- 1: Enable

4.1.21. PktAggregate=value

value
- 0: Disable
- 1: Enable

4.1.22. TurboRate=value

Value
- 0: Disable
- 1: Enable

Note: not support yet

4.1.23. NoForwarding=value

value
- 0: Disable

1: Enable

4.1.24. NoForwardingBTNBSSID=value

value
0: Disable
1: Enable

4.1.25. HideSSID=value

value
0: Disable
1: Enable

4.1.26. ShortSlot=value

value
0: Disable
1: Enable

4.1.27. AutoChannelSelect=value

Value (auto channel select when driver is loaded, only support in B/G band)
0: Disable
1: Enable

4.1.28. HSCounter=value

Value
0: Disable
1: Enable
Note: HotSpot counter, record last data packet time, tx byte count, rx byte count per client

4.1.29. AccessPolicy0=value

value
0: Disable
1: Allow all
2: Reject all

4.1.30. AccessControlList0=value

value
[Mac Address];[Mac Address];...
Example:
00:10:20:30:40:50;0A:0b:0c:0D:0e:0f;1a:2b:3c:4d:5e:6f
Note:
ACL for Bssid0, max=64

4.1.31. AccessPolicy1=value

value
0: Disable
1: Allow all
2: Reject all

4.1.32. AccessControlList1=value

value
[Mac Address];[Mac Address];...
Example:
00:10:20:30:40:50;0A:0b:0c:0D:0e:0f;1a:2b:3c:4d:5e:6f
Note:
ACL for Bssid1, max=64

4.1.33. AccessPolicy2=value

value

0: Disable
1: Allow all
2: Reject all

4.1.34. AccessControlList2=value

value

[Mac Address];[Mac Address];...

Example:

00:10:20:30:40:50;0A:0b:0c:0D:0e:0f;1a:2b:3c:4d:5e:6f

Note:

ACL for Bssid2, max=64

4.1.35. AccessPolicy3=value

value

0: Disable
1: Allow all
2: Reject all

4.1.36. AccessControlList3=value

value

[Mac Address];[Mac Address];...

Example:

00:10:20:30:40:50;0A:0b:0c:0D:0e:0f;1a:2b:3c:4d:5e:6f

Note:

ACL for Bssid3, max=64

4.2. iwpriv ra0 set [parameters]=[Val]

4.2.1. parameter :: DriverVersion

[Val] range:
{0}

4.2.2. parameter :: CountryRegion

[Val] range:
{0~6}
Explanation: Set country region
0: use 1 ~ 11 Channel
1: use 1 ~ 13 Channel
2: use 10, 11 Channel
3: use 10 ~ 13 Channel
4: use 14 Channel
5: use 1 ~ 14 Channel
6: use 3 ~ 9 Channel
7: use 5-13 Channel

4.2.3. parameter :: CountryRegionABand

[Val] range:
{0~7}
Explanation: Set country region for A band
value
0: channel 36, 40, 44, 48, 52, 56, 60, 64, 149, 153, 157, 161, 165
1: channel 36, 40, 44, 48, 52, 56, 60, 64, 100, 104, 108, 112, 116, 120, 124, 128, 132, 136,140
2: channel 36, 40, 44, 48, 52, 56, 60, 64
3: channel 52, 56, 60, 64, 149, 153, 157, 161
4: channel 149, 153, 157, 161, 165
5: channel 149, 153, 157, 161
6: channel 36, 40, 44, 48
7: channel 36, 40, 44, 48, 52, 56, 60, 64, 100, 104, 108, 112, 116, 120, 124, 128, 132, 136, 140, 149, 153, 157, 161, 165
8: channel 52, 56, 60, 64
9: channel 34, 38, 42, 46
10: channel 34, 36, 38, 40, 42, 44, 46, 48, 52, 56, 60, 64

4.2.4. parameter :: CountryCode

[Val] range:
2 characters, like TW for Taiwan.

Please refer to ISO3166 code list for other countries and can be found at
http://www.iso.org/iso/en/prods-services/iso3166ma/02iso-3166-code-lists/list-en1.html#sz

4.2.5. parameter :: HSCounter

[Val] range:
{0, 1}
Explanation: Set HotSpot counter Enable or Disable
0: Disable,
1: Enable

4.2.6. parameter :: AccessPolicy

[Val] range:
{0~2}
Explanation: Set Access control policy
0: Disble,
1: Allow All,
2: Reject All

4.2.7. parameter :: AccessControlList

[Val] range:
{"[MAC address];[MAC address];..."}
Explanation: Set Access control MAC table list
Up to 64 MAC address.

4.2.8. parameter :: Debug

[Val] range:
{0 ~ 5}
Explanation: Set Debug level

4.2.9. parameter :: ResetCounter

[Val] range:
{0}
Explanation: Reset all statistics counter

4.2.10. parameter :: RadioOn

[Val] range:
{0,1}
Explanation: Turn radio on or off
0: Off,
1: On

4.2.11. parameter :: SiteSurvey

[Val] range:
{1}
Explanation: Issue a site survey command to driver

4.2.12. parameter :: RetryLimit

[Val] range:
{0~1}
0: Short Retry
1: Long Retry
Explanation: Set Data Frames RetryLimit.

4.2.13. parameter :: TxQueueSize

[Val] range:
{1 ~ }
Explanation:Set TxQueue size.

4.2.14. parameter :: CountryString

[Val] range: (**Case insensitive**)
32 characters, like Taiwan.

Please refer to ISO3166 code list for other countries and can be found at
http://www.iso.org/iso/en/prods-services/iso3166ma/02iso-3166-code-lists/list-en1.html#sz

Item	Country Number	ISO Name	Country Name (CountryString)	Support 802.11A	802.11A Country Region	Support 802.11G	802.11G Country Region
1.	0	DB	Debug	Yes	A_BAND_REGION_7	Yes	G_BAND_REGION_5
2.	8	AL	ALBANIA	No	A_BAND_REGION_0	Yes	G_BAND_REGION_1
3.	12	DZ	ALGERIA	No	A_BAND_REGION_0	Yes	G_BAND_REGION_1
4.	32	AR	ARGENTINA	Yes	A_BAND_REGION_3	Yes	G_BAND_REGION_1
5.	51	AM	ARMENIA	Yes	A_BAND_REGION_2	Yes	G_BAND_REGION_1
6.	36	AU	AUSTRALIA	Yes	A_BAND_REGION_0	Yes	G_BAND_REGION_1
7.	40	AT	AUSTRIA	Yes	A_BAND_REGION_1	Yes	G_BAND_REGION_1
8.	31	AZ	AZERBAIJAN	Yes	A_BAND_REGION_2	Yes	G_BAND_REGION_1
9.	48	BH	BAHRAIN	Yes	A_BAND_REGION_0	Yes	G_BAND_REGION_1
10.	112	BY	BELARUS	No	A_BAND_REGION_0	Yes	G_BAND_REGION_1
11.	56	BE	BELGIUM	Yes	A_BAND_REGION_1	Yes	G_BAND_REGION_1
12.	84	BZ	BELIZE	Yes	A_BAND_REGION_4	Yes	G_BAND_REGION_1
13.	68	BO	BOLIVIA	Yes	A_BAND_REGION_4	Yes	G_BAND_REGION_1
14.	76	BR	BRAZIL	Yes	A_BAND_REGION_1	Yes	G_BAND_REGION_1
15.	96	BN	BRUNEI DARUSSALAM	Yes	A_BAND_REGION_4	Yes	G_BAND_REGION_1
16.	100	BG	BULGARIA	Yes	A_BAND_REGION_1	Yes	G_BAND_REGION_1
17.	124	CA	CANADA	Yes	A_BAND_REGION_0	Yes	G_BAND_REGION_0
18.	152	CL	CHILE	Yes	A_BAND_REGION_0	Yes	G_BAND_REGION_1
19.	156	CN	CHINA	Yes	A_BAND_REGION_4	Yes	G_BAND_REGION_1
20.	170	CO	COLOMBIA	Yes	A_BAND_REGION_0	Yes	G_BAND_REGION_0
21.	188	CR	COSTA RICA	No	A_BAND_REGION_0	Yes	G_BAND_REGION_1
22.	191	HR	CROATIA	Yes	A_BAND_REGION_2	Yes	G_BAND_REGION_1
23.	196	CY	CYPRUS	Yes	A_BAND_REGION_1	Yes	G_BAND_REGION_1
24.	203	CZ	CZECH REPUBLIC	Yes	A_BAND_REGION_2	Yes	G_BAND_REGION_1
25.	208	DK	DENMARK	Yes	A_BAND_REGION_1	Yes	G_BAND_REGION_1
26.	214	DO	DOMINICAN REPUBLIC	Yes	A_BAND_REGION_0	Yes	G_BAND_REGION_0
27.	218	EC	ECUADOR	No	A_BAND_REGION_0	Yes	G_BAND_REGION_1
28.	818	EG	EGYPT	Yes	A_BAND_REGION_2	Yes	G_BAND_REGION_1
29.	222	SV	EL SALVADOR	No	A_BAND_REGION_0	Yes	G_BAND_REGION_1
30.	233	EE	ESTONIA	Yes	A_BAND_REGION_1	Yes	G_BAND_REGION_1
31.	246	FI	FINLAND	Yes	A_BAND_REGION_1	Yes	G_BAND_REGION_1
32.	250	FR	FRANCE	Yes	A_BAND_REGION_2	Yes	G_BAND_REGION_1
33.	268	GE	GEORGIA	Yes	A_BAND_REGION_2	Yes	G_BAND_REGION_1
34.	276	DE	GERMANY	Yes	A_BAND_REGION_1	Yes	G_BAND_REGION_1
35.	300	GR	GREECE	Yes	A_BAND_REGION_1	Yes	G_BAND_REGION_1
36.	320	GT	GUATEMALA	Yes	A_BAND_REGION_0	Yes	G_BAND_REGION_0
37.	340	HN	HONDURAS	No	A_BAND_REGION_0	Yes	G_BAND_REGION_1
38.	344	HK	HONG KONG	Yes	A_BAND_REGION_0	Yes	G_BAND_REGION_1
39.	348	HU	HUNGARY	Yes	A_BAND_REGION_1	Yes	G_BAND_REGION_1
40.	352	IS	ICELAND	Yes	A_BAND_REGION_1	Yes	G_BAND_REGION_1
41.	356	IN	INDIA	Yes	A_BAND_REGION_0	Yes	G_BAND_REGION_1
42.	360	ID	INDONESIA	Yes	A_BAND_REGION_4	Yes	G_BAND_REGION_1
43.	364	IR	IRAN	Yes	A_BAND_REGION_4	Yes	G_BAND_REGION_1
44.	372	IE	IRELAND	Yes	A_BAND_REGION_1	Yes	G_BAND_REGION_1
45.	376	IL	ISRAEL	No	A_BAND_REGION_0	Yes	G_BAND_REGION_1
46.	380	IT	ITALY	Yes	A_BAND_REGION_1	Yes	G_BAND_REGION_1
47.	392	JP	JAPAN	Yes	A_BAND_REGION_9	Yes	G_BAND_REGION_1
48.	400	JO	JORDAN	Yes	A_BAND_REGION_0	Yes	G_BAND_REGION_1

49.	398	KZ	KAZAKHSTAN	No	A_BAND_REGION_0	Yes	G_BAND_REGION_1
50.	408	KP	KOREA DEMOCRATIC	Yes	A_BAND_REGION_5	Yes	G_BAND_REGION_1
51.	410	KR	KOREA REPUBLIC OF	Yes	A_BAND_REGION_5	Yes	G_BAND_REGION_1
52.	414	KW	KUWAIT	No	A_BAND_REGION_0	Yes	G_BAND_REGION_1
53.	428	LV	LATVIA	Yes	A_BAND_REGION_1	Yes	G_BAND_REGION_1
54.	422	LB	LEBANON	No	A_BAND_REGION_0	Yes	G_BAND_REGION_1
55.	438	LI	LIECHTENSTEIN	Yes	A_BAND_REGION_1	Yes	G_BAND_REGION_1
56.	440	LT	LITHUANIA	Yes	A_BAND_REGION_1	Yes	G_BAND_REGION_1
57.	442	LU	LUXEMBOURG	Yes	A_BAND_REGION_1	Yes	G_BAND_REGION_1
58.	446	MO	MACAU	Yes	A_BAND_REGION_0	Yes	G_BAND_REGION_1
59.	807	MK	MACEDONIA	No	A_BAND_REGION_0	Yes	G_BAND_REGION_1
60.	458	MY	MALAYSIA	Yes	A_BAND_REGION_0	Yes	G_BAND_REGION_1
61.	484	MX	MEXICO	Yes	A_BAND_REGION_0	Yes	G_BAND_REGION_0
62.	492	MC	MONACO	Yes	A_BAND_REGION_2	Yes	G_BAND_REGION_1
63.	504	MA	MOROCCO	No	A_BAND_REGION_0	Yes	G_BAND_REGION_1
64.	528	NL	NETHERLANDS	Yes	A_BAND_REGION_1	Yes	G_BAND_REGION_1
65.	554	NZ	NEW ZEALAND	Yes	A_BAND_REGION_0	Yes	G_BAND_REGION_1
66.	578	NO	NORWAY	Yes	A_BAND_REGION_0	Yes	G_BAND_REGION_0
67.	512	OM	OMAN	Yes	A_BAND_REGION_0	Yes	G_BAND_REGION_1
68.	586	PK	PAKISTAN	No	A_BAND_REGION_0	Yes	G_BAND_REGION_1
69.	591	PA	PANAMA	Yes	A_BAND_REGION_0	Yes	G_BAND_REGION_0
70.	604	PE	PERU	Yes	A_BAND_REGION_4	Yes	G_BAND_REGION_1
71.	608	PH	PHILIPPINES	Yes	A_BAND_REGION_4	Yes	G_BAND_REGION_1
72.	616	PL	POLAND	Yes	A_BAND_REGION_1	Yes	G_BAND_REGION_1
73.	620	PT	PORTUGAL	Yes	A_BAND_REGION_1	Yes	G_BAND_REGION_1
74.	630	PR	PUERTO RICO	Yes	A_BAND_REGION_0	Yes	G_BAND_REGION_0
75.	634	QA	QATAR	No	A_BAND_REGION_0	Yes	G_BAND_REGION_1
76.	642	RO	ROMANIA	No	A_BAND_REGION_0	Yes	G_BAND_REGION_1
77.	643	RU	RUSSIA FEDERATION	No	A_BAND_REGION_0	Yes	G_BAND_REGION_1
78.	682	SA	SAUDI ARABIA	No	A_BAND_REGION_0	Yes	G_BAND_REGION_1
79.	702	SG	SINGAPORE	Yes	A_BAND_REGION_0	Yes	G_BAND_REGION_1
80.	703	SK	SLOVAKIA	Yes	A_BAND_REGION_1	Yes	G_BAND_REGION_1
81.	705	SI	SLOVENIA	Yes	A_BAND_REGION_1	Yes	G_BAND_REGION_1
82.	710	ZA	SOUTH AFRICA	Yes	A_BAND_REGION_1	Yes	G_BAND_REGION_1
83.	724	ES	SPAIN	Yes	A_BAND_REGION_1	Yes	G_BAND_REGION_1
84.	752	SE	SWEDEN	Yes	A_BAND_REGION_1	Yes	G_BAND_REGION_1
85.	756	CH	SWITZERLAND	Yes	A_BAND_REGION_1	Yes	G_BAND_REGION_1
86.	760	SY	SYRIAN ARAB REPUBLIC	No	A_BAND_REGION_0	Yes	G_BAND_REGION_1
87.	158	TW	TAIWAN	Yes	A_BAND_REGION_3	Yes	G_BAND_REGION_0
88.	764	TH	THAILAND	No	A_BAND_REGION_0	Yes	G_BAND_REGION_1
89.	780	TT	TRINIDAD AND TOBAGO	Yes	A_BAND_REGION_2	Yes	G_BAND_REGION_1
90.	788	TN	TUNISIA	Yes	A_BAND_REGION_2	Yes	G_BAND_REGION_1
91.	792	TR	TURKEY	Yes	A_BAND_REGION_2	Yes	G_BAND_REGION_1
92.	804	UA	UKRAINE	No	A_BAND_REGION_0	Yes	G_BAND_REGION_1
93.	784	AE	UNITED ARAB EMIRATES	No	A_BAND_REGION_0	Yes	G_BAND_REGION_1
94.	826	GB	UNITED KINGDOM	Yes	A_BAND_REGION_1	Yes	G_BAND_REGION_1
95.	840	US	UNITED STATES	Yes	A_BAND_REGION_0	Yes	G_BAND_REGION_0
96.	858	UY	URUGUAY	Yes	A_BAND_REGION_5	Yes	G_BAND_REGION_1
97.	860	UZ	UZBEKISTAN	Yes	A_BAND_REGION_1	Yes	G_BAND_REGION_0
98.	862	VE	VENEZUELA	Yes	A_BAND_REGION_5	Yes	G_BAND_REGION_1
99.	704	VN	VIET NAM	No	A_BAND_REGION_0	Yes	G_BAND_REGION_1
100.	887	YE	YEMEN	No	A_BAND_REGION_0	Yes	G_BAND_REGION_1
101.	716	ZW	ZIMBABWE	No	A_BAND_REGION_0	Yes	G_BAND_REGION_1

4.2.15. parameter :: SSID

[Val] range:

{0~z, less than 32 characters}
Explanation: Set SoftAP SSID

4.2.16. parameter :: WirelessMode

[Val] range:
{0~2}
Explanation: Set Wireless Mode
0:11b/g mixed,
1:11b only,
2:11g only
3:11a only

4.2.17. parameter :: TxRate

[Val] range:
{0~12}
Explanation: Set TxRate
a mode
0:Auto,
1:6Mbps,
2:9Mbps,
3:12Mbps,
4:18Mbps,
5:24Mbps,
6:36Mbps,
7:48Mbps,
8:54Mbps
9:6Mbps,
10:9Mbps, // Auto tune TxRate between 6 and 9
11:12Mbps, // Auto tune TxRate between 6, 9 and 12
12:18Mbps, // Auto tune TxRate between 6, 9, 12 and 18
13:24Mbps, // Auto tune TxRate between 6, 9, 12, 18 and 24
14:36Mbps, // Auto tune TxRate between 6, 9, 12, 18, 24 and 36
15:48Mbps, // Auto tune TxRate between 6, 9, 12, 18, 24, 36 and 48
16:54Mbps // Auto tune TxRate between 6, 9, 12, 18, 24, 36, 48 and 54
b/g mode
0:Auto,
1:1Mbps,
2:2Mbps,
3:5.5Mbps,
4:11Mbps,
5:6Mbps,
6:9Mbps,
7:12Mbps,
8:18Mbps,
9:24Mbps,
10:36Mbps,
11:48Mbps,
12:54Mbps
13:1Mbps,
14:2Mbps, auto tune tx rate between 1M, 2M
15:5.5Mbps, auto tune tx rate between 1M, 2M, 5.5M
16:11Mbps,
17:6Mbps,
18:9Mbps,
19:12Mbps,
20:18Mbps,
21:24Mbps,

22:36Mbps,
23:48Mbps,
24:54Mbps

TxRate			
Mode	**Value**		**Rate (unit: Mbps)**
802.11A		0.	Auto
	Fixed Rate	1.	6
		2.	9
		3.	12
		4.	18
		5.	24
		6.	36
		7.	48
		8.	54
	Rate Pool for Rate Switching	9.	6
		10.	6, 9
		11.	6, 9, 12
		12.	6, 9, 12, 18
		13.	6, 9, 12, 18, 24
		14.	6, 9, 12, 18, 24, 36
		15.	6, 9, 12, 18, 24, 36, 48
		16.	6, 9, 12, 18, 24, 36, 48, 54
Mode	**Value**		**Rate (unit: Mbps)**
802.11B/G		0.	Auto
	Fixed Rate	1.	1
		2.	2
		3.	5.5
		4.	11
		5.	6
		6.	9
		7.	12
		8.	18
		9.	24
		10.	36
		11.	48
		12.	54
	Rate Pool for Rate Switching	13.	1
		14.	1, 2
		15.	1, 2, 5.5
		16.	1, 2, 5.5, 11
		17.	1, 2, 5.5, 11, 6
		18.	1, 2, 5.5, 11, 6, 9
		19.	1, 2, 5.5, 11, 6, 9, 12
		20.	1, 2, 5.5, 11, 6, 9, 12, 18
		21.	1, 2, 5.5, 11, 6, 9, 12, 18, 24
		22.	1, 2, 5.5, 11, 6, 9, 12, 18, 24, 36
		23.	1, 2, 5.5, 11, 6, 9, 12, 18, 24, 36, 48
		24.	1, 2, 5.5, 11, 6, 9, 12, 18, 24, 36, 48, 54

4.2.18. parameter :: BasicRate

[Val] range:
{0~4095}
Explanation: Be careful to set this value, if you don't know what this is,

please don't set this field

Examples:

Basic Rate Bit Map (max. 12-bit, represent max. 12 basic rates)												
Bit	11	10	9	8	7	6	5	4	3	2	1	0
Rate	54	48	36	24	18	12	9	6	11	5.5	2	1
Set	0	1	0	1	0	1	0	1	1	1	1	1
Hex	5				5				F			
Decimal	1375											

4.2.19. parameter :: Channel

[Val] range:
802.11b/g: {1~14} depends on CountryRegion setting
802.11a : {36~165} depends on CountryRegion setting
Explanation: Set Channel

4.2.20. parameter :: BeaconPeriod

[Val] range:
{50~1024} ms
Explanation: Set beacon period

4.2.21. parameter :: DtimPeriod

[Val] range:
{1~255}
Explanation: Set Dtim period

4.2.22. parameter :: TxPower

[Val] range:
{1~100}
Explanation: Set AP Tx power percentage

4.2.23. parameter :: BGProtection

[Val] range:
{0~2}
Explanation: Set 11B/11G Protection
0:Auto,
1:Always on,
2:Always off

4.2.24. parameter :: DisableOLBC

[Val] range:
{0~1}
Explanation: Set OLBC detection
0:Enable
1:Disable

4.2.25. parameter :: TxAntenna

[Val] range: **not support yet**
{0~2}
Explanation:

4.2.26. parameter :: RxAntenna

[Val] range: **not support yet**
{0~2}
Explanation:

4.2.27. parameter :: TxPreamble

[Val] range:
{0~2}
Explanation: Set TxPreamble
0: Long Preamble
1: Short Preamble
2: Auto

4.2.28. parameter :: RTSThreshold

[Val] range:
{1~2347}
Explanation: Set RTS Threshold

4.2.29. parameter :: FragThreshold

[Val] range:
{256~2346}
Explanation: Set Fragment Threshold

4.2.30. parameter :: TxBurst

[Val] range:
{0, 1}
Explanation: Set TxBurst Enable or Disable
0: Disable,
1: Enable

4.2.31. parameter :: PktAggregate

[Val] range:
{0, 1}
Explanation: Set Ralink proprietary packet aggregate Enable or Disable
0: Disable,
1: Enable

4.2.32. parameter :: TurboRate

[Val] range:
{0, 1}
Explanation: Set TurboRate Enable or Disable (**Not support yet**)
0: Disable,
1: Enable

4.2.33. parameter :: NoForwarding

[Val] range:
{0, 1}
Explanation: Set No Forwarding Enable or Disable
0: Disable,
1: Enable

4.2.34. parameter :: NoForwardingBTNBSSID

[Val] range:
{0, 1}
Explanation: Set No Forwarding between each BSSID interface
0: Disable,
1: Enable

4.2.35. parameter :: HideSSID

[Val] range:

{0, 1}
Explanation: Set Hide SSID Enable or Disable
0: Disable,
1: Enable

4.2.36. parameter :: ShortSlot

[Val] range:
{0, 1}
Explanation: Set Short Slot Time Enable or Disable
0: Disable,
1: Enable

4.3. USAGE - iwpriv ra0 get_site_survey

iwpriv ra0 get_site_survey
where
[Parameters] [Val] range explanation

4.3.1. parameter :: get_site_survey

[Val] range:
{}
Explanation: get data from kernel and write to standard output

4.4. USAGE - iwpriv ra0 get_mac_table

iwpriv ra0 get_mac_table
where
[Parameters] [Val] range explanation

4.4.1. parameter :: get_mac_table

[Val] range:
{}
Explanation: get associated STA's MAC address

4.5. USAGE - iwpriv ra0 chk_asic_ver

iwpriv ra0 chk_asic_ver
where
[Parameters] [Val] range explanation

4.5.1. parameter :: chk_asic_ver

[Val] range:
{}
Explanation: Check ASIC hard code Version and EEPROM match or not.

4.6. USAGE - iwpriv ra0 stat

iwpriv ra0 stat
where
[Parameters] [Val] range explanation

4.6.1. parameter :: stat

[Val] range:
{}
Explanation: Read statistics counter.

4.7. Examples

4.7.1. Example I

```
iwpriv ra0 set CountryRegion=6
iwpriv ra0 set SSID=SoftAp-1
iwpriv ra0 set WirelessMode=0
iwpriv ra0 set TxRate=0
iwpriv ra0 set Channel=1
iwpriv ra0 set BeaconPeriod=100
iwpriv ra0 set BGProtection=1
iwpriv ra0 set TxPreamble=0
iwpriv ra0 set RTSThreshold=2347
iwpriv ra0 set FragThreshold=2346
iwpriv ra0 set TxBurst=0
iwpriv ra0 set TurboRate=0
iwpriv ra0 set NoForwarding=0
iwpriv ra0 set NoForwardingBTNBSSID=0
iwpriv ra0 set HideSSID=0
iwpriv ra0 set ShortSlot=0
iwpriv ra0 set AuthMode=SHARED
iwpriv ra0 set EncrypType=WEP
iwpriv ra0 set DefaultKeyID=1
iwpriv ra0 set Key1=1234567890
iwpriv ra0 set Key2=passd
iwpriv ra0 set Key3=1234567890123456789012345 6
iwpriv ra0 set key4=enterpassword
iwpriv ra0 set AccessPolicy=1
iwpriv ra0 set AccessControlList=   "00:03:A0:10:0E:10;
                                    00:08:0c:FD:e1:00;
                                    1a:28:40:42:ce:6f"
iwpriv ra0 set WPAPSK=0123456789
iwpriv ra0 set Debug=0
iwpriv ra0 set ResetCounter=1
```

4.7.2. Example II

One iwpriv command sets two parameters.

```
iwpriv ra0 set Channel=8
iwpriv ra0 set SSID=SoftAp-1
```

5. WPS – Wi-Fi Protected Setup

5.1. Simple Config Architectural Overview

This section presents a high-level description of the Simple Config architecture. Much of the material is taken directly from the Simple Config specification.

Figure 1 depicts the major components and their interfaces as defined by Wi-Fi Simple Config Spec. There are three logical components involved: the Registrar, the access point (AP), and the Enrollee.

- The **Enrollee** is a device seeking to join a WLAN domain. Once an Enrollee obtains a valid credential, it becomes a member.
- A **Registrar** is an entity with the authority to issue and revoke domain credentials. A registrar can be integrated into an AP.
- The **AP** can be either a WLAN AP or a wireless router.

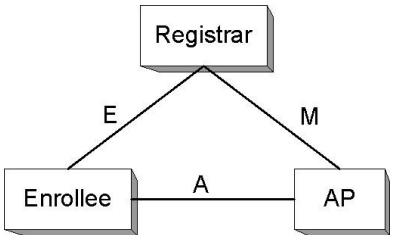


Figure 1. Components and Interfaces

Registration initiation is ordinarily accomplished by a user action such as powering up the Enrollee and, optionally, running a setup wizard on the Registrar (PC).

5.1.1. Interface E

This interface is logically located between the Enrollee and the Registrar (physically, the AP can work as a proxy to convey the messages). The functionality of Interface E is to enable the Registrar to discover and issue WLAN Credentials to the Enrollee. Interface E may include only WLAN communication or it may also include communication across an out-of-band channel.

5.1.1.1. Enrollee

The Enrollee implements Interface E by:

1. Including a Simple Config IE in 802.11 probe messages.
2. Including a device password on a display or printed label for in-band configuration.
3. Optionally supporting one or more out-of-band configuration channels.
4. Implementing the "Device" part of the Registration Protocol.
5. Optionally receiving ad-hoc probe-responses from wireless Registrars.

5.1.1.2. Registrar

The Registrar implements Interface E by:

1. Processing Enrollee (device or AP) Discovery data in Probe messages (for wireless Registrars) and/or UPnP (for Ethernet-based Registrars).
2. Implementing the "Registrar" part of the Registration Protocol.
3. Optionally supporting one or more out-of-band configuration channels.
4. Configuring the AP with the Enrollee's MAC address and Credential using Interface M.
5. Optionally respond to Enrollee Probe-Requests via an ad-hoc Probe-Response.

5.1.2. Interface M

This interface is between the AP and the Registrar. Interface M enables an external Registrar to manage a Wi-Fi Simple Config AP. Wi-Fi Simple Config uses a similar protocol for setting up the AP Management interface as for issuing credentials to Enrollee devices.

5.1.2.1. AP

The AP implements Interface M by:

1. Acting as the Enrollee in the Registration Protocol for initial setup with one or more external Registrars. This includes sending its own Discovery message across all appropriate channels (Ethernet and/or 802.11 probe response over Wi-Fi). Support for at least three external Registrars is required.
2. Implementing the Management Interface described in the **WFADevice** and **WFAWLANConfig** Service documents. This requires the AP to be a UPnP device that includes support for the Wi-Fi Simple Config proxy service.
3. Monitoring 802.11 probe request and EAP messages from Enrollees and converting them to UPnP Event messages according to the method described in the WFAWLANConfig Service document.

5.1.2.2. Registrar

The Registrar implements Interface M by:

1. Processing AP Discovery messages across 802.11 and/or Ethernet.
2. Receiving and processing Enrollee Discovery and Registration messages forwarded by the AP.
3. Optionally receiving and processing Enrollee Discovery and Registration messages sent in ad hoc mode.
4. Implementing the Registrar side of the Registration Protocol to gain management rights over the AP or to issue WLAN credentials to Enrollees
5. Configuring the AP with the MAC address and/or per-device Credential of the Enrollee.
6. Implementing the Management Interface described in the WFADevice and WFAWLANConfig Service documents. This requires the Registrar to function as a UPnP control point.

5.1.3. Interface A

This interface is between the Enrollee and the AP. The function of Interface A is to enable discovery of the Simple Config WLAN and to enable communication between the Enrollee and Ethernet-only Registrars.

5.1.3.1. AP

The AP implements Interface A by:

1. Sending out 802.11 beacons indicating support for Simple Config and generating Probe Response messages containing a description of the AP.
2. Implementing an 802.1X authenticator and the Simple Config EAP method.
3. Proxying 802.11 probe request and EAP messages between Enrollees and external Registrars as described in the WFADevice and WFAWLANConfig Service documents.

5.1.3.2. Enrollee

The Enrollee implements Interface A by:

1. Discovering a Simple Config AP and/or wireless external Registrar and sending it 802.11 probe requests including the Enrollee Discovery data.
2. Implementing an 802.1X supplicant and the Simple Config Registration Protocol EAP method.

5.2. Supported Parameters in RT61AP.dat

5.2.1. WscConfMode=value

value

0x0: Disable
0x1: Enrollee
0x2: Proxy
0x4: Registrar

5.2.2. WscConfStatus=value

value

1: AP is un-configured
2: AP is configured

5.3. iwpriv ra0 set [parameters]=[value]

Syntax:		Example	
Section#	parameters	5.3.1	wscConfMode
	Explanation		Set WPS function
	Value:		Value:
	0: …		0x0: Disable
	1: …		0x1: Enrollee
	.: …		0x2: Proxy

5.3.1. WscConfMode

Set WPS function, **bitwise**.

value:

0x0: Disable
0x1: Enrollee
0x2: Proxy
0x4: Registrar

5.3.2. WscConfStatus

Set WPS AP SC (Simple Config) State.

value:

1: AP is un-configured
2: AP is configured

5.3.3. WscMode

Set WPS Configured Methods.

value:

1: use PIN code (Personal Identification Number)
2: use PBC (Push Button Communication)

5.3.4. WscStatus

Get WPS Configured Methods.

value:

0	Not Used
1	Idle
2	WSC Process Fail
3	Start WSC Process
4	Received EAPOL-Start
5	Sending EAP-Req(ID)
6	Receive EAP-Rsp(ID)
7	Receive EAP-Req with wrong WSC SMI Vendor Id
8	Receive EAPReq with wrong WSC Vendor Type
9	Sending EAP-Req(WSC_START)
10	Send M1
11	Received M1
12	Send M2
13	Received M2
14	Received M2D
15	Send M3
16	Received M3
17	Send M4
18	Received M4
19	Send M5
20	Received M5
21	Send M6
22	Received M6

23 Send M7
24 Received M7
25 Send M8
26 Received M8
27 Processing EAP Response (ACK)
28 Processing EAP Request (Done)
29 Processing EAP Response (Done)
30 Sending EAP-Fail
31 WSC_ERROR_HASH_FAIL
32 WSC_ERROR_HMAC_FAIL
33 WSC_ERROR_DEV_PWD_AUTH_FAIL
34 Configured

5.3.5. WscPinCode

Input Enrollee's Pin Code to AP-Registrar.

value:

8-digits

5.3.6. WscOOB

Reset WPS AP to the OOB (out-of-box) configuration.

value:

0: Disable
1: Enable

5.3.7. WscGetConf

Trigger WPS AP to do simple config with WPS Client.

value:

0: Disable
1: Enable

5.4. Examples

Easy Setting

1 Init steps:

1) iwpriv ra0 set WscConfMode=7 # bit0=Enrolle, bit1=Proxy, bit2=Registrar
2) iwpriv ra0 set WscConfStatus=1 # 1=unconfigured, 2=configured

2 Pin Code Steps:

1) iwpriv ra0 set WscMode=1 # 1=Pin code, 2=PBC.
2) iwpriv ra0 set WscPinCode=24543057 # From STA UI get the Pin code.
3) iwpriv ra0 set WscGetConf=1 # Trigger WPS AP to do simple config with WPS Client.

3 PBC Steps:

1) iwpriv ra0 set WscMode=2 # 1=Pin code, 2=PBC
2) iwpriv ra0 set WscGetConf=1 # Trigger WPS AP to do simple config with WPS Client

5.4.1. Disable WPS function support

⇨ iwpriv ra0 set WscConfMode=0

5.4.2. Enable WPS function support

⇨ iwpriv ra0 set WscConfMode =7 (Binary: 111)
(AP could be Registrar(0x4), Proxy(0x2) or Enrollee(0x1))

5.4.3. WPS AP SC (Simple Config) State

⇨ iwpriv ra0 set WscConfStatus=1 (AP is un-configured)
⇨ iwpriv ra0 set WscConfStatus=2 (AP is configured)

5.4.4. WPS Configured Methods

⇨ iwpriv ra0 set WscMode =1 (use PIN code)
⇨ iwpriv ra0 set WscMode =2 (use PBC)

5.4.5. Input Enrollee's Pin Code to AP-Registrar

⇨ iwpriv ra0 set WscPinCode=xxxxxxxx

5.4.6. Reset WPS AP to the OOB configuration

⇨ iwpriv ra0 set WscOOB=1
(Security: WPAPSK/TKIP, psk: "RalinkInitialAPxx1234" ; SC state: 0x1)
(SSID: RalinkInitialAPxxxxxx, last three characters of AP MAC address)

5.4.7. Trigger WPS AP to do simple config with WPS Client

⇨ iwpriv ra0 set WscGetConf=1

5.4.8. AP services as Enrollee by using PIN code

⇨ iwpriv ra0 set WscMode=1
⇨ iwpriv ra0 set WscGetConf=1

5.4.9. AP services as Enrollee by using PBC

⇨ iwpriv ra0 set WscMode=2
⇨ iwpriv ra0 set WscGetConf=1

5.4.10. AP services as Internal Registrar using PIN code

⇨ iwpriv ra0 set WscMode=1
⇨ iwpriv ra0 set WscPinCode=xxxxxxxx (PIN code from Enrollee, len=8)
⇨ iwpriv ra0 set WscGetConf=1

5.4.11. AP services as Internal Registrar using PBC

- ⇨ iwpriv ra0 set WscMode=2
- ⇨ iwpriv ra0 set WscGetConf=1

5.5. Notes

1. AP services as Enrollee:
   - ⇨ If AP-Enrollee SC state is 0x1, AP will restart with new configurations.
   - ⇨ If AP-Enrollee SC state is 0x2, AP sends own configurations to external-registrar and ignores configurations from external-registrar.
2. AP services as Registrar:
   - ⇨ If AP-Registrar SC state is 0x1, the security mode will be WPAPSK/TKIP and generate random 64bytes psk; after process, AP will restart with new security.
3. WPS AP only services one WPS client at a time.
4. WPS AP only can work in ra0.

5.6. New files for WPS AP

1. wsc.c
2. wsc_tlv.c
3. sha2.c
4. hmac.c
5. dh_key.c
6. evp_enc.c

5.7. New compile flag for WPS AP

WFLAGS += -DWSC_SUPPORT

5.8. New items for RT61AP.dat file

1. WscConfMode=0
2. WscConfStatus=1

5.9. Related Documents

1. Wi-Fi Protected Setup Specification v1.0 (member only)
2. Wi-Fi Protected Setup White Paper
3. Introducing Wi-Fi Protected Setup
4. WSC Linux* Reference Implementation
5. How to Use Windows Connect Now Configuration to Enable Simple Setup for Consumer Wi-Fi Networks [WinHEC 2006; 5.83 MB]
6. Network Infrastructure Device Implementer's Guide

5.10. UPNP Daemon HOWTO

5.10.1. Build WPS UPnP daemon

5.10.1.1. Requirements:

1. Linux platform
2. Ralink wireless driver version which support WPS
3. Libupnp
   - ⇨ You can download the libupnp source code from the following URL: http://upnp.sourceforge.net/
   - ⇨ libupnp-1.3.1 is preferred version. For other versions, you may need to patch our modification to the library yourself.
4. POSIX thread library
   - ⇨ Both libupnp and our WPS UPnP daemon need the POSIX thread library, following are recommended pthread library version.
     - For uCLibc, need the version >= 0.9.27
     - For GLIBC, need the version >= 2.3.2
   - ⇨ If your pthread library is older than upper list, you may need to upgrade it.

5.10.1.2. Build and Run:

1. Modify the "$(work_directory)/wsc_upnp/Makefile" and change the compile flags depends on your target platform.
   - ⇨ Ex. For arm-linux target platform, you may need to set the following fags:
     - CROSS_COMPILE = arm-linux-
     - TARGET_HOST = arm-linux
2. Modify the "$(work_directory)/wsc_upnp/libupnp-1.3.1/Makefile.src" and change the configure parameters.
   - ⇨ Ex. For big-endian system, you may need to add CFAGS as following:
     - ./configure --host=$(TARGET_HOST) CFLAGS="-mbig-endian"
3. Compile it
   - ⇨ Run "make" in "$(work_directory)/wsc_upnp", after successful compilation, you will get an execution file named "wscd".
4. Install
   - ⇨ Create a sub-directory named "xml" in the "/etc" of your target platform
   - ⇨ Copy all files inside in "$(work_directory)/wsc_upnp/xml" to "/etc/xml"
     - Copy the "wscd" to the target platform.
5. Run it
   - ⇨ Before run it, be sure the target platform already has set the default route. Or the WPS daemon will failed when do initialization.
   - ⇨ Now you can run it by following command:
     - /bin/wscd –m 1 –d 3

5.10.2. Related Documents

1. WPS Specification (Simple_Config_v1.0g.pdf)
2. UPnP Device Architecture 1.0
3. Windows Connect Now-NET Version 1.0
4. WFAWLANConfig:1 Service Template Version 1.01
5. WFA Device:1 Device Template Version 1.01

6. WMM Parameters

6.1. Setting Parameters

1. Set 'WmmCapable' as 1 to turn on WMM QoS support
2. Parameters of 'APAifsn', 'APCwmin', 'APCwmax', 'APTxop', 'APACM' are WMM parameter for AP
3. Parameters of 'BSSAifsn', 'BSSCwmin', 'BSSCwmax', 'BSSTxop', 'BSSACM' are WMM parameter for station
4. Parameter of AckPolicy is for Ack policy which support normal Ack or no Ack
5. Default WMM parameters for STA and AP

Table 4 Default WMM Parameters for the STA

AC	CW_{min}	CW_{max}	AIFSN	TXOP Limit (802.11b)	TXOP Limit (802.11a/g)
AC_BK	15	1023	7	0	0
AC_BE	15	1023	3	0	0
AC_VI	7	15	2	188 6.016ms	94 3.008ms
AC_VO	3	7	2	102 3.264ms	47 1.504ms

Table 5 Default WMM Parameters for the AP

AC	CW_{min}	CW_{max}	AIFSN	TXOP Limit (802.11b)	TXOP Limit (802.11a/g)
AC_BK	15	1023	7	0	0
AC_BE	15	63	3	0	0
AC_VI	7	15	1	188 6.016ms	94 3.008ms
AC_VO	3	7	1	102 3.264ms	47 1.504ms

6. All WMM parameters do not support iwpriv command but 'WmmCapable'', please store all parameter to RT61AP.dat, and restart driver

7. The format for WMM parameter is as followed,
   APAifsn=3;7;1;1 //AC_BE, AC_BK, AC_VI, AC_VO

6.2. How to turn on WMM test in RT61 SoftAP

1. **WmmCapable=1**
   For each BSSID:
   0: Disable WMM,
   1: Enable WMM
   (If the parameter sets to 1, the relative BSSID will turn on WMM)

2. **TxBurst=0**

3. **Parameters for AP (for each AC (access category))**

```
APAifsn=3;7;1;1          // AC_BE;AC_BK;AC_VI;AC_VO
APCwmin=4;4;3;2          // AC_BE;AC_BK;AC_VI;AC_VO
APCwmax=6;10;4;3         // AC_BE;AC_BK;AC_VI;AC_VO
APTxop=0;0;94;47         // AC_BE;AC_BK;AC_VI;AC_VO
APACM=0;0;0;0            // AC_BE;AC_BK;AC_VI;AC_VO
```

4. **Parameters for all STAs (for each AC (access category))**

```
BSSAifsn=3;7;2;2         // AC_BE;AC_BK;AC_VI;AC_VO
BSSCwmin=4;4;3;2         // AC_BE;AC_BK;AC_VI;AC_VO
BSSCwmax=10;10;4;3       // AC_BE;AC_BK;AC_VI;AC_VO
BSSTxop=0;0;94;47        // AC_BE;AC_BK;AC_VI;AC_VO
BSSACM=0;0;0;0           // AC_BE;AC_BK;AC_VI;AC_VO
```

5. **Ack policy**

```
AckPolicy=0;0;0;0        // AC_BE;AC_BK;AC_VI;AC_VO; 0: Normal ACK, 1: No ACK
```

All default values comply with Wi-Fi spec.

6.3. The ACKs

1. **Current driver of RT61AP only support NORMAL_ACK and NO_ACK.**
   Section 11.1, item 4
   4. Parameter of AckPolicy is for Ack policy which support **normal Ack or no Ack**.
   The other two ack types have to be supported by hardware .

2. **The difference of ACKs**
   - **a.** NORMAL_ACK is used to ACK data packet.
   - **b.** NO_ACK is used never ACK any data packet.
   - **c.** NO_EXPLICIT_ACK have two ways to implement,
     - By received packet count threshold to ACK.
     - By timeing period threshold to ACK.
   - **d.** BLOCK_ACK is used to ACK data packet per ACK request packet received.
     - If peer didn't request to ACK then never ACK.
     - This type of ACK is depends on what AIR quality is.
       1.) AIR quality is bad, then the ACK should be mostly required.
       2.) AIR quality is good, then the ACK period maybe longer or even needn't ACK.

3. **Reference:**

Below table is pasted from IEEE802.11e-D13.0 for your reference.(Page 27 and 28)

Table 3.2—Ack policy field in QoS control field of QoS data frames

Bits in QoS Control field		Meaning
Bit 5	**Bit 6**	
0	0	**Normal acknowledgement.** The addressed recipient returns an ACK or QoS +CF-Ack frame after a SIFS period, according to the procedures defined in 9.2.8, 9.3.3 and 9.9.2.3. The Ack Policy field is set to this value in all directed frames in which the sender requires acknowledgement. For QoS Null (no data) frames, this is the only permissi-ble value for the Ack Policy field.
1	0	**No Acknowledgement.** The addressed recipient takes no action upon receipt of the frame. More details are provided in 9.11. The Ack Policy is set to this value in all directed frames in which the sender does not require acknowledgement. This combination is also used for broadcast and multicast frames that use the QoS frame format.
0	1	**No Explicit Acknowledgement.** There may be a response frame to the frame that is received, but it is neither the ACK nor any Data frame of subtype +CF-Ack. For Data frames of subtype QoS CF-Poll and subtype QoS CF-Ack+CF-Poll, this is the only permissible value for the Ack Policy field.
1	1	**Block Acknowledgement.** The addressed recipient takes no action upon the receipt of the frame except for recording the state. The recipient can expect a BlockAckReq frame in the future to which it responds using the procedure described in 9.10.

6.4. Access Precedence and Outgoing Frame Classification

1. 802.1e-D13

1.1. Section 7.3.2.16 Traffic Classification (TCLAS) Element

Table 20.7—Frame classifier type	
Classifier Type	**Classifier Parameters**
0	Ethernet parameters
1	TCP/UDP IP parameters
2	IEEE 802.1D/Q Parameters
3-255	Reserved

1.2. Section 9.1.3.1 HCF contention-based channel access (EDCA)

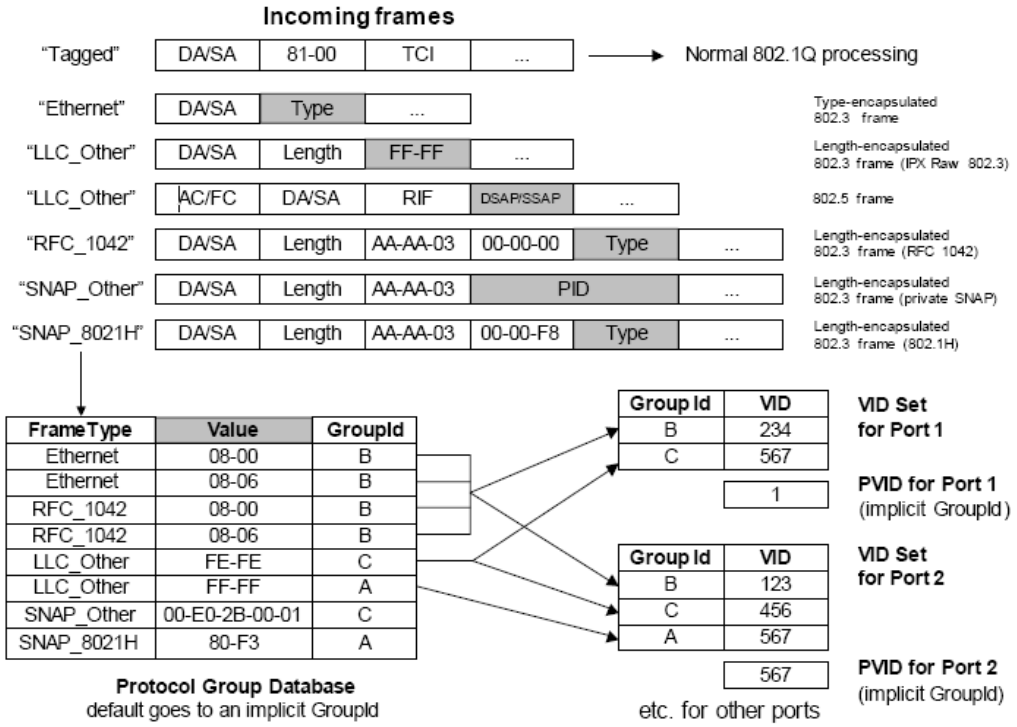
Table 20.23—User priority to Access Category mappings				
Priority	**User priority (UP - Same as 802.1D User Priority)**	**802.1D Designation**	**Access Category (AC)**	**Designation (Informative)**
lowest	1	BK	AC_BK	Background
	2	-	AC_BK	Background
	0	BE	AC_BE	Best Effort
	3	EE	AC_BE	Best Effort
	4	CL	AC_VI	Video
	5	VI	AC_VI	Video
	6	VO	AC_VO	Voice
highest	7	NC	AC_VO	Voice

2. 802.1Q-2003

2.1. Section 8.9 VLAN classification

3. 802.1q-rev-d4.0-2005-05-19

3.1. Section 6.8 Protocol VLAN classification



NOTE—The PID shown in this figure is a Protocol Identifier, as defined in 5.3 of IEEE Std 802. It is a 5-octet value consisting of a 3-octet OUI value followed by a 2-octet locally administered identifier.

Figure 6-2—Example of operation of port-and-protocol based classification

3.2. Section 9. Tagged frame format

Table 9-1—802.1Q Ethernet Type allocations		
Tag Type	Name	Value
VLAN TAG	802.1Q Tag Protocol Type (802.1QTagType)	81-00

4. RFC 2474

Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers (802.11e - Differentiated Services Code Point (DSCP))

5. RFC 791

Internet Protocol

6. RFC 795

6.1. Service mappings – TOS of IP Header

The IP Type of Service has the following fields:

Bit 0-2	Precedence.
Bit 3	0 = Normal Delay, 1 = Low Delay.
Bit 4	0 = Normal Throughput, 1 = High Throughput.
Bit 5	0 = Normal Relibility, 1 = High Relibility.
Bit 6-7	Reserved for Future Use.

0	1	2	3	4	5	6	7
PRECEDENCE			D	T	R	00	
111 - Network Control 110 - Internetwork Control 101 - CRITIC/ECP 100 - Flash Override 011 - Flash 010 - Immediate 001 – Priority 000 - Routine							

6.5. Supported Parameters in RT61AP.dat

6.5.1. WmmCapable=value

value
- 0: Disable
- 1: Enable

6.5.2. APAifsn=value

value

APAifsn=3;7;1;1 // AC_BE, AC_BK, AC_VI, AC_VO

6.5.3. APCwmin=value

value

APCwmin=4;4;3;2 // AC_BE, AC_BK, AC_VI, AC_VO

6.5.4. APCwmax =value

value

APCwmax=6;10;4;3 // AC_BE, AC_BK, AC_VI, AC_VO

6.5.5. APTxop =value

value

APTxop=0;0;94;47 // AC_BE, AC_BK, AC_VI, AC_VO

6.5.6. APACM =value

value

APACM=0;0;0;0 // AC_BE, AC_BK, AC_VI, AC_VO

6.5.7. BSSAifsn =value

value

BSSAifsn=3;7;2;2 // AC_BE, AC_BK, AC_VI, AC_VO

6.5.8. BSSCwmin =value

value

BSSCwmin=4;4;3;2 // AC_BE, AC_BK, AC_VI, AC_VO

6.5.9. BSSCwmax =value

value

BSSCwmax=10;10;4;3 // AC_BE, AC_BK, AC_VI, AC_VO

6.5.10. BSSTxop =value

value

BSSTxop=0;0;94;47 // AC_BE, AC_BK, AC_VI, AC_VO

6.5.11. BSSACM =value

value

BSSACM=0;0;0;0 // AC_BE, AC_BK, AC_VI, AC_VO

6.5.12. AckPolicy =value

value

AckPolicy=0;0;0;0 // AC_BE, AC_BK, AC_VI, AC_VO

6.5.13. APSDCapable=value

value **[Valid on WmmCapable=1]**
- 0: Disable
- 1: Enable

6.5.14. DLSCapable=value

value **[Valid on WmmCapable=1]**

0: Disable
1: Enable

6.5.15. EthWithVLANTag=value [RTL865x Only]

value

0: Disable
1: Enable

6.6. iwpriv ra0 set [parameters]=[Val]

6.6.1. parameter :: WmmCapable

[Val] range:
{0, 1}
Explanation: Set WmmCapable Enable or Disable
0: Disable,
1: Enable

7. IEEE802.11h+d

DFS - Dynamic Frequency Selection

7.1. IEEE802.11d

Regulatory Domains

1. To turn on IEEE802.11d, just fill up the parameter of 'CountryCode', according to ISO3166 code list. This parameter can work in A/B/G band.

2. The parameter of "CountryCode' needs to match with 'CountryRegion' or 'CountryRegionABand' depends on A or B/G band

3. Wi-Fi test requirement for IEEE802.11d
   - Country code IE(0x07) includes in beacon frame and probe response
   - Power constraint IE(32) includes in beacon frame and probe response

7.2. IEEE802.11h

Spectrum and Transmit Power Management

1. To turn on IEEE802.11h, just fill up the parameters of 'IEEE80211H', 'AutoChannelSelect' as 1, WirelessMode set as 3 to support A band. This parameter can work in only A band.

2. Use 'CSPeriod' to determine how many beacons before channel switch

3. Driver will turn off BBP tuning temporarily in radar detection mode

4. If turn on IEEE802.11h, AP will have 60sec to do channel available check, and will not send beacon and can not be connect.

5. Wi-Fi test requirement for IEEE802.11h
   - Force AP switch channel, AP will stop beacon transmit between 15 sec
   - At least five beacon includes channel switch announcement IE (37)in beacon frame

6. ETSI test requirement, please refer to ETSI EN 301 893 for V1.2.3 detail

Table D.1: DFS requirement values

Parameter	Value
Channel Availability Check Time	60 s
Channel Move Time	10 s
Channel Closing Transmission Time	260 ms

Table D.2: Interference Threshold values, Master

Maximum Transmit Power	Value (see note)
≥ 200 mW	-64 dBm
< 200 mW	-62 dBm
NOTE: This is the level at the input of the receiver assuming a 0 dBi receive antenna.	

Table D.3: Interference Threshold values, Slave

Maximum Transmit Power	Value (see note)
≥ 200 mW	-64 dBm
< 200 mW	N/A
NOTE: This is the level at the input of the receiver assuming a 0 dBi receive antenna.	

7.3. Supported Parameters in RT61AP.dat

7.3.1. IEEE80211H=value

[Val] range:
{0, 1}
Explanation: Spectrum management. This field can be enable only in A band

7.3.2. CSPeriod=value

value
0 ~ 255
Note :
Channel switch period (Beacon count), unit is based on Beacon interval.

7.3.3. MaxTxPowerLevel=value

value
0 ~ 255
Note :
Co-used with CountryCode to set MaxTxPowerLevel.

7.4. iwpriv ra0 set [parameters]=[Val]

7.4.1. parameter :: IEEE80211H

[Val] range:
{0, 1}
Explanation: Spectrum management. This field can be enable only in A band

7.4.2. parameter ::RDDurRegion

value
0: JAP for Japan
1: FCC for North America
2: CE for ETSI
Explanation: Set radar detection duration region.

7.4.3. parameter ::CSPeriod

value
0 ~ 255
Explanation: Channel switch period (Beacon count), unit is based on Beacon interval. The value indicate how many Channel-Switch Announcements will be sent.

7.4.4. parameter ::dfstest

value
0: Auto CTS, Depends on throughput loading to decrease CTS number, AP will select a properly CTS transmission rate.
1: Set to CTS count to fixed, High CTS transmission rate. It could increase DFS Hit rate but also impact throughput.
Note :
Use to improve the throughput on DFS test and throughput test.

8. Security Policy

8.1. All possible combinations of security policy

Type I. No Radius
(Must set parameter of IEEE8021X as FALSE)

	OPEN	SHARED	WEPAUTO
NONE	V	X	X
WEP	V	V	V
802.1x daemon	Off	Off	Off

Type II. With Radius (Non WiFi standard)
(Must set parameter of IEEE8021X as TRUE)

	OPEN
NONE	V
WEP	V
802.1x daemon	On

Type III. With WPA
(Must set parameter of IEEE8021X as FALSE)

	WPAPSK	WPA2PSK	WPAPSK WPA2PSK	WPA	WPA2	WPA WPA2
TKIP	V	V	V	V	V	V
AES	V	V	V	V	V	V
BOTH	V	V	V	V	V	V
802.1x daemon	Off	Off	Off	On	On	On

The "off" of 802.1x daemon means may be off, it also can be "on"
However "on" of 802.1x daemon means must be "on"
There are no relationship between the parameter of IEEE8021X and 802.1x daemon (rt61apd).

8.2. WPA2 setting

1. All settings are same as WPA, but modify attributes --- AuthMode, EncrypType, PreAuth, PMKCachePeriod.

8.3. Supported Parameters in RT61AP.dat

8.3.1. PreAuth=value

Value

0: Disable
1: Enable

Note:

Set WPA2 PMKID cache timeout period, after time out, the cached key will be delete

8.3.2. AuthMode=value

value

OPEN
SHARED
WEPAUTO
WPAPSK
WPA
WPA2PSK
WPA2
WPA1WPA2 :WPA/WPA2 mix mode
WPAPSKWPA2PSK :WPAPSK/WPA2PSK mix mode

NOTE :

WPA and analogous only support TKIP and AES as encryption method.
SHARED only supports Wep as encryption method.
WEPAUTO means AP can accept STA connect to it using **OPEN-WEP** or **SHARED-WEP**

8.3.3. EncrypType=value

value

NONE: For AuthMode=OPEN
WEP: For AuthMode=OPEN or AuthMode=SHARED
TKIP: For AuthMode=WPAPSK/WPA2PSK, WPA/WPA2, mix mode
AES: For AuthMode=WPAPSK/WPA2PSK, WPA/WPA2, mix mode
TKIPAES: TKIP/AES mix mode

8.3.4. DefaultKeyID=value

value

1 ~ 4 for WEP
2 for TKIP, AES(WPA/WPA2,WPAPSK/WPA2PSK)

8.3.5. Key1Type=value

value

0: Hexadecimal
1: Ascii

8.3.6. Key1Str=value

value

10 or 26 hexadecimal characters eg: 012345678
5 or 13 ascii characters eg: passd

8.3.7. Key2Type=value

value

0: Hexadecimal
1: Ascii

8.3.8. Key2Str=value

value

10 or 26 hexadecimal characters eg: 012345678
5 or 13 ascii characters eg: passd

8.3.9. Key3Type=value

value
0: Hexadecimal
1: Ascii

8.3.10. Key3Str=value

value
10 or 26 hexadecimal characters eg: 012345678
5 or 13 ascii characters eg: passd

8.3.11. Key4Type=value

value
0: Hexadecimal
1: Ascii

8.3.12. Key4Str=value

value
10 or 26 hexadecimal characters eg: 012345678
5 or 13 ascii characters eg: passd

8.3.13. WPAPSK=value

value
8 ~ 63 ascii characters
or
64 hexadecimal characters

8.3.14. RekeyMethod=value

Value (for WPA/WPA2)
TIME: Time rekey
PKT: Packet rekey
DISABLE: Disable rekey

8.3.15. RekeyInterval=value

Value (for WPA/WPA2)
0 ~ 0x3fffff ;unit:1seconds/1000packets

8.3.16. PMKCachePeriod=value

Value (for WPA2)
0 ~ ;unit:minute

8.4. iwpriv ra0 set [parameters]=[Val]

8.4.1. parameter :: PreAuth

[Val] range:
{0~1}
Explanation: Set WPA2 pre-authentication mode

8.4.2. parameter :: AuthMode

[Val] range:

{OPEN, WEPAUTO, SHARED, WPAPSK, WPA, WPA2PSK, WPA2, WPA1WPA2, WPAPSKWPA2PSK}
Explanation: Set Authentication mode

8.4.3. parameter :: EncrypType

[Val] range:
{NONE, WEP, TKIP, AES, TKIPAES}
Explanation: Set Encryption Type

8.4.4. parameter :: DefaultKeyID

[Val] range:
{1~4}
Explanation: Set Default Key ID
1 ~ 4 for WEP
2 for TKIP, AES(WPA/WPA2,WPAPSK/WPA2PSK)

8.4.5. parameter :: Key1

[Val] range:
{5 ascii characters or
10 hex number or
13 ascii characters or
26 hex numbers}
Explanation: Set Key1 String

8.4.6. parameter :: Key2

[Val] range:
{5 ascii characters or
10 hex number or
13 ascii characters or
26 hex numbers}
Explanation: Set Key2 String

8.4.7. parameter :: Key3

[Val] range:
{5 ascii characters or
10 hex number or
13 ascii characters or
26 hex numbers}
Explanation: Set Key3 String

8.4.8. parameter :: Key4

[Val] range:
{5 ascii characters or
10 hex number or
13 ascii characters or
26 hex numbers}
Explanation: Set Key4 String

8.4.9. parameter :: WPAPSK

[Val] range:
{8~63 ASCII or 64 HEX characters}
Explanation: WPA Pre-Shared Key

8.4.10. parameter :: RekeyMethod

[Val] range:
{TIME, PKT,NONE}

Explanation: Set group rekey interval-unit's type

8.4.11. parameter :: RekeyInterval

[Val] range:
{0~0x3fffffff}
Explanation: Set group rekey interval. 0 to disable rekey.
Unit:1seconds/1000packets dependent on Rekeytype

8.4.12. parameter :: PMKCachePeriod

[Val] range:
{0~ } unit: minute
Explanation: Set WPA2 PMKID cache timeout period, after time out, the cached key will be delete

8.5. Examples

8.5.1. Example I

On Step-by-Step setting of how to set SoftAP using WPAPSK security mechanism with encryption method TKIP. Assume rt61 softap set PreShared Key as "myownpresharedkey". Please ensure to set SSID, before/after set WPAPSK.

0. load rt61ap driver
1. iwpriv ra0 set AuthMode=WPAPSK
2. iwpriv ra0 set EncrypType=TKIP
3. iwpriv ra0 set IEEE8021X=0
4. iwpriv ra0 set SSID=myownssid
5. iwpriv ra0 set WPAPSK=myownpresharedkey
6. iwpriv ra0 set DefaultKeyID=2
7. iwpriv ra0 set SSID=myownssid

8.5.2. Example II

On Step-by-Step setting of how to set SoftAP using WEP security mechanism. Assume rt61 softap uses user-defined key.

0. load rt61ap driver
1. iwpriv ra0 set AuthMode=SHARED
2. iwpriv ra0 set EncrypType=WEP
3. iwpriv ra0 set IEEE8021X=0
4. iwpriv ra0 set Key1=0123456789
5. iwpriv ra0 set DefaultKeyID=1
6. iwpriv ra0 set SSID=myownssid

8.5.3. Example III

On Step-by-Step setting of how to set SoftAP using OPEN security mechanism.

0. load rt61ap driver
1. iwpriv ra0 set AuthMode=OPEN
2. iwpriv ra0 set EncrypType=NONE
3. iwpriv ra0 set IEEE8021X=0
4. iwpriv ra0 set SSID=myownssid

8.5.4. Example IV

Change setting to WPAPSK with AES.

0. iwpriv ra0 set AuthMode=WPAPSK
1. iwpriv ra0 set EncrypType=AES
2. iwpriv ra0 set IEEE8021X=0
3. iwpriv ra0 set SSID=MySsid
4. iwpriv ra0 set WPAPSK=MyPassword
5. iwpriv ra0 set DefaultKeyID=2
6. iwpriv ra0 set SSID=MySsid

Note1:
Step 3 is a must for calculating WPAPSK Key, which requires both SSID and WPAPSK.
Note2:
Step 5 will make driver to reload all settings. step5 must be the same with step3.

8.5.5. Example V

Change setting to OPEN, no 802.1x.

0. iwpriv ra0 set AuthMode= OPEN
1. iwpriv ra0 set EncrypType= NONE
2. iwpriv ra0 set IEEE8021X=0
3. iwpriv ra0 set SSID=MySsid

Note1:
Step 3 will make driver to reload all setting.

9. WDS

Wireleee Distribution System

9.1. WDS Setup

1. edit file in /etc/Wireless/RT61AP/RT61AP.dat to add
   (a). WdsEnable=1
   (b). WdsList=00:10:20:30:40:50; ;Another AP's MAC address
   (c). WdsEncrypType=NONE ;the encryption type in WDS interface
2. edit script file bridge_setup according to **the number of WDS-AP**
   add "/usr/sbin/brctl addif br0 ra1" and "/sbin/ifconfig ra1 0.0.0.0" to relative place
3. re-load driver(rt61ap.o)
4. run bridge_setup

9.2. WDS Usage

1. each WDS APs need seting as same channel, encryption type.(not support mixed mode, like WPAPSKWPA2PSK).
2. WDS Security support up to pre-shared key, this is inter AP's security and no 802.1x support.
3. In case want have auto-learning WDS peers, Lazy mode is the one. But have to note that can't set each AP to Lazy mode, otherwise no addr4 will be carried by each AP. This means that there at least has one AP have to fill WDS list.

9.3. Supported Parameters RT61AP.dat

9.3.1. WdsEnable=value

value

- **0: Disable**
  disable all Wds function.
- **1: Restrict mode**
  turn on Wds function, the peer Wds APs are according to the mac address listed in "WdsList" field below.
- **2: Bridge mode**
  turn on Wds function, the peer Wds APs are according to the mac address listed in "WdsList" field below and will also support Lazy mode. In this mode, AP will not send beacon out and will not deal with probe request packets, therefore STA will not possible to connect with it.
- **3: Repeater mode**
  turn on Wds function, same as restrict mode in functionality and also support Lazy mode.
- **4: Lazy mode**
  turn on Wds function, and auto learning from WDS packet which with addr4 field.

9.3.2. WdsList=value

value

[Mac Address];[Mac Address];...

Example:

00:10:20:30:40:50;0A:0b:0c:0D:0e:0f;1a:2b:3c:4d:5e:6f

Note :

max=4

9.3.3. WdsEncrypType=value

value

NONE
WEP
TKIP
AES

9.3.4. WdsKey=value

value

Depends on the setting of WdsEncrypType.

Main BSSID's EncrypType	WDS's WdsEncrypType	Peer AP WDS's WdsEncrypType	Remark
NONE	NONE	NONE	
WEP	WEP	WEP	Using legacy key setting method
TKIP	TKIP	TKIP	WDS's key is from WdsKey
TKIP	AES	AES	WDS's key is from WdsKey
AES	TKIP	TKIP	WDS's key is from WdsKey
AES	AES	AES	WDS's key is from WdsKey
TKIPAES	TKIP	TKIP	WDS's key is from WdsKey
TKIPAES	AES	AES	WDS's key is from WdsKey

9.4. iwpriv ra0 set [parameters]=[Val]

10. Authenticator

rt61apd - user space IEEE 802.1X Authenticator

10.1. Introduction

Rt61apd is an optional user space component for RT61 SoftAP driver.
It adds 802.1x Authenticator feature using external RADIUS Authentication Server(AS).

10.1.1. IEEE 802.1X features in rt61apd

IEEE Std 802.1X-2001 is a standard for port-based network access control. It introduces a extensible mechanism for authenticating and authorizing users.

Rt61apd implements partial IEEE 802.1x features that helps AS authorizing Supplicant and in the mean time proves itself a valid Authenticator for AS.
Noticed that Key management state machine is not included in rt61apd.
And those keys management is included in RT61 SoftAp driver.

Rt61apd relays the frames between the Supplicant and the AS.
Not until either one timeout or Success or Fail frame indicated does rt61apd finish the authentication process.
The port control entity is implemented in SoftAp driver for RT61.

10.1.2. How to start rt61apd

Manually start rt61apd, type

- $rt61apd

10.1.3. Support for WPA2

If the binding ethernet interface is not eth0, you need to modify "Ethifname=eth0" as whatever you need in RT61AP.dat

10.1.4. rt61apd configuration for IEEE 802.1X

Please add 4 required parameters in the configuration file for RT61 a/b/g SoftAp driver.

```
RADIUS_Server='192.168.2.3'
RADIUS_Port='1812'
RADIUS_Key='password'
own_ip_addr='your_ip_addr'
```

The word in ' ' must be replaced with your own correct setting.
Please make sure 'your_ip_addr' and RADIUS_Server is connected and RADIUS_Server's IAS (or related) services are started.

10.1.5. Support Multiple RADIUS Server

1. We use complier option to turn on/off the multiple RADIUS servers for 802.1x.

   If you want to enable the feature, make sure that "MULTIPLE_RADIUS" is defined in Makefile. Default is disabled.

   Besides, you must modify the file "RT61AP.dat" to co-operate with 802.1x. We add some variables to configure individual RADIUS server IP address, port and secret key per wireless interface.

   For example :
   RADIUS_Server_ra0=<ip_arrdess>
   RADIUS_Port_ra0=<port_number>
   RADIUS_Key_ra0=<secret_key>
   RADIUS_Server_ra1=<ip_arrdess>
   RADIUS_Port_ra1=<port_number>
   RADIUS_Key_ra1=<secret_key>
   .....
   .....
   RADIUS_Server_raN=<ip_arrdess>
   RADIUS_Port_raN=<port_number>
   RADIUS_Key_raN=<secret_key>

   If your wireless interface prefix is not "ra", please modify these variables.
   ex:
   If the wireless interface name is "wlan0", then
   RADIUS_Server_wlan0=<ip_arrdess>
   RADIUS_Port_wlan0=<port_number>
   RADIUS_Key_wlan0=<secret_key>

2. Then we can start to run 802.1x. The setting command is
   "rt61apd <wireless_if_name>"
   <wireless_if_name> is wireless interface name.(ex. ra0, ra1,.....)

   ex.
   "rt61apd ra0"
   "rt61apd ra1"

   After per task is started successfully, we record its Process ID in /var/run/auth_raN.pid. (N is integer)

   If "ra" is not the wireless interface prefix, please modify it in rt61apd.c.

10.2. Supported Parameters in RT61AP.dat

10.2.1. IEEE8021X=value

value

0: Disable

1: Enable

NOTE :

This field is enable only when Radius-WEP mode on, otherwise must disable

10.2.2. Ethifname=value

value

eth0

Interface to radius server, used by rt61apd

10.2.3. RADIUS_Server=xxx.xxx.xx.xx

IP for Radius server

10.2.4. RADIUS_Port=1812

This is port number for IAS service in Authentication Server(AS).
Default is 1812.

10.2.5. RADIUS_Key=value

value

string, suggested longer than 8 ascii characters.
This is Radius Secret shared with Authenticator and AS.

10.2.6. own_ip_addr=xxx.xxx.xx.xx

this is the ip address of our SoftAP.

10.2.7. session_timeout_interval = value

value

0, or >=60

0 to disable reauthentication for every session.
>=60 to set reauthenticaion interval with unit of second.

Note:

1. **xxx.xxx.xx.xx is a IP address**
2. *** represents the parameters for 802.1x daemon-rt61apd**

10.3. iwpriv ra0 set [parameters]=[Val]

10.3.1. parameter :: IEEE8021X

[Val] range:
{0, 1}
Explanation: Set 8021X-WEP mode on, this field is enable only when Radius-WEP or Radius-NONE mode on, otherwise must disable

10.4. Examples

10.4.1. Example I

On Step-by-Step setting of how to set SoftAP using WPA security mechanism. Assume rt61 softap has ip address 192.168.1.138, AS(Authentication Server) has IP address 192.168.1.1, Radius Secret is myownkey.

0. load rt61ap driver
   - $insmod rt61ap.o
1. First edit configuration file with correct value, esp. the following parameters that relate to the authentication features of RT61AP
   RADIUS_Server=192.168.1.1
   RADIUS_Port=1812
   RADIUS_Key=myownkey
   own_ip_addr=192.168.1.138
2. start rt61apd daemon by typing.
   - $rt61apd
3. iwpriv ra0 set AuthMode=WPA
4. iwpriv ra0 set EncrypType=TKIP
5. iwpriv ra0 set DefaultKeyID=2
6. iwpriv ra0 set IEEE8021X=0
7. iwpriv ra0 set SSID=myownssid

10.4.2. Example II

Change 802.1x settings to WPA with TKIP, using 802.1x authentication.

0. Modify 4 parameters
   RADIUS_Server=192.168.2.3
   RADIUS_Port=1812
   RADIUS_Key=password
   own_ip_addr=192.168.1.123 in the RT61AP.dat and save.
1. iwpriv ra0 set AuthMode=WPA
2. iwpriv ra0 set EncrypType=TKIP
3. iwpriv ra0 set IEEE8021X=0
4. iwpriv ra0 set SSID=myownssid

Note:
Step 4 restarts the rt61apd, and is essential.

10.4.3. Example III

Change setting to OPEN/WEP with 802.1x.

0. iwpriv ra0 set AuthMode= OPEN
1. iwpriv ra0 set EncrypType= WEP

2. iwpriv ra0 set IEEE8021X=1

Note1:
"IEEE8021X=1" only when Radius-WEP or Radius-NONE mode on, otherwise must "IEEE8021X=0".

10.4.4. Example IV

Change setting to OPEN/NONE with 802.1x.

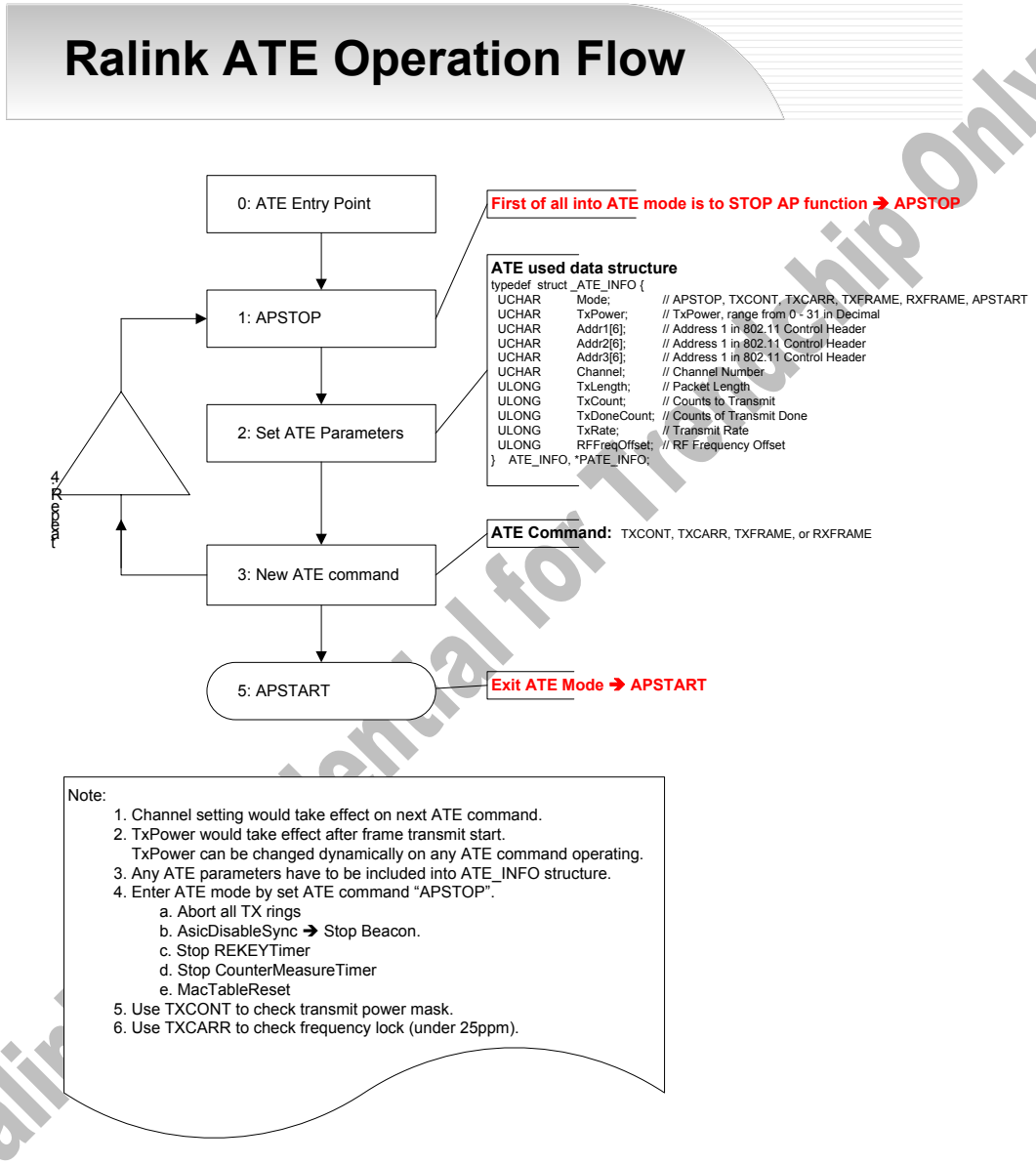
0. iwpriv ra0 set AuthMode= OPEN
1. iwpriv ra0 set EncrypType= NONE
2. iwpriv ra0 set IEEE8021X=1

Note1:
"IEEE8021X=1" only when Radius-WEP or Radius-NONE mode on , otherwise must "IEEE8021X=0".

11. ATE Test Command Format

******** IMPORTANT ********

If you are not familiar with hardware, it is recommanded not to modify hardware default value.



11.1. iwpriv ra0 set [parameters]=[val]

iwpriv ra0 set [parameters]=[val]
where

[parameters]	[val] constraints	explanation

11.1.1. parameter :: ATEDA
[val] constraints ::
xx:xx:xx:xx:xx:xx
Explanation: set ATE frame header addr1

11.1.2. parameter :: ATESA
[val] constraints ::
xx:xx:xx:xx:xx:xx
Explanation: set ATE frame header addr2

11.1.3. parameter :: ATEBSSID
[val] constraints ::
xx:xx:xx:xx:xx:xx
Explanation: set ATE frame header addr3

11.1.4. parameter :: ATETXPOW
[val] constraints ::
0 ~ 31
Explanation: set ATE Tx power

11.1.5. parameter :: ATECHANNEL
[val] constraints ::
802.11b/g: 1 ~ 14 depends on CountryRegion setting
802.11a : 36 ~ 165 depends on CountryRegion setting
Explanation: set ATE Channel

11.1.6. parameter :: ATETXFREQOFFSET
[val] constraints ::
0 ~ 63
Explanation: set ATE RF frequency offset

11.1.7. parameter :: ATETXLEN
[val] constraints ::
24 ~ 1500
Explanation: set ATE frame length

11.1.8. parameter :: ATETXCNT
[val] constraints ::
1 ~
Explanation: set ATE frame Tx count

11.1.9. parameter :: ATETXRATE
[val] constraints ::
0 ~ 11
Explanation: set ATE frame Tx rate (rate_1 ~ rate_54)

11.1.10. parameter :: ATERXFER
[val] constraints ::
0 : Disable counter show up
1 : Enable counter show up
Explanation: set ATE to periodic show up RxPER and RxTotalCount

11.1.11. parameter :: ATE

[val] constraints ::
- APSTOP
- APSTART
- TXCONT
- TXCARR
- TXFRAME
- RXFRAME

Explanation: set ATE actions
- APSTOP - stop AP & ATE function
- APSTART - start AP function
- TXCONT - start AP continuous TX
- TXCARR - start AP carrier test
- TXFRAME - transmit frame
- RXFRAME - continuous RX

11.2. iwpriv ra0 bbp [parameters]=[val]

iwpriv ra0 bbp [parameters]=[val]
where

[parameters]	[val] constraints	explanation

11.2.1. parameter :: 0 ~

[val] constraints ::
xx
Explanation: read/write BBP register

11.3. iwpriv ra0 mac [parameters]=[val]

iwpriv ra0 set mac=[val]
where

[parameters]	[val] constraints	explanation

11.3.1. parameter :: 0 ~

[val] constraints ::
xxxxxxxx
Explanation: read/write MAC register

11.4. iwpriv ra0 e2p [parameters]=[val]

iwpriv ra0 set e2p=[val]
where

[parameters]	[val] constraints	explanation

11.4.1. parameter :: 0 ~

[val] constraints ::
xxxx
Explanation: read/write E2PROM

11.5. Example

11.5.1. Set ATE associative argument

- **Check EVM & Power**

```
iwpriv ra0 set ATE=APSTOP                ; Stop AP working
iwpriv ra0 set ATEDA=00:11:22:33:44:55
iwpriv ra0 set ATESA=00:aa:bb:cc:dd:ee
iwpriv ra0 set ATEBSSID=00:11:22:33:44:55
iwpriv ra0 set ATETXRATE=11              ; Set Tx Rate 0~11
iwpriv ra0 set ATECHANNEL=1              ; Set Ate channel
iwpriv ra0 set ATETXLEN=1024             ; Tx frame length(no include 802.11 header)
iwpriv ra0 set ATETXPOW=18               ; Tx power(decmial)
iwpriv ra0 set ATETXCNT=100000           ; Tx frame count(decmial, this value must
                                         ; larger than meausurement period)
iwpriv ra0 set ATE=TXFRAME               ; Start Tx Frame
...                                      ; Measure EVM and Power with instrument
iwpriv ra0 set ATETXPOW=19               ; dynamic adjust Tx Power in decmial
...
iwpriv ra0 set ATETXPOW=20
...
iwpriv ra0 set ATE=APSTOP                ; Stop
```

- **Check Carrier**

```
iwpriv ra0 set ATE=APSTOP
iwpriv ra0 set ATETXRATE=11              ; Set Tx Rate 0~11
iwpriv ra0 set ATETXCNT=50               ; Tx frame count(decmial)
iwpriv ra0 set ATE=TXFRAME               ; Start Tx Frame(inform BBP to change
                                         ; modulation mode)
iwpriv ra0 set ATE=TXCARR                ; Start Tx carrier
...                                      ; Measure carrier with instrument
iwpriv ra0 set ATE=APSTOP
```

- **Check specturm mask**

```
iwpriv ra0 set ATE=APSTOP
iwpriv ra0 set ATETXRATE=11              ; Set Tx Rate 0~11
iwpriv ra0 set ATETXCNT=50               ; Tx frame count(decmial)
iwpriv ra0 set ATE=TXCONT                ; Start continuous TX
...                                      ; Measure specturm mask with instrument
iwpriv ra0 set ATE=APSTOP
```

- **Frequency offset tuning**

```
iwpriv ra0 set ATE=APSTOP
iwpriv ra0 set ATETXRATE=11              ; Set Tx Rate 0~11
iwpriv ra0 set ATETXCNT=50               ; Tx frame count(decmial)
iwpriv ra0 set ATE=TXFRAME               ; Start Tx Frame
iwpriv ra0 set ATETXFREQOFFSET=0         ; Set frequency offset 0(decimal)
iwpriv ra0 set ATE=TXCARR                ; Start Tx carrier
...                                      ; Measure carrier frequency with instrument
iwpriv ra0 set ATETXFREQOFFSET=10        ; Dynamic turning frequency offset
                                         ; 10(decimal)
...
iwpriv ra0 set ATETXFREQOFFSET=20        ; Dynamic turning frequency offset
                                         ; 20(decimal)
...
iwpriv ra0 set ATE=APSTOP                ; Stop
                                         ; Store the tuning result to EEPROM
```

- **Rx**

```
iwpriv ra0 set ATE=APSTOP
iwpriv ra0 set ResetCounter=0            ; Reset statistic counter
iwpriv ra0 set ATETXRATE=11              ; Set Tx Rate 0~11
iwpriv ra0 set ATE=RXFRAME               ; Start Rx
...                                      ; Transmit test packets
```

```
iwpriv ra0 set ATE=APSTOP                ; Stop
iwpriv ra0 stat
```

11.5.2. Hardware access

```
iwpriv ra0 bbp 0                         # read BBP register 0
iwpriv ra0 bbp 0=12                      # write BBP register 0 as 0x12
iwpriv ra0 mac 0                         # read MAC register 0
iwpriv ra0 mac 0=1234abcd                # write MAC register 0 as 0x1234abcd
iwpriv ra0 e2p 0                         # read E2PROM 0
iwpriv ra0 e2p c=12ab                    # write E2PROM 0xc as 0x12ab
```

11.5.3. Statistic counter operation

```
iwpriv ra0 stat                          # read statistic counter
iwpriv ra0 set ResetCounter=0            # reset statistic counter
```

11.5.4. Suggestion:

1. To turn on ATE functionality, you have to add compile flag "RALINK_ATE" to Makefile
2. Before doing ATE testing, please stop AP function
3. If you want to test another ATE action, prefer to stop AP & ATE function
4. All ATE function settings will lose efficacy after reboot.
5. Before hardware register access, please reference hardware spec.

Note.

In ATE mode, the channel must set via "ATECHANNEL"

12. AP Client

12.1. Introduction

The AP-Client function provides a 1-to-N MAC address mapping mechanism such that multiple stations behind the AP can transparently connect to the other AP even they didn't support WDS. When enable the AP-Client function, RT61 driver will create two interfaces, one is the AP interface which provide the features of Access Point, the other is the station interface used to connect to the remote AP. Besides, a software bridge function used to forwarding packets between this two interfaces.

The figure 1 shows the network topology and operation module of our AP-client function. The AP1 is an AP-Client feature enabled Access Point and have two wireless interfaces, ra0 and cli0, which provide the AP and station functions, respectively. The AP2 is a legacy Access Point that supports normal AP functions. STA1 associated to AP1 and the STA4 associated to AP2. In general, if the STA1 want to communicate with STA4, the AP2 and AP1 must support WDS or a physical network connection between AP1 and AP2. Now, with the support the AP-Client function, the AP1 can use build-in station interface cli0 connect to AP2, and then STA1 can communicate with STA4 transparently and didn't do any modifications. Also, the stations connect to the AP1 through the Ethernet line also can communicate with STA4 or access the Internet through AP2 transparently.

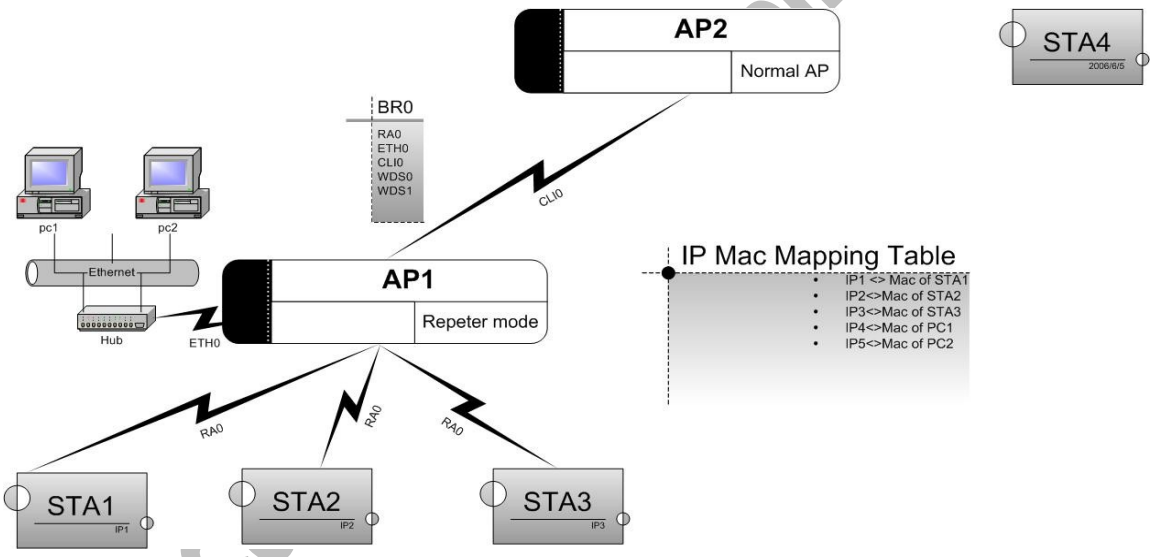


Figure 1. The network topology and operation module of AP-Client

Before enable the AP-Client feature, there are some restrictions need to remind

(1). Due to the limitation of 1-to-N MAC address mapping, our AP-Client function currently support following protocols:
- **(a).** All IP-based network applications
- **(b).** ARP
- **(c).** DHCP
- **(d).** PPPoE

(2). The last 2 hexadecimal number of the Mac address of our device must be the multiple of 2 or 4.

(3). The OS must provide a software bridge function can bridge multiple network interfaces.

It's simple to enable the feature of AP-Client, you just need to set the flag "HAS_APCLIENT" as "y" in the driver Makefile and got it.

12.2. Supported Parameters in RT61AP.dat

12.2.1. ApCliEnable=value

value
- 0: Disable
- 1: Enable

12.2.2. ApCliSsid=value

value
- 1~32 ascii characters

12.2.3. ApCliBssid=value

value
- [Mac Address]
- eg:
- 00:10:20:30:40:50

12.2.4. ApCliAuthMode=value

value
- OPEN
- SHARED
- WPAPSK
- WPA2PSK

Note:
- WPAPSK and analogous only support TKIP and AES as encryption method.
- SHARED only supports WEP as encryption method.

12.2.5. ApCliEncrypType=value

value
- NONE: For ApCliAuthMode=OPEN
- WEP: For ApCliAuthMode=OPEN or SHARED
- TKIP: For ApCliAuthMode=WPAPSK or WPA2PSK
- AES: For ApCliAuthMode=WPAPSK or WPA2PSK

12.2.6. ApCliWPAPSK=value

Value
- 8 ~ 63 ascii characters
- or
- 64 hexadecimal characters

12.2.7. ApCliDefaultKeyID=value

value
- 1 ~ 4 For WEP
- 2 For TKIP, AES(WPAPSK/WPA2PSK)

12.2.8. ApCliKey1Type=value

value
- 0: Hexadecimal
- 1: Ascii

12.2.9. ApCliKey1Str=value

value
- 10 or 26 hexadecimal characters eg: 012345678
- 5 or 13 ascii characters eg: passd

12.2.10. ApCliKey2Type=value

value

0: Hexadecimal
1: Ascii

12.2.11. ApCliKey2Str=value

value

10 or 26 hexadecimal characters eg: 012345678
5 or 13 ascii characters eg: passd

12.2.12. ApCliKey3Type=value

value

0: Hexadecimal
1: Ascii

12.2.13. ApCliKey3Str=value

value

10 or 26 hexadecimal characters eg: 012345678
5 or 13 ascii characters eg: passd

12.2.14. ApCliKey4Type=value

value

0: Hexadecimal
1: Ascii

12.2.15. ApCliKey4Str=value

value

10 or 26 hexadecimal characters eg: 012345678
5 or 13 ascii characters eg: passd

12.3. Setup AP Client

1. Edit file in /etc/Wireless/RT61AP/RT61AP.dat to add
   - **a)** ApCliEnable=1
   - **b)** ApCliSsid=AP2
   - **c)** ApCliBssid=00:10:20:30:40:50 (optional)
   - **d)** ApCliWPAPSK=87654321 (for WPAPSK or WPA2PSK mode)
   - **e)** ApCliAuthMode=WPAPSK
   - **f)** ApCliEncrypType=TKIP
   - **g)** ApCliDefaultKeyID=2
   - **h)** ApCliKey1Type=0 (the following for WEP)
   - **i)** ApCliKey1Str=
   - **j)** ApCliKey2Type=0
   - **k)** ApCliKey2Str=
   - **l)** ApCliKey3Type=0
   - **m)** ApCliKey3Str=
   - **n)** ApCliKey4Type=0
   - **o)** ApCliKey4Str=

2. Like the procedure of bringing up main BSSID (ra0), it also must to add "/sbin/ifconfig apcli0 up" and "/usr/sbin/brctl addif br0 apcli0".

3. The AP-client's security policy must be the same as main BSSID (ra0). So, if modifying these parameters "AuthMode" and "EncrypType" for main BSSID (ra0), it would affect the security policy of AP client (apcli0) at the same time.
   In the meantime, the whole AP's security policy only supports NONE, WEP (OPEN, SHARED), WPAPSK and WPA2PSK (TKIP, AES).

4. Set the "HAS_APCLIENT" flag as "y" in Makefile to enable or disable this function.
   If "y" selected, then MBSSID will be automatically enabled, AP-client use one of BSSIDs.

5. If enable AP client function, the multiple BSSID number would be 3 and the field 'BssidNum' shall larger than 1 and less than 3.

6. Users can also configure AP Client by iwpriv command.

12.4. WPS on AP Client

12.4.1. New command:

iwpriv apcli0 set **ApCliWscSsid**=xxxx

12.4.2. Support commands:

iwpriv apcli0 set WscConfMode=0 (Disable WPS) or 1 (Enrollee)
iwpriv apcli0 set WscMode=1 (PIN) or 2 (PBC)
iwpriv apcli0 set WscStatus=1
iwpriv apcli0 set WscGetConf=1
iwpriv apcli0 stat (check Pin Code)

12.4.3. NOT support commands:

iwpriv apcli0 set WscConfStatus=1 or 2
iwpriv apcli0 set WscOOB=1
iwpriv apcli0 set WscOpenOOB=1

12.4.4. NOT used commands:

iwpriv apcli0 set WscPinCode=xxxxxxxx

12.5. iwpriv apcli0 set [parameter]=[Val]

12.5.1. parameter :: ApCliEnable

[Val] range:
{0~1}
Explanation: Enable or disable the AP-Client
0: Disable
1: Enable

12.5.2. parameter :: ApCliSsid

[Val] range:
{0~z, less than 32 characters}
Explanation: Set SSID which AP client wants to join

12.5.3. parameter :: ApCliBssid

[Val] range:
{[MAC address]}
Explanation: Set BSSID which AP Client wants to join

p.s. It is an optional command. Users can indicate the desired BSSID by this command. Otherwise, AP Client can also get appropriate BSSID according to SSID automatically.

12.5.4. parameter :: ApCliAuthMode

[Val] range:
{OPEN, SHARED, WPAPSK, WPA2PSK}
Explanation: Set AP Client Authentication mode

12.5.5. parameter :: ApCliEncrypType

[Val] range:
{NONE, WEP, TKIP, AES}
Explanation: Set AP Client Encryption Type

12.5.6. parameter :: ApCliWPAPSK

[Val] range:
{8~63 ASCII or 64 HEX characters}
Explanation: AP Client WPA Pre-Shared Key

12.5.7. parameter :: ApCliDefaultKeyID

[Val] range:
{1~4}
Explanation: Set AP Client Default Key ID
1 ~ 4 for WEP
2 for TKIP, AES(WPAPSK/WPA2PSK)

12.5.8. parameter :: ApCliKey1

[Val] range:
{5 ascii characters or
10 hex number or
13 ascii characters or
26 hex numbers}
Explanation: Set AP Client Key1 String

12.5.9. parameter :: ApCliKey2

[Val] range:

{5 ascii characters or
10 hex number or
13 ascii characters or
26 hex numbers}
Explanation: Set AP Client Key2 String

12.5.10. parameter :: ApCliKey3

[Val] range:
{5 ascii characters or
10 hex number or
13 ascii characters or
26 hex numbers}
Explanation: Set AP Client Key3 String

12.5.11. parameter :: ApCliKey4

[Val] range:
{5 ascii characters or
10 hex number or
13 ascii characters or
26 hex numbers}
Explanation: Set AP Client Key4 String

12.5.12. parameter :: ApCliWscSsid

[Val] range:
{0~z, less than 32 characters}
Explanation: Set SSID which AP client wants to join

12.6. Example

12.6.1. Example I : Enable AP Client with none data security

1. iwpriv ra0 set AuthMode=OPEN
2. iwpriv ra0 set EncrypType=NONE
3. iwpriv ra0 set IEEE8021X=0
4. iwpriv apcli0 set ApCliSsid=AP2
5. iwpriv apcli0 set ApCliBssid=11:22:33:44:55:66 (Optional)
6. iwpriv ra0 set ApCliAuthMode=OPEN
7. iwpriv ra0 set ApCliEncrypType=NONE
8. iwpriv apcli0 set ApCliEnable=1
9. iwpriv ra0 set SSID=AP1

p.s. Must set ra0 SSID in step 9

12.6.2. Example II : OPEN WEP setting

1. iwpriv ra0 set AuthMode=OPEN
1. iwpriv ra0 set EncrypType=WEP
2. iwpriv ra0 set IEEE8021X=0
3. iwpriv ra0 set DefaultKeyID=1
4. iwpriv ra0 set Key1=1234567890
5. iwpriv apcli0 set ApCliSsid=AP2
6. iwpriv ra0 set ApCliAuthMode=OPEN
7. iwpriv ra0 set ApCliEncrypType=WEP
8. iwpriv ra0 set ApCliDefaultKeyID=3
9. iwpriv ra0 set ApCliKey3=1234567890
10. iwpriv apcli0 set ApCliEnable=1
11. iwpriv ra0 set SSID=AP1

p.s. Must set ra0 SSID in step 12

12.6.3. Example III : Shared WEP setting

1. iwpriv ra0 set AuthMode=SHARED
2. iwpriv ra0 set EncrypType=WEP
3. iwpriv ra0 set IEEE8021X=0
4. iwpriv ra0 set DefaultKeyID=1
5. iwpriv ra0 set Key1=1234567890
6. iwpriv apcli0 set ApCliSsid=AP2
7. iwpriv ra0 set ApCliAuthMode=SHARED
8. iwpriv ra0 set ApCliEncrypType=WEP
9. iwpriv ra0 set ApCliDefaultKeyID=2
10. iwpriv ra0 set ApCliKey2=0987654321
11. iwpriv apcli0 set ApCliEnable=1
12. iwpriv ra0 set SSID=AP1

p.s. Must set ra0 SSID in step 12

12.6.4. Example IV : WPAPSK-TKIP setting

1. iwpriv ra0 set AuthMode=WPAPSK
2. iwpriv ra0 set EncrypType=TKIP
3. iwpriv ra0 set IEEE8021X=0
4. iwpriv ra0 set SSID=AP1
5. iwpriv ra0 set WPAPSK=12345678
6. iwpriv ra0 set DefaultKeyID=2
7. iwpriv apcli0 set ApCliSsid=AP2
8. iwpriv ra0 set ApCliAuthMode=WPAPSK
9. iwpriv ra0 set ApCliEncrypType=TKIP
10. iwpriv apcli0 set ApCliEnable=1
11. iwpriv ra0 set ApCliDefaultKeyID=2
12. iwpriv apcli0 set ApCliWPAPSK=87654321
13. iwpriv ra0 set SSID=AP1

p.s. Must set ra0 SSID again in step 13

12.6.5. Example V : WPA2PSK-AES setting

1. iwpriv ra0 set AuthMode=WPA2PSK
2. iwpriv ra0 set EncrypType=AES
3. iwpriv ra0 set IEEE8021X=0
4. iwpriv ra0 set SSID=AP1
5. iwpriv ra0 set WPAPSK=12345678
6. iwpriv ra0 set DefaultKeyID=2
7. iwpriv apcli0 set ApCliSsid=AP2
8. iwpriv ra0 set ApCliAuthMode=WPA2PSK
9. iwpriv ra0 set ApCliEncrypType=AES
10. iwpriv apcli0 set ApCliEnable=1
11. iwpriv apcli0 set ApCliWPAPSK=87654321
12. iwpriv ra0 set ApCliDefaultKeyID=2
13. iwpriv ra0 set SSID=AP1

p.s. Must set ra0 SSID again in step 13

12.6.6. Example VI: main BSSID WPAPSKWPA2PSK/TKIPAES mixed mode, AP Client Shared/WEP

1. iwpriv ra0 set AuthMode=WPAPSKWPA2PSK
2. iwpriv ra0 set EncrypType=TKIPAES
3. iwpriv ra0 set IEEE8021X=0
4. iwpriv ra0 set SSID=AP1
5. iwpriv ra0 set WPAPSK=12345678
6. iwpriv ra0 set DefaultKeyID=2
7. iwpriv apcli0 set ApCliSsid=AP2
8. iwpriv ra0 set ApCliAuthMode=SHARED
9. iwpriv ra0 set ApCliEncrypType=WEP
10. iwpriv apcli0 set ApCliEnable=1
11. iwpriv ra0 set ApCliDefaultKeyID=1
12. iwpriv ra0 set ApCliKey1=0987654321
13. iwpriv ra0 set SSID=AP1

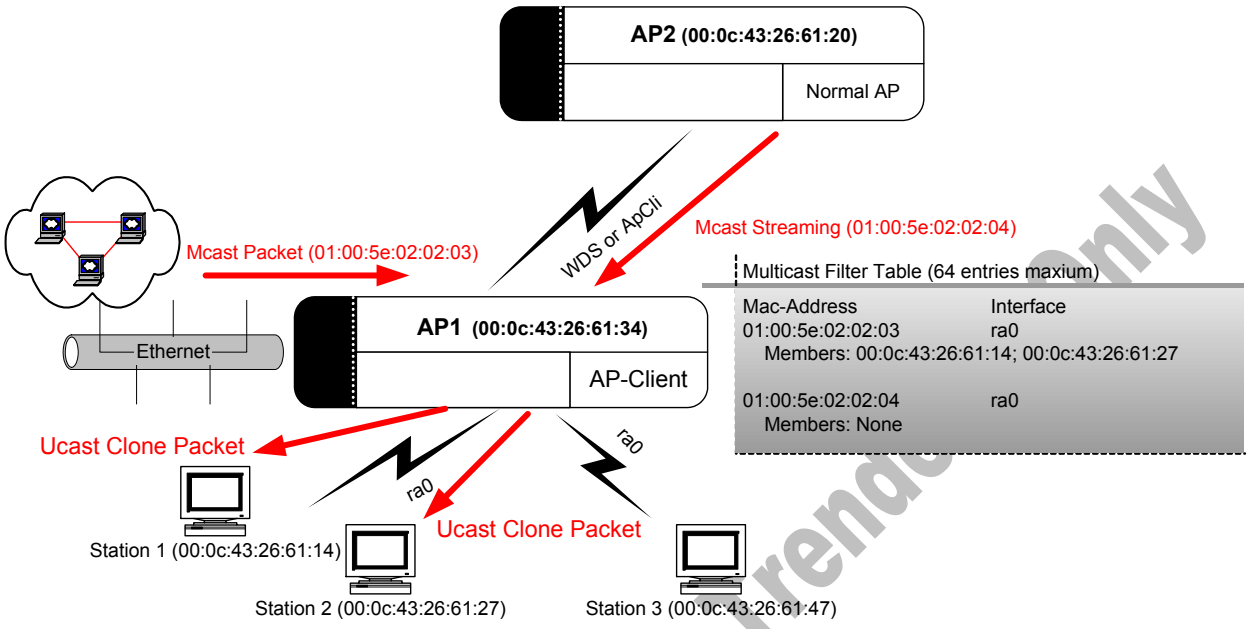
p.s. Must set ra0 SSID again in step 13

12.6.7. Example VII: Setup ApClient WPS

14. iwpriv ra0 set AuthMode=WPAPSKWPA2PSK
15. iwpriv ra0 set EncrypType=TKIPAES
16. iwpriv ra0 set IEEE8021X=0
17. iwpriv ra0 set SSID=AP1
18. iwpriv ra0 set WPAPSK=12345678
19. iwpriv ra0 set DefaultKeyID=2
20. iwpriv apcli0 set ApCliSsid=AP2
21. iwpriv ra0 set ApCliAuthMode=SHARED
22. iwpriv ra0 set ApCliEncrypType=WEP
23. iwpriv apcli0 set ApCliEnable=1
24. iwpriv ra0 set ApCliDefaultKeyID=1
25. iwpriv ra0 set ApCliKey1=0987654321
26. iwpriv ra0 set SSID=AP1

p.s. Must set ra0 SSID again in step 13

13. IGMP Snooping



13.1. IGMP Table Learning:

An IGMP table entry consists of Group-Id (Multicast MAC Address), Net-Interface and Member-List. For example, in the picture above we see the "Multicast Filter Table" of AP1 have two IGMP entries. One is "01:00:5e:02:02:03" with two members and another is "01:00:5e:02:02:04 with empty member list". AP will automatically insert or remove the entry from table by snooping the IGMP-Membership report packet from Station behind AP. And it also could be manual add and del by iwpriv command.

13.2. Multicast Packet Process:

Once a multicast packet whether it comes from portal, WDS or AP-Client. AP will go through the Multicast-filter table to find a match rule for the incoming packet. If have no any match rule in the table then AP will simply drops it. If it does then there are two cases how AP handles a multicast packet. The first cast is the match entry has no member then AP just forwards it to all stations behind the net-interface. If the match entry has members then AP will do unicast clone for all members.

For example, AP1 receive a multicast packet with group-Id, "01:00:5e:02:02:03", comes from Ethernet then AP1 check the multicast table using group-Id and fount it match the entry with 2 members. So AP1 clone the multicast packet and sent them to Station 1 and Station 2. Another case a multicast packet with group-id (01:00:5e:02:02:04) be sent to AP1 then AP1 just forward it to all Stations behind interface, ra0 since the match entry have no member.

13.3. Iwpriv command for IGMP-Snooping:

13.3.1. IgmpSnEnable

The IGMP snooping function and multicast packet filter can be enabled or disabled at running time by iwpriv command "set IgmpSnEnable=<0|1>".

For example:

iwpriv ra0 set IgmpSnEnable=1

iwpriv ra0 set IgmpSnEnable=0

13.3.2. IgmpAdd :: Group-ID

It also provide a command let user add a entry by iwpriv command "set IgmpAdd=<Group-ID>", Group-ID could be a MAC address or a IP address.

For example:

iwpriv ra0 set IgmpAdd=226.2.2.3
iwpriv ra0 set IgmpAdd=01:00:5e:02:02:03

13.3.3. IgmpAdd :: Group-Member

Or just add members into a Group by command "set IgmpAdd=<Group-ID-[Member]-… >", Group-ID could be a MAC address or a IP address.

For example:

iwpriv ra0 set IgmpAdd=226.2.2.3-00:0c:43:26:61:27-00:0c:43:26:61:28
iwpriv ra0 set IgmpAdd=01:00:5e:02:02:03-00:0c:43:26:61:27-00:0c:43:26:61:28

13.3.4. IgmpDel::Group-ID

Also the entry can be deleted by command "set IgmpDelEntry=<Group-ID>".

For example:

iwpriv ra0 set IgmpDel=226.2.2.3
iwpriv ra0 set IgmpDel=01:00:5e:02:02:03

13.3.5. IgmpDel::Group-Member

Or just delete a member from a Group by command "set IgmpDel=<Group-ID-[Member]-… >", Group-ID could be a MAC address or a IP address.

For example:

iwpriv ra0 set IgmpDel=226.2.2.3-00:0c:43:26:61:27-00:0c:43:26:61:28
iwpriv ra0 set IgmpDel=01:00:5e:02:02:03-00:0c:43:26:61:27-00:0c:43:26:61:28

13.3.6. IgmpTabShow

There's a one more command to show up the multiple-cast filter table.

For example:

Iwpriv ra0 set IgmpTabShow=1

14. SNMP MIBs

14.1. RT61AP Supported v.s. IEEE802dot11-MIB

IEEE802dot11-MIB	Access	Sup-port	OID	RT61AP .dat
ieee802dot11				
dot11smt		-		
dot11StationConfigTable	not-accessible	-		
dot11StationConfigEntry	not-accessible	-		
dot11StationID	read-write	Y	OID_802_3_CURRENT_ADDRESS	N
dot11MediumOccupancyLimit	read-write	N		N
dot11CFPollable	read-only	N		N
dot11CFPPeriod	read-write	N		N
dot11CFPMaxDuration	read-write	N		N
dot11AuthenticationResponseTimeOut	read-write	N		N
dot11PrivacyOptionImplemented	read-only	Y	RT_OID_802_11_PRIVACYOPTIONIMPLEMENTED	N
dot11PowerManagementMode	read-write	Y	RT_OID_802_11_POWERMANAGEMENTMODE	N
dot11DesiredSSID	read-write	N		N
dot11DesiredBSSType	read-write	N		N
dot11OperationalRateSet	read-write	N		N
dot11BeaconPeriod	read-write	N		N
dot11DTIMPeriod	read-write	N		N
dot11AssociationResponseTimeOut	read-write	N		N
dot11DisassociateReason	read-only	N		N
dot11DisassociateStation	read-only	N		N
dot11DeauthenticateReason	read-only	N		N
dot11DeauthenticateStation	read-only	N		N
dot11AuthenticateFailStatus	read-only	N		N
dot11AuthenticateFailStation	read-only	N		N
dot11AuthenticationAlgorithmsTable	not-accessible	-		-
dot11AuthenticationAlgorithmsEntry	not-accessible	-		-
dot11AuthenticationAlgorithmsIndex	not-accessible	Y		N
dot11AuthenticationAlgorithm	read-only	Y		N
dot11AuthenticationAlgorithmsEnable	read-write	Y		N
dot11WEPDefaultKeysTable	not-accessible	-		-
dot11WEPDefaultKeysEntry	not-accessible	-		-
dot11WEPDefaultKeyIndex	not-accessible	Y		N
dot11WEPDefaultKeyValue	read-write	Y	OID_802_11_WEPDEFAULTKEYVALUE	Y
dot11WEPKeyMappingsTable	not-accessible	-		-
dot11WEPKeyMappingsEntry	not-accessible	-		-
dot11WEPKeyMappingIndex	not-accessible	N		N
dot11WEPKeyMappingAddress	read-create	N		N
dot11WEPKeyMappingWEPOn	read-create	N		N
dot11WEPKeyMappingValue	read-create	N		N
dot11WEPKeyMappingStatus	read-create	N		N

dot11PrivacyTable	not-accessible	-		
dot11PrivacyEntry	not-accessible	-		
dot11PrivacyInvoked	read-write	Y		N
dot11WEPDefaultKeyID	read-write	Y	OID_802_11_WEPDEFAULTKEYID	Y
dot11WEPKeyMappingLength	read-write	Y	RT_OID_802_11_WEPKEYMAPPINGLENGTH	N
dot11ExcludeUnencrypted	read-write	N		N
dot11WEPICVErrorCount	read-only	N		N
dot11WEPExcludedCount	read-only	N		N
dot11SMTnotification	-	-		
dot11Disassociate	-	N		N
dot11Deauthenticate	-	N		N
dot11AuthenticateFail	-	N		N
dot11mac				
dot11OperationTable	not-accessible	-		
dot11OperationEntry	not-accessible	-		
dot11MACAddress	read-only	Y	RT_OID_802_11_MAC_ADDRESS	N
dot11RTSThreshold	read-write	Y	OID_802_11_RTS_THRESHOLD	Y
dot11ShortRetryLimit	read-write	Y	OID_802_11_SHORTRETRYLIMIT	N
dot11LongRetryLimit	read-write	Y	OID_802_11_LONGRETRYLIMIT	N
dot11FragmentationThreshold	read-write	Y	OID_802_11_FRAGMENTATION_THRESHOLD	Y
dot11MaxTransmitMSDULifetime	read-write	N		N
dot11MaxReceiveLifetime	read-write	N		N
dot11ManufacturerID	read-only	Y	RT_OID_802_11_MANUFACTUREID	N
dot11ProductID	read-only	Y	RT_OID_802_11_PRODUCTID	N
dot11CountersTable	not-accessible	-		
dot11CountersEntry	not-accessible	-		
dot11TransmittedFragmentCount	read-only	Y	OID_802_11_STATISTICS	N
dot11MulticastTransmittedFrameCount	read-only	Y	OID_802_11_STATISTICS	N
dot11FailedCount	read-only	Y	OID_802_11_STATISTICS	N
dot11RetryCount	read-only	Y	OID_802_11_STATISTICS	N
dot11MultipleRetryCount	read-only	Y	OID_802_11_STATISTICS	N
dot11FrameDuplicateCount	read-only	Y	OID_802_11_STATISTICS	N
dot11RTSSuccessCount	read-only	Y	OID_802_11_STATISTICS	N
dot11RTSFailureCount	read-only	Y	OID_802_11_STATISTICS	N
dot11ACKFailureCount	read-only	Y	OID_802_11_STATISTICS	N
dot11ReceivedFragmentCount	read-only	Y	OID_802_11_STATISTICS	N
dot11MulticastReceivedFrameCount	read-only	Y	OID_802_11_STATISTICS	N
dot11FCSErrorCount	read-only	Y	OID_802_11_STATISTICS	N
dot11TransmittedFrameCount	read-only	N		N
dot11WEPUndecryptableCount	read-only	N		N
dot11GroupAddressesTable	not-accessible	-		-
dot11GroupAddressesEntry	not-accessible	-		-
dot11GroupAddressesIndex	not-accessible	N		N
dot11Address	read-create	N		N
dot11GroupAddressesStatus	read-create	N		N
dot11res				

dot11resAttribute				
dot11ResourceTypeIDName	read-only	-		
dot11ResourceInfoTable	not-accessible	-		
dot11ResourceInfoEntry	not-accessible	-		
dot11manufacturerOUI	read-only	Y	RT_OID_802_11_MANUFACTUREROUI	N
dot11manufacturerName	read-only	Y	RT_OID_802_11_MANUFACTURERNAME	N
dot11manufacturerProductName	read-only	Y	RT_OID_DEVICE_NAME	N
dot11manufacturerProductVersion	read-only	Y	RT_OID_VERSION_INFO	N
dot11phy				
dot11PhyOperationTable	not-accessible	-		
dot11PhyOperationEntry	not-accessible	-		
dot11PHYType	read-only	Y	RT_OID_802_11_PHY_MODE	N
dot11CurrentRegDomain	read-write	Y		Y
dot11TempType	read-only	N		N
dot11PhyAntennaTable	not-accessible	-		
dot11PhyAntennaEntry	not-accessible	-		
dot11CurrentTxAntenna	read-write	Y	OID_802_11_TX_ANTENNA_SELECTED	N
dot11DiversitySupport	read-only	Y	OID_802_11_RX_ANTENNA_SELECTED	N
dot11CurrentRxAntenna	read-write	Y	OID_802_11_RX_ANTENNA_SELECTED	N
dot11PhyTxPowerTable	not-accessible	-		
dot11PhyTxPowerEntry	not-accessible	-		
dot11NumberSupportedPowerLevels	read-only	N		N
dot11TxPowerLevel1	read-only	N		N
dot11TxPowerLevel2	read-only	N		N
dot11TxPowerLevel3	read-only	N		N
dot11TxPowerLevel4	read-only	N		N
dot11TxPowerLevel5	read-only	N		N
dot11TxPowerLevel6	read-only	N		N
dot11TxPowerLevel7	read-only	N		N
dot11TxPowerLevel8	read-only	N		N
dot11CurrentTxPowerLevel	read-write	N		N
dot11PhyFHSSTable	not-accessible	-		
dot11PhyFHSSEntry	not-accessible	-		
dot11HopTime	read-only	N		N
dot11CurrentChannelNumber	read-write	N		N
dot11MaxDwellTime	read-only	N		N
dot11CurrentDwellTime	read-write	N		N
dot11CurrentSet	read-write	N		N
dot11CurrentPattern	read-write	N		N
dot11CurrentIndex	read-write	N		N
dot11PhyDSSSTable	not-accessible	-		
dot11PhyDSSSEntry	not-accessible	-		
dot11CurrentChannel	read-write	Y	OID_802_11_CURRENTCHANNEL	Y
dot11CCAModeSupported	read-only	N		N
dot11CurrentCCAMode	read-write	N		N
dot11EDThreshold	read-write	N		N

dot11PhyIRTable	not-accessible	-		
dot11PhyIREntry	not-accessible	-		
dot11CCAWatchdogTimerMax	read-write	N		N
dot11CCAWatchdogCountMax	read-write	N		N
dot11CCAWatchdogTimerMin	read-write	N		N
dot11CCAWatchdogCountMin	read-write	N		N
dot11RegDomainsSupportedTable	not-accessible	-		
dot11RegDomainsSupportEntry	not-accessible	-		
dot11RegDomainsSupportIndex	not-accessible	Y		N
dot11RegDomainsSupportValue	read-only	Y		N
dot11AntennasListTable	not-accessible	-		
dot11AntennasListEntry	not-accessible	-		
dot11AntennaListIndex	not-accessible	Y		N
dot11SupportedTxAntenna	read-write	Y	OID_802_11_TX_ANTENNA_SELECTED	N
dot11SupportedRxAntenna	read-write	Y	OID_802_11_RX_ANTENNA_SELECTED	N
dot11DiversitySelectionRx	read-write	Y	OID_802_11_RX_ANTENNA_SELECTED	N
dot11SupportedDataRatesTxTable	not-accessible	-		
dot11SupportedDataRatesTxEntry	not-accessible	-		
dot11SupportedDataRatesTxIndex	not-accessible	Y		N
dot11SupportedDataRatesTxValue	read-only	Y	OID_802_11_DESIRED_RATES	N
dot11SupportedDataRatesRxTable	not-accessible	-		
dot11SupportedDataRatesRxEntry	not-accessible	-		
dot11SupportedDataRatesRxIndex	not-accessible	Y	OID_802_11_DESIRED_RATES	
dot11SupportedDataRatesRxValue	read-only	Y		
dot11PhyOFDMTable	not-accessible	-		
dot11PhyOFDMEntry	not-accessible	-		
dot11CurrentFrequency	read-write	N	OID_802_11_CURRENTCHANNEL	Y
dot11TIThreshold	read-write	N		N
dot11FrequencyBandsSupported	read-only	N		N

14.2. RALINK OID for SNMP MIB

RALINK OID for SNMP		
Value	**Name**	**Structure**
0x010B	OID_802_11_NUMBER_OF_ANTENNAS	USHORT numant;
0x010C	OID_802_11_RX_ANTENNA_SELECTED	USHORT whichant;
0x010D	OID_802_11_TX_ANTENNA_SELECTED	USHORT whichant;
0x050C	RT_OID_802_11_PHY_MODE	ULONG linfo;
0x050E	OID_802_11_DESIRED_RATES	typedef UCHAR NDIS_802_11_RATES[NDIS_802_11_LENGTH_RATES]; #define NDIS_802_11_LENGTH_RATES 8
0x0514	OID_802_11_RTS_THRESHOLD	ULONG linfo;
0x0515	OID_802_11_FRAGMENTATION_THRESHOLD	ULONG linfo;
0x0607	RT_OID_DEVICE_NAME	char name[128];
0x0608	RT_OID_VERSION_INFO	typedef struct PACKED _RT_VERSION_INFO{ UCHAR DriverVersionW; UCHAR DriverVersionX; UCHAR DriverVersionY; UCHAR DriverVersionZ; UINT DriverBuildYear; UINT DriverBuildMonth; UINT DriverBuildDay; } RT_VERSION_INFO, *PRT_VERSION_INFO;
0x060A	OID_802_3_CURRENT_ADDRESS	char addr[128];

0x060E	OID_802_11_STATISTICS	typedef struct _NDIS_802_11_STATISTICS
		{
		ULONG Length; // Length of structure
		ULONG TransmittedFragmentCount;
		ULONG MulticastTransmittedFrameCount;
		ULONG FailedCount;
		ULONG RetryCount;
		ULONG MultipleRetryCount;
		ULONG RTSSuccessCount;
		ULONG RTSFailureCount;
		ULONG ACKFailureCount;
		ULONG FrameDuplicateCount;
		ULONG ReceivedFragmentCount;
		ULONG MulticastReceivedFrameCount;
		ULONG FCSErrorCount;
		} NDIS_802_11_STATISTICS, PNDIS_802_11_STATISTICS;

0x0700	RT_OID_802_11_MANUFACTUREROUI	char oui[128];
0x0701	RT_OID_802_11_MANUFACTURERNAME	char name[128];
0x0702	RT_OID_802_11_RESOURCETYPEIDNAME	char name[128];
0x0703	RT_OID_802_11_PRIVACYOPTIONIMPLEMENTED	ULONG linfo;
0x0704	RT_OID_802_11_POWERMANAGEMENTMODE	ULONG linfo;
0x0705	OID_802_11_WEPDEFAULTKEYVALUE	typedef struct _DefaultKeyIdxValue { UCHAR KeyIdx; UCHAR Value[16]; }DefaultKeyIdxValue;
0x0706	OID_802_11_WEPDEFAULTKEYID	UCHAR keyid;
0x0707	RT_OID_802_11_WEPKEYMAPPINGLENGTH	UCHAR len;
0x0708	OID_802_11_SHORTRETRYLIMIT	ULONG linfo;
0x0709	OID_802_11_LONGRETRYLIMIT	ULONG linfo;
0x0710	RT_OID_802_11_PRODUCTID	char id[128];
0x0711	RT_OID_802_11_MANUFACTUREID	char id[128];
0x0712	OID_802_11_CURRENTCHANNEL	UCHAR channel
0x0713	RT_OID_802_11_MAC_ADDRESS	char macaddress[128]

15. IOCTL – I/O Control Interface

15.1. Parameters for iwconfig's IOCTL

Access	Description	ID	Parameters
Get	BSSID, MAC Address	SIOCGIFHWADDR	wrq->u.name, (length = 6)
	WLAN Name	SIOCGIWNAME	wrq->u.name = "RT61 SoftAP", length = strlen(wrq->u.name)

	SSID	SIOCGIWESSID	struct iw_point *erq = &wrq->u.essid;
			erq->flags=1;
			erq->length = pAd->PortCfg.MBSSID[pAd->IoctlIF].SsidLen;
			if(erq->pointer)
			{
			if(copy_to_user(erq->pointer,
			pAd->PortCfg.MBSSID[pAd->IoctlIF].Ssid,
			erq->length))
			{
			Status = -EFAULT;
			break;
			}
			}

	Channel / Frequency (Hz)	SIOCGIWFREQ	wrq->u.freq.m = pAd->PortCfg.Channel;
			wrq->u.freq.e = 0;
			wrq->u.freq.i = 0;

	Bit Rate (bps)	SIOCGIWRATE	wrq->u.bitrate.value = RateIdTo500Kbps[pAd->PortCfg.MBSSID[pAd->IoctlIF].TxRate] * 500000; wrq->u.bitrate.disabled = 0;
	AP's MAC address	SIOCGIWAP	wrq->u.ap_addr.sa_family = ARPHRD_ETHER; memcpy(wrq->u.ap_addr. sa_data, &pAd->PortCfg.MBSSID[pAd->IoctlIF].Bssid, ETH_ALEN);
	Operation Mode	SIOCGIWMODE	wrq->u.mode = IW_MODE_INFRA;
	Range of Parameters	SIOCGIWRANGE	range.we_version_compiled = WIRELESS_EXT; range.we_version_source = 14;

	Scanning Results	SIOCGIWSCAN	typedef struct _NDIS_802_11_SITE_SURVEY_TABLE
			{
			LONG Channel;
			LONG Rssi;
			UCHAR Ssid[33];
			UCHAR Bssid[18];
			UCHAR EncrypT[8];
			} NDIS_802_11_SITE_SURVEY_TABLE,
			*PNDIS_802_11_SITE_SURVEY_TABLE;
			
			wrq->u.data.length = N* sizeof(NDIS_802_11_SITE_SURVEY_TABLE);
			copy_to_user(wrq->u.data.pointer, site_survey_table, wrq->u.data.length);

	Client Association List	SIOCGIWAPLIST	typedef struct _NDIS_802_11_STATION_TABLE
			{
			UCHAR MacAddr[18];
			ULONG Aid;
			ULONG PsMode;
			ULONG LastDataPacketTime;
			ULONG RxByteCount;
			ULONG TxByteCount;
			ULONG CurrTxRate;
			ULONG LastTxRate;
			} NDIS_802_11_STATION_TABLE, *PNDIS_802_11_STATION_TABLE;
			
			wrq->u.data.length = i * sizeof(NDIS_802_11_STATION_TABLE);
			copy_to_user(wrq->u.data.pointer, sta_list_table, wrq->u.data.length);

Set	Trigger Scanning	SIOCSIWSCAN	ApSiteSurvey(pAd);

15.2. Parameters for iwpriv's IOCTL

Please refer section 4 and 5 to have iwpriv parameters and values.

Parameters:

```
int     socket_id;
char    name[25];          // interface name
char    data[255];         // command string
struct  iwreq wrq;
```

Default setting:

```
wrq.ifr_name = name = "ra0";          // interface name
wrq.u.data.pointer = data;            // data buffer of command string
wrq.u.data.length = strlen(data);     // length of command string
wrq.u.data.flags = 0;
```

15.2.1. Set Data, Parameters is Same as iwpriv

Command and IOCTL Function		
Set Data		
Function Type	**Command**	**IOCTL**

RTPRIV_IOCTL_SET	iwpriv ra0 **set SSID=RT61AP**	sprintf(name, "ra0");
		strcpy(data, "**SSID=RT61AP**");
		strcpy(wrq.ifr_name, name);
		wrq.u.data.length = strlen(data);
		wrq.u.data.pointer = data;
		wrq.u.data.flags = 0;
		ioctl(socket_id, RTPRIV_IOCTL_SET, &wrq);

15.2.2. Get Data, Parameters is Same as iwpriv

Command and IOCTL Function		
Get Data		
Function Type	**Command**	**IOCTL**

RTPRIV_IOCTL_STATISTICS	Iwpriv ra0 **stat**	sprintf(name, "ra0");
		strcpy(data, "**stat**");
		strcpy(wrq.ifr_name, name);
		wrq.u.data.length = strlen(data);
		wrq.u.data.pointer = data;
		wrq.u.data.flags = 0;
		ioctl(socket_id, RTPRIV_IOCTL_STATISTICS, &wrq);

RTPRIV_IOCTL_GSITESURVEY	Iwpriv ra0 **get_site_survey**	sprintf(name, "ra0");
		strcpy(data, "**get_site_survey**");
		strcpy(wrq.ifr_name, name);
		wrq.u.data.length = strlen(data);
		wrq.u.data.pointer = data;
		wrq.u.data.flags = 0;
		ioctl(socket_id, RTPRIV_IOCTL_GSITESURVEY, &wrq);

RTPRIV_IOCTL_GET_MAC_TABLE	Iwpriv ra0 **get_mac_table**	sprintf(name, "ra0");
		strcpy(data, "**get_mac_table**");
		strcpy(wrq.ifr_name, name);
		wrq.u.data.length = strlen(data);
		wrq.u.data.pointer = data;
		wrq.u.data.flags = 0;
		ioctl(socket_id, RTPRIV_IOCTL_GET_MAC_TABLE, &wrq);

15.2.3. Set Data: BBP, MAC and EEPROM

Command and IOCTL Function		
Set Data: BBP, MAC and EEPROM, Parameters is Same as iwpriv		
Type	**Command**	**IOCTL**

RTPRIV_IOCTL_BBP (Set BBP Register Value)	Iwpriv ra0 **bbp 17=32**	sprintf(name, "ra0");
		strcpy(data, " **bbp 17=32**");
		strcpy(wrq.ifr_name, name);
		wrq.u.data.length = strlen(data);
		wrq.u.data.pointer = data;
		wrq.u.data.flags = 0;
		ioctl(socket_id, **RTPRIV_IOCTL_BBP**, &wrq);

RTPRIV_IOCTL_MAC (Set MAC Register Value)	Iwpriv ra0 **mac 3000=12345678**	sprintf(name, "ra0");
		strcpy(data, " **mac 3000=12345678**");
		strcpy(wrq.ifr_name, name);
		wrq.u.data.length = strlen(data);
		wrq.u.data.pointer = data;
		wrq.u.data.flags = 0;
		ioctl(socket_id, **RTPRIV_IOCTL_MAC**, &wrq);

RTPRIV_IOCTL_E2P (Set EEPROM Value)	Iwpriv ra0 **e2p 40=1234**	sprintf(name, "ra0");
		strcpy(data, " **e2p 40=1234**");
		strcpy(wrq.ifr_name, name);
		wrq.u.data.length = strlen(data);
		wrq.u.data.pointer = data;
		wrq.u.data.flags = 0;
		ioctl(socket_id, **RTPRIV_IOCTL_E2P**, &wrq);

15.2.4. Get Data: BBP, MAC and EEPROM

Command and IOCTL Function		
Get Data: BBP, MAC and EEPROM , Parameters is Same as iwpriv		
Type	**Command**	**IOCTL**

RTPRIV_IOCTL_BBP
(Get BBP Register Value)
	Iwpriv ra0 **bbp 17**	sprintf(name, "ra0");
		strcpy(data, " **bbp 17**");
		strcpy(wrq.ifr_name, name);
		wrq.u.data.length = strlen(data);
		wrq.u.data.pointer = data;
		wrq.u.data.flags = 0;
		ioctl(socket_id, **RTPRIV_IOCTL_BBP**, &wrq);

RTPRIV_IOCTL_MAC
(Get MAC Register Value)
	Iwpriv ra0 **mac 3000**	sprintf(name, "ra0");
		strcpy(data, " **mac 3000**");
		strcpy(wrq.ifr_name, name);
		wrq.u.data.length = strlen(data);
		wrq.u.data.pointer = data;
		wrq.u.data.flags = 0;
		ioctl(socket_id, **RTPRIV_IOCTL_MAC**, &wrq);

RTPRIV_IOCTL_E2P
(Get EEPROM Value)
	Iwpriv ra0 **e2p 40**	sprintf(name, "ra0");
		strcpy(data, " **e2p 40**");
		strcpy(wrq.ifr_name, name);
		wrq.u.data.length = strlen(data);
		wrq.u.data.pointer = data;
		wrq.u.data.flags = 0;
		ioctl(socket_id, **RTPRIV_IOCTL_E2P**, &wrq);

15.2.5. Set Raw Data

IOCTL Function	
Set Raw Data by I/O Control Interface	
Function Type	**IOCTL**

RTPRIV_IOCTL_RADIUS_DATA	sprintf(name, "ra0");
	strcpy(wrq.ifr_name, name);
	memset(data, 0x55, 100);
	wrq.u.data.length = 100;
	wrq.u.data.pointer = data;
	wrq.u.data.flags = 0;
	ioctl(socket_id, RTPRIV_IOCTL_RADIUS_DATA, &wrq);

RTPRIV_IOCTL_ADD_WPA_KEY	NDIS_802_11_KEY *vp;
	
	sprintf(name, "ra0");
	strcpy(wrq.ifr_name, name);
	memset(data, 0, sizeof(NDIS_802_11_KEY));
	vp = (NDIS_802_11_KEY *)&data;
	vp->Length = sizeof(NDIS_802_11_KEY);
	memset(vp->addr, 0x11, 6);
	vp->KeyIndex = 2;
	vp->KeyLength = 32;
	memset(vp->KeyMaterial, 0xAA, 32);
	wrq.u.data.length = sizeof(NDIS_802_11_KEY);
	wrq.u.data.pointer = data;
	wrq.u.data.flags = 0;
	ioctl(socket_id, RTPRIV_IOCTL_ADD_WPA_KEY, &wrq);

RTPRIV_IOCTL_ADD_PMKID_CACHE	NDIS_802_11_KEY *vp;
	
	sprintf(name, "ra0");
	strcpy(wrq.ifr_name, name);
	memset(data, 0, sizeof(NDIS_802_11_KEY));
	vp = (NDIS_802_11_KEY *)&data;
	vp->Length = sizeof(NDIS_802_11_KEY);
	memset(vp->addr, 0x11, 6);
	vp->KeyIndex = 2;
	vp->KeyLength = 32;
	memset(vp->KeyMaterial, 0xBB, 32);
	wrq.u.data.length = sizeof(NDIS_802_11_KEY);
	wrq.u.data.pointer = data;
	wrq.u.data.flags = 0;
	ioctl(socket_id, RTPRIV_IOCTL_ADD_PMKID_CACHE, &wrq);

15.2.6. Set Raw Data with Flags

IOCTL Function	
Set Raw Data by I/O Control Interface with Flags	
Function Type	**IOCTL**

RT_SET_APD_PID	sprintf(name, "ra0");
	strcpy(wrq.ifr_name, name);
	memset(data, 0, 4);
	data[0] = 12;
	wrq.u.data.length = 4;
	wrq.u.data.pointer = data;
	wrq.u.data.flags = RT_SET_APD_PID;
	ioctl(socket_id, RT_PRIV_IOCTL, &wrq);

RT_SET_DEL_MAC_ENTRY	sprintf(name, "ra0");
	strcpy(wrq.ifr_name, name);
	memset(data, 0xdd, 6);
	strcpy(wrq.ifr_name, name);
	wrq.u.data.length = 6;
	wrq.u.data.pointer = data;
	wrq.u.data.flags = RT_SET_DEL_MAC_ENTRY;
	ioctl(socket_id, RT_PRIV_IOCTL, &wrq);

RT_OID_WSC_SET_SELECTED_REGISTRAR	sprintf(name, "ra0");
	strcpy(wrq.ifr_name, name);
	memset(data, decodeStr, decodeLen);
	strcpy(wrq.ifr_name, name);
	wrq.u.data.length = decodeLen;
	wrq.u.data.pointer = data;
	wrq.u.data.flags =
	RT_OID_WSC_SET_SELECTED_REGISTRAR;
	ioctl(socket_id, RT_PRIV_IOCTL, &wrq);

RT_OID_WSC_EAPMSG	sprintf(name, "ra0");
	strcpy(wrq.ifr_name, name);
	memset(data, wscU2KMsg, wscU2KMsgLen);
	strcpy(wrq.ifr_name, name);
	wrq.u.data.length = wscU2KMsgLen;
	wrq.u.data.pointer = data;
	wrq.u.data.flags = RT_OID_WSC_EAPMSG;
	ioctl(socket_id, RT_PRIV_IOCTL, &wrq);

15.2.7. Get Raw Data with Flags

IOCTL Function	
Get Raw Data by I/O Control Interface with Flags	
Function Type	**IOCTL**

RT_QUERY_ATE_TXDONE_COUNT	sprintf(name, "ra0");
	strcpy(wrq.ifr_name, name);
	memset(data, 0, 4);
	wrq.u.data.length = 4;
	wrq.u.data.pointer = data;
	wrq.u.data.flags = RT_QUERY_ATE_TXDONE_COUNT;
	ioctl(socket_id, RT_PRIV_IOCTL, &wrq);

RT_QUERY_SIGNAL_CONTEXT	RT_SIGNAL_STRUC *sp;
	
	sprintf(name, "ra0");
	strcpy(wrq.ifr_name, name);
	memset(data, 0, sizeof(RT_SIGNAL_STRUC));
	strcpy(wrq.ifr_name, name);
	wrq.u.data.length = sizeof(RT_SIGNAL_STRUC);
	wrq.u.data.pointer = data;
	wrq.u.data.flags = RT_QUERY_SIGNAL_CONTEXT;

	ioctl(socket_id, **RT_PRIV_IOCTL**, &wrq);

RT_OID_WSC_QUERY_STATUS	sprintf(name, "ra0");
	strcpy(wrq.ifr_name, name);
	memset(data, 0, sizeof(INT));
	strcpy(wrq.ifr_name, name);
	wrq.u.data.length = sizeof(INT);
	wrq.u.data.pointer = data;
	wrq.u.data.flags = **RT_OID_WSC_QUERY_STATUS**;
	ioctl(socket_id, **RT_PRIV_IOCTL**, &wrq);

RT_OID_WSC_PIN_CODE	sprintf(name, "ra0");
	strcpy(wrq.ifr_name, name);
	memset(data, 0, sizeof(ULONG));
	strcpy(wrq.ifr_name, name);
	wrq.u.data.length = sizeof(ULONG);
	wrq.u.data.pointer = data;
	wrq.u.data.flags = **RT_OID_WSC_PIN_CODE**;
	ioctl(socket_id, **RT_PRIV_IOCTL**, &wrq);

RT_OID_WSC_UUID	sprintf(name, "ra0");
	strcpy(wrq.ifr_name, name);
	memset(data, 0, sizeof(UCHAR));
	strcpy(wrq.ifr_name, name);
	wrq.u.data.length = sizeof(UCHAR);
	wrq.u.data.pointer = data;
	wrq.u.data.flags = **RT_OID_WSC_UUID**;
	ioctl(socket_id, **RT_PRIV_IOCTL**, &wrq);

RT_OID_WSC_MAC_ADDRESS	sprintf(name, "ra0");
	strcpy(wrq.ifr_name, name);
	memset(data, 0, MAC_ADDR_LEN);
	strcpy(wrq.ifr_name, name);
	wrq.u.data.length = MAC_ADDR_LEN;
	wrq.u.data.pointer = data;
	wrq.u.data.flags = **RT_OID_WSC_MAC_ADDRESS**;
	ioctl(socket_id, **RT_PRIV_IOCTL**, &wrq);

RT_OID_GET_PHY_MODE	sprintf(name, "ra0");
	strcpy(wrq.ifr_name, name);
	memset(data, 0, sizeof(ULONG));
	strcpy(wrq.ifr_name, name);
	wrq.u.data.length = sizeof(ULONG);
	wrq.u.data.pointer = data;
	wrq.u.data.flags = **RT_OID_GET_PHY_MODE**;
	ioctl(socket_id, **RT_PRIV_IOCTL**, &wrq);

RT_OID_GET_LLTD_ASSO_TANLE	sprintf(name, "ra0");
	strcpy(wrq.ifr_name, name);
	memset(data, 0,
	sizeof(RT_LLTD_ASSOICATION_TABLE));
	strcpy(wrq.ifr_name, name);
	wrq.u.data.length =
	sizeof(RT_LLTD_ASSOICATION_TABLE);
	wrq.u.data.pointer = data;
	wrq.u.data.flags = **RT_OID_GET_LLTD_ASSO_TANLE**;
	ioctl(socket_id, **RT_PRIV_IOCTL**, &wrq);

15.3. Sample User Space Application

```
//============================================================================
//
// rtuser:
//   1. User space application to demo how to use IOCTL function.
//   2. Most of the IOCTL function is defined as "CHAR" type and return with string message.
//   3. Use sscanf to get the raw data back from string message.
//   4. The command format "parameter=value" is same as iwpriv command format.
//   5. Remember to insert driver module and bring interface up prior execute rtuser.
//        change folder path to driver "Module"
//        dos2unix *          ; in case the files are modified from other OS environment
//        chmod 644 *
//        chmod 755 Configure
//        make config
//        make
//        insmod rt61ap.o
//        ifconfig ra0 up
//
// Refer linux/if.h to have
//        #define ifr_name    ifr_ifrn.ifrn_name            /* interface name   */
//
// Make:
//        cc -Wall -ortuser rtuser.c
//
// Run:
//        ./rtuser
//
//============================================================================

#include <stdio.h>
#include <string.h>
#include <sys/socket.h>
#include <sys/ioctl.h>
#include <unistd.h>                /* for close */
#include <linux/wireless.h>

//============================================================================

#if WIRELESS_EXT <= 11
#ifndef SIOCDEVPRIVATE
#define SIOCDEVPRIVATE              0x8BE0
#endif
#define SIOCIWFIRSTPRIV             SIOCDEVPRIVATE
#endif

//
//SET/GET CONVENTION :
// * ---------------------
// * Simplistic summary :
// * o even numbered ioctls are SET, restricted to root, and should not
// *     return arguments (get_args = 0).
// * o odd numbered ioctls are GET, authorised to anybody, and should
// *     not expect any arguments (set_args = 0).
//
#define RT_PRIV_IOCTL                   (SIOCIWFIRSTPRIV + 0x01)
#define RTPRIV_IOCTL_SET                (SIOCIWFIRSTPRIV + 0x02)
#define RTPRIV_IOCTL_BBP                (SIOCIWFIRSTPRIV + 0x03)
#define RTPRIV_IOCTL_MAC                (SIOCIWFIRSTPRIV + 0x05)
#define RTPRIV_IOCTL_E2P                (SIOCIWFIRSTPRIV + 0x07)
#define RTPRIV_IOCTL_STATISTICS         (SIOCIWFIRSTPRIV + 0x09)
#define RTPRIV_IOCTL_ADD_PMKID_CACHE    (SIOCIWFIRSTPRIV + 0x0A)
#define RTPRIV_IOCTL_RADIUS_DATA        (SIOCIWFIRSTPRIV + 0x0C)
#define RTPRIV_IOCTL_GSITESURVEY        (SIOCIWFIRSTPRIV + 0x0D)
#define RTPRIV_IOCTL_ADD_WPA_KEY        (SIOCIWFIRSTPRIV + 0x0E)
#define RTPRIV_IOCTL_GET_MAC_TABLE      (SIOCIWFIRSTPRIV + 0x0F)


#define OID_GET_SET_TOGGLE              0x8000

#define RT_QUERY_ATE_TXDONE_COUNT       0x0401
```

```
#define RT_QUERY_SIGNAL_CONTEXT         0x0402
#define RT_SET_APD_PID                  (OID_GET_SET_TOGGLE + 0x0405)
#define RT_SET_DEL_MAC_ENTRY            (OID_GET_SET_TOGGLE + 0x0406)

//----------------------------------------------------------

#ifndef   TRUE
#define   TRUE      1
#endif

#ifndef   FALSE
#define   FALSE     0
#endif

#define MAC_ADDR_LEN                    6
#define ETH_LENGTH_OF_ADDRESS           6
#define MAX_LEN_OF_MAC_TABLE            64

//----------------------------------------------------------

typedef struct _COUNTERS
{
     unsigned long  TxSuccessTotal;
     unsigned long  TxSuccessWithoutRetry;
     unsigned long  TxSuccessWithRetry;
     unsigned long  TxFailWithRetry;
     unsigned long  RtsSuccess;
     unsigned long  RtsFail;
     unsigned long  RxSuccess;
     unsigned long  RxWithCRC;
     unsigned long  RxDropNoBuffer;
     unsigned long  RxDuplicateFrame;
     unsigned long  FalseCCA;
     unsigned long  RssiA;
     unsigned long  RssiB;
}    COUNTERS;

//----------------------------------------------------------

typedef   struct _SITE_SURVEY
{
     unsigned char       channel;
     unsigned short      rssi;
     unsigned char       ssid[33];
     unsigned char       bssid[6];
     unsigned char       security[9];
}    SITE_SURVEY;

//----------------------------------------------------------

typedef struct _COUNTER_HOTSPOT
{
     // unsigned long     LinkUpTime;
     unsigned long       LastDataPacketTime;
     unsigned long       TotalTxByteCount;
     unsigned long       TotalRxByteCount;
} COUNTER_HOTSPOT, *PCOUNTER_HOTSPOT;

typedef struct _RT_802_11_MAC_ENTRY
{
     unsigned char       Addr[ETH_LENGTH_OF_ADDRESS];
     unsigned char       Aid;
     unsigned char       Psm;     // 0:PWR_ACTIVE, 1:PWR_SAVE
     COUNTER_HOTSPOT     HSCounter;
} RT_802_11_MAC_ENTRY, *PRT_802_11_MAC_ENTRY;

typedef struct _RT_802_11_MAC_TABLE
{
     unsigned long       Num;
     RT_802_11_MAC_ENTRY Entry[MAX_LEN_OF_MAC_TABLE];
} RT_802_11_MAC_TABLE, *PRT_802_11_MAC_TABLE;

// Key mapping keys require a BSSID
```

```
typedef struct _NDIS_802_11_KEY
{
     unsigned long       Length;           // Length of this structure
     unsigned char       addr[6];
     unsigned long       KeyIndex;
     unsigned long       KeyLength;        // length of key in bytes
     unsigned char       KeyMaterial[32];  // variable length depending on above field
} NDIS_802_11_KEY, *PNDIS_802_11_KEY;

typedef struct _RT_SIGNAL_STRUC {
     unsigned short      Sequence;
     unsigned char       MacAddr[MAC_ADDR_LEN];
     unsigned char       CurrAPAddr[MAC_ADDR_LEN];
     unsigned char       Sig;
} RT_SIGNAL_STRUC, *PRT_SIGNAL_STRUC;

//-------------------------------------------------------

COUNTERS       counter;
SITE_SURVEY    SiteSurvey[100];
char           data[4096];

//=========================================================================

int main( int argc, char ** argv )
{
     char      name[25];
     int       socket_id;
     struct    iwreq wrq;
     int       ret;

     // open socket based on address family: AF_NET ----------------------------
     socket_id = socket(AF_INET, SOCK_DGRAM, 0);
     if(socket_id < 0)
     {
          printf("\nrtuser::error::Open socket error!\n\n");
          return -1;
     }

     // set interface name as "ra0" ---------------------------------------
     sprintf(name, "ra0");
     memset(data, 0x00, 255);
//
//example of iwconfig ioctl function ========================================
//
     // get wireless name -------------------------------------------------
     strcpy(wrq.ifr_name, name);
     wrq.u.data.length = 255;
     wrq.u.data.pointer = data;
     wrq.u.data.flags = 0;
     ret = ioctl(socket_id, SIOCGIWNAME, &wrq);
     if(ret != 0)
     {
          printf("\nrtuser::error::get wireless name\n\n");
          goto rtuser_exit;
     }

     printf("\nrtuser[%s]:%s\n", name, wrq.u.name);
//
//example of iwpriv ioctl function ==========================================
//
     //WPAPSK, remove "set" string ---------------------------------------------
     memset(data, 0x00, 255);
     strcpy(data, "WPAPSK=11223344");
     strcpy(wrq.ifr_name, name);
     wrq.u.data.length = strlen(data)+1;
     wrq.u.data.pointer = data;
     wrq.u.data.flags = 0;
     ret = ioctl(socket_id, RTPRIV_IOCTL_SET, &wrq);
     if(ret != 0)
     {
          printf("\nrtuser::error::set wpapsk\n\n");
          goto rtuser_exit;
```

```
}
//set e2p, remove "e2p" string ------------------------------------------
memset(data, 0x00, 255);
strcpy(data, "80=1234");
strcpy(wrq.ifr_name, name);
wrq.u.data.length = strlen(data)+1;
wrq.u.data.pointer = data;
wrq.u.data.flags = 0;
ret = ioctl(socket_id, RTPRIV_IOCTL_E2P, &wrq);
if(ret != 0)
{
     printf("\nrtuser::error::set eeprom\n\n");
     goto rtuser_exit;
}

//printf("\n%s\n", wrq.u.data.pointer);
{
     int addr, value, p1;

     // string format: "\n[0x%02X]:0x%04X  " ==> "[0x20]:0x0C02"
     sscanf(wrq.u.data.pointer, "\n[%dx%02X]:%04X  ", &p1, &addr, &value);
     printf("\nSet EEP[0x%02X]:0x%04X\n", addr, value);
}

//get e2p, remove "e2p" string ------------------------------------------
memset(data, 0x00, 255);
strcpy(data, "80");
strcpy(wrq.ifr_name, name);
wrq.u.data.length = strlen(data)+1;
wrq.u.data.pointer = data;
wrq.u.data.flags = 0;
ret = ioctl(socket_id, RTPRIV_IOCTL_E2P, &wrq);
if(ret != 0)
{
     printf("\nrtuser::error::get eeprom\n\n");
     goto rtuser_exit;
}

//printf("\n%s\n", wrq.u.data.pointer);
{
     int addr, value, p1, p2;

     // string format: "\n[0x%02X]:0x%04X  " ==> "[0x20]:0x0C02"
     sscanf(wrq.u.data.pointer, "\n[%dx%04X]:%dx%X  ", &p1, &addr, &p2, &value);
     printf("\nGet EEP[0x%02X]:0x%04X\n", addr, value);
}

//set mac, remove "mac" string ------------------------------------------
memset(data, 0x00, 255);
strcpy(data, "2b4f=1");
strcpy(wrq.ifr_name, name);
wrq.u.data.length = strlen(data)+1;
wrq.u.data.pointer = data;
wrq.u.data.flags = 0;
ret = ioctl(socket_id, RTPRIV_IOCTL_MAC, &wrq);
if(ret != 0)
{
     printf("\nrtuser::error::set mac register\n\n");
     goto rtuser_exit;
}

//printf("\n%s\n", wrq.u.data.pointer);
{
     int addr, value, p1;

     // string format: "\n[0x%02X]:0x%04X  " ==> "[0x20]:0x0C02"
     sscanf(wrq.u.data.pointer, "\n[%dx%08X]:%08X  ", &p1, &addr, &value);
     printf("\nSet MAC[0x%08X]:0x%08X\n", addr, value);
}

//get mac, remove "mac" string ------------------------------------------
memset(data, 0x00, 255);
strcpy(data, "2b4f");
```

```
strcpy(wrq.ifr_name, name);
wrq.u.data.length = strlen(data)+1;
wrq.u.data.pointer = data;
wrq.u.data.flags = 0;
ret = ioctl(socket_id, RTPRIV_IOCTL_MAC, &wrq);
if(ret != 0)
{
     printf("\nrtuser::error::get mac register\n\n");
     goto rtuser_exit;
}

//printf("\n%s\n", wrq.u.data.pointer);
{
     int addr, value, p1;

     // string format: "\n[0x%02X]:0x%04X  " ==> "[0x20]:0x0C02"
     sscanf(wrq.u.data.pointer, "\n[%dx%08X]:%08X  ", &p1, &addr, &value);
     printf("\nGet MAC[0x%08X]:0x%08X\n", addr, value);
}

//set bbp, remove "bbp" string ------------------------------------------
memset(data, 0x00, 255);
strcpy(data, "17=32");
strcpy(wrq.ifr_name, name);
wrq.u.data.length = strlen(data)+1;
wrq.u.data.pointer = data;
wrq.u.data.flags = 0;
ret = ioctl(socket_id, RTPRIV_IOCTL_BBP, &wrq);
if(ret != 0)
{
     printf("\nrtuser::error::set bbp register\n\n");
     goto rtuser_exit;
}

//printf("\n%s\n", wrq.u.data.pointer);
{
     int id, addr, value, p1;

     // string format: "\n[0x%02X]:0x%04X  " ==> "[0x20]:0x0C02"
     sscanf(wrq.u.data.pointer, "\nR%02d[%dx%02X]:%02X\n", &id, &p1, &addr, &value);
     printf("\nSet BBP R%02d[0x%02X]:0x%02X\n", id, addr, value);
}

//get bbp, remove "bbp" string ------------------------------------------
memset(data, 0x00, 255);
strcpy(data, "17");
strcpy(wrq.ifr_name, name);
wrq.u.data.length = strlen(data)+1;
wrq.u.data.pointer = data;
wrq.u.data.flags = 0;
ret = ioctl(socket_id, RTPRIV_IOCTL_BBP, &wrq);
if(ret != 0)
{
     printf("\nrtuser::error::get bbp register\n\n");
     goto rtuser_exit;
}

//printf("\n%s\n", wrq.u.data.pointer);
{
     int id, addr, value, p1;

     // string format: "\n[0x%02X]:0x%04X  " ==> "[0x20]:0x0C02"
     sscanf(wrq.u.data.pointer, "\nR%02d[%dx%02X]:%02X  ", &id, &p1, &addr, &value);
     printf("\nGet BBP R%02d[0x%02X]:0x%02X\n", id, addr, value);
}

//get statistics, remove "stat" string ------------------------------------
memset(data, 0x00, 2048);
strcpy(data, "");
strcpy(wrq.ifr_name, name);
wrq.u.data.length = 0;
wrq.u.data.pointer = data;
wrq.u.data.flags = 0;
```

```
	ret = ioctl(socket_id, RTPRIV_IOCTL_STATISTICS, &wrq);
	if(ret != 0)
	{
		printf("\nrtuser::error::get statistics\n\n");
		goto rtuser_exit;
	}

	printf("\n========= Get AP Statistics ==========\n");
	{
		int i;
		char *sp = wrq.u.data.pointer;
		unsigned long *cp = (unsigned long *)&counter;

		for (i = 0 ; i < 13 ; i++)
		{
			sp = strstr(sp, "= ");
			sp = sp+2;
			sscanf(sp, "%ul", (unsigned int *)&cp[i]);
		}
	  printf("Tx success                        = %u\n", (unsigned int)counter.TxSuccessTotal);
	  printf("Tx success without retry          = %u\n", (unsigned int)
	                                                      counter.TxSuccessWithoutRetry);
	  printf("Tx success after retry            = %u\n", (unsigned int)counter.TxSuccessWithRetry);
	  printf("Tx fail to Rcv ACK after retry    = %u\n", (unsigned int)counter.TxFailWithRetry);
	  printf("RTS Success Rcv CTS               = %u\n", (unsigned int)counter.RtsSuccess);
	  printf("RTS Fail Rcv CTS                  = %u\n", (unsigned int)counter.RtsFail);
	  printf("Rx success                        = %u\n", (unsigned int)counter.RxSuccess);
	  printf("Rx with CRC                       = %u\n", (unsigned int)counter.RxWithCRC);
	  printf("Rx drop due to out of resource= %u\n", (unsigned int)counter.RxDropNoBuffer);
	  printf("Rx duplicate frame                = %u\n", (unsigned int)counter.RxDuplicateFrame);
	  printf("False CCA (one second)            = %u\n", (unsigned int)counter.FalseCCA);
	  printf("RSSI-A                            = %d\n", (  signed int)counter.RssiA);
	  printf("RSSI-B (if available)             = %d\n", (  signed int)counter.RssiB);
	}

#if 0
	//set AP to do site survey, remove "set" string ----------------------------
	memset(data, 0x00, 255);
	strcpy(data, "SiteSurvey=1");
	strcpy(wrq.ifr_name, name);
	wrq.u.data.length = strlen(data)+1;
	wrq.u.data.pointer = data;
	wrq.u.data.flags = 0;
	ret = ioctl(socket_id, RTPRIV_IOCTL_SET, &wrq);
#endif

	//get AP's site survey, remove "get_site_survey" string -------------------
	memset(data, 0x00, 2048);
	strcpy(data, "");
	strcpy(wrq.ifr_name, name);
	wrq.u.data.length = 4096;
	wrq.u.data.pointer = data;
	wrq.u.data.flags = 0;
	ret = ioctl(socket_id, RTPRIV_IOCTL_GSITESURVEY, &wrq);
	if(ret != 0)
	{
		printf("\nrtuser::error::get site survey\n\n");
		goto rtuser_exit;
	}

	//printf("\n%s\n", wrq.u.data.pointer);
	printf("\n========== Get Site Survey AP List ==========");
	if(wrq.u.data.length > 0)
	{
		int  i, apCount;
		char *sp, *op;
		int  len = wrq.u.data.length;

		op = sp = wrq.u.data.pointer;
		sp = sp+1+8+8+35+19+8+1;
		i = 0;
		// santy check
		//   1. valid char data
```

```
        //   2. rest length is larger than per line length ==> (1+8+8+35+19+8+1)
        while(*sp && ((len - (sp-op)) > (1+8+8+35+19+8)))
        {
            //if(*sp++ == '\n')
            //   continue;
            //printf("\n\nAP Count: %d\n", i);

            sscanf(sp, "%d", (int *)&SiteSurvey[i].channel);
            //printf("channel: %d\n", SiteSurvey[i].channel);

            sp = strstr(sp, "-");
            sscanf(sp, "-%d", (int *)&SiteSurvey[i].rssi);
            //printf("rssi: -%d\n", SiteSurvey[i].rssi);

            sp = sp+8;
            strncpy((char *)&SiteSurvey[i].ssid, sp, 32);
            SiteSurvey[i].ssid[32] = '\0';
            //printf("ssid: %s\n", SiteSurvey[i].ssid);

            sp = sp+35;
            sscanf(sp, "%02x:%02x:%02x:%02x:%02x:%02x",
                        (int *)&SiteSurvey[i].bssid[0], (int *)&SiteSurvey[i].bssid[1],
                        (int *)&SiteSurvey[i].bssid[2], (int *)&SiteSurvey[i].bssid[3],
                        (int *)&SiteSurvey[i].bssid[4], (int *)&SiteSurvey[i].bssid[5]);
            //printf("bssid: %02x:%02x:%02x:%02x:%02x:%02x\n",
            //        SiteSurvey[i].bssid[0], SiteSurvey[i].bssid[1],
            //        SiteSurvey[i].bssid[2], SiteSurvey[i].bssid[3],
            //        SiteSurvey[i].bssid[4], SiteSurvey[i].bssid[5]);

            sp = sp+19;
            strncpy((char *)&SiteSurvey[i].security, sp, 8);
            SiteSurvey[i].security[8] = '\0';
            //printf("security: %s\n", SiteSurvey[i].security);

            sp = sp+8+1;
            i = i+1;
        }

        apCount = i;
        printf("\n%-4s%-8s%-8s%-35s%-20s%-8s\n",
            "AP", "Channel", "RSSI", "SSID", "BSSID", "Security");
        for(i = 0 ; i < apCount ; i++)
        {//4+8+8+35+20+8
            printf("%-4d", i+1);
            printf("%-8d", SiteSurvey[i].channel);
            printf("-%-7d", SiteSurvey[i].rssi);
            printf("%-35s", SiteSurvey[i].ssid);
            printf("%02X:%02X:%02X:%02X:%02X:%02X   ",
                        SiteSurvey[i].bssid[0], SiteSurvey[i].bssid[1],
                        SiteSurvey[i].bssid[2], SiteSurvey[i].bssid[3],
                        SiteSurvey[i].bssid[4], SiteSurvey[i].bssid[5]);
            printf("%-8s\n", SiteSurvey[i].security);
        }
    }

    //get AP's mac table, remove "get_mac_table" string ----------------------
    memset(data, 0x00, 2048);
    strcpy(data, "");
    strcpy(wrq.ifr_name, name);
    wrq.u.data.length = 2048;
    wrq.u.data.pointer = data;
    wrq.u.data.flags = 0;
    ret = ioctl(socket_id, RTPRIV_IOCTL_GET_MAC_TABLE, &wrq);
    if(ret != 0)
    {
        printf("\nrtuser::error::get mac table\n\n");
        goto rtuser_exit;
    }

    printf("\n========== Get Associated MAC Table ==========");
    {
        RT_802_11_MAC_TABLE      *mp;
        int       i;
```

```
            mp = (RT_802_11_MAC_TABLE *)wrq.u.data.pointer;
            printf("\n%-4s%-20s%-4s%-10s%-10s%-10s\n",
                 "AID", "MAC_Address", "PSM", "LastTime", "RxByte", "TxByte");

            for(i = 0 ; i < mp->Num ; i++)
            {
                 printf("%-4d", mp->Entry[i].Aid);
                 printf("%02X:%02X:%02X:%02X:%02X:%02X   ",
                           mp->Entry[i].Addr[0], mp->Entry[i].Addr[1],
                           mp->Entry[i].Addr[2], mp->Entry[i].Addr[3],
                           mp->Entry[i].Addr[4], mp->Entry[i].Addr[5]);
                 printf("%-4d", mp->Entry[i].Psm);
                 printf("%-10u", (unsigned int)mp->Entry[i].HSCounter.LastDataPacketTime);
                 printf("%-10u", (unsigned int)mp->Entry[i].HSCounter.TotalRxByteCount);
                 printf("%-10u", (unsigned int)mp->Entry[i].HSCounter.TotalTxByteCount);
                 printf("\n");
            }
            printf("\n");
       }

//set: raw data
//   RTPRIV_IOCTL_RADIUS_DATA
//   RTPRIV_IOCTL_ADD_WPA_KEY
//   RTPRIV_IOCTL_ADD_PMKID_CACHE

       //set RADIUS Data ------------------------------------------------------
       printf("\nrtuser::set radius data\n\n");
       memset(data, 0x55, 100);
       strcpy(wrq.ifr_name, name);
       wrq.u.data.length = 100;
       wrq.u.data.pointer = data;
       wrq.u.data.flags = 0;
       ret = ioctl(socket_id, RTPRIV_IOCTL_RADIUS_DATA, &wrq);
       if(ret != 0)
       {
            printf("\nrtuser::error::set radius data\n\n");
            goto rtuser_exit;
       }

       //add WPA Key ----------------------------------------------------------
       printf("\nrtuser::add wpa key\n\n");
       {
            NDIS_802_11_KEY         *vp;

            memset(data, 0, sizeof(NDIS_802_11_KEY));
            vp = (NDIS_802_11_KEY *)&data;

            vp->Length = sizeof(NDIS_802_11_KEY);
            memset(vp->addr, 0x11, 6);
            vp->KeyIndex = 2;
            vp->KeyLength = 32;
            memset(vp->KeyMaterial, 0xAA, 32);

            strcpy(wrq.ifr_name, name);
            wrq.u.data.length = sizeof(NDIS_802_11_KEY);
            wrq.u.data.pointer = data;
            wrq.u.data.flags = 0;
            ret = ioctl(socket_id, RTPRIV_IOCTL_ADD_WPA_KEY, &wrq);
            if(ret != 0)
            {
                 printf("\nrtuser::error::add wpa key\n\n");
                 goto rtuser_exit;
            }
       }

       //add PMKID_CACHE ------------------------------------------------------
       printf("\nrtuser::add PMKID_CACHE\n\n");
       {
            NDIS_802_11_KEY          *vp;

            memset(data, 0, sizeof(NDIS_802_11_KEY));
            vp = (NDIS_802_11_KEY *)&data;
```

```
            vp->Length = sizeof(NDIS_802_11_KEY);
            memset(vp->addr, 0x11, 6);
            vp->KeyIndex = 2;
            vp->KeyLength = 32;
            memset(vp->KeyMaterial, 0xBB, 32);

            strcpy(wrq.ifr_name, name);
            wrq.u.data.length = sizeof(NDIS_802_11_KEY);
            wrq.u.data.pointer = data;
            wrq.u.data.flags = 0;
            ret = ioctl(socket_id, RTPRIV_IOCTL_ADD_PMKID_CACHE, &wrq);
            if(ret != 0)
            {
                 printf("\nrtuser::error::add PMKID_CACHE\n\n");
                 goto rtuser_exit;
            }
       }

//set: raw data
//    RT_SET_APD_PID
//    RT_SET_DEL_MAC_ENTRY

       //set APD_PID ------------------------------------------------------------
       printf("\nrtuser::set APD_PID\n\n");
       memset(data, 0, 4);
       data[0] = 12;
       strcpy(wrq.ifr_name, name);
       wrq.u.data.length = 4;
       wrq.u.data.pointer = data;
       wrq.u.data.flags = RT_SET_APD_PID;
       ret = ioctl(socket_id, RT_PRIV_IOCTL, &wrq);
       if(ret != 0)
       {
            printf("\nrtuser::error::set APD_PID\n\n");
            goto rtuser_exit;
       }

       //set DEL_MAC_ENTRY ------------------------------------------------------
       printf("\nrtuser::set DEL_MAC_ENTRY\n\n");
       memset(data, 0xdd, 6);
       strcpy(wrq.ifr_name, name);
       wrq.u.data.length = 6;
       wrq.u.data.pointer = data;
       wrq.u.data.flags = RT_SET_DEL_MAC_ENTRY;
       ret = ioctl(socket_id, RT_PRIV_IOCTL, &wrq);
       if(ret != 0)
       {
            printf("\nrtuser::error::set DEL_MAC_ENTRY\n\n");
            goto rtuser_exit;
       }

//get: raw data
//    RT_QUERY_ATE_TXDONE_COUNT
//    RT_QUERY_SIGNAL_CONTEXT

       //get ATE_TXDONE_COUNT ---------------------------------------------------
       printf("\nrtuser::get ATE_TXDONE_COUNT\n\n");
       memset(data, 0, 4);
       strcpy(wrq.ifr_name, name);
       wrq.u.data.length = 4;
       wrq.u.data.pointer = data;
       wrq.u.data.flags = RT_QUERY_ATE_TXDONE_COUNT;
       ret = ioctl(socket_id, RT_PRIV_IOCTL, &wrq);
       if(ret != 0)
       {
            printf("\nrtuser::error::get ATE_TXDONE_COUNT\n\n");
            goto rtuser_exit;
       }
       printf("\nATE_TXDONE_COUNT:: %08lx\n\n", (unsigned long)*wrq.u.data.pointer);

       //get SIGNAL_CONTEXT -----------------------------------------------------
       printf("\nrtuser::get SIGNAL_CONTEXT\n\n");
```

```
    {
        RT_SIGNAL_STRUC          *sp;

        memset(data, 0, sizeof(RT_SIGNAL_STRUC));
        strcpy(wrq.ifr_name, name);
        wrq.u.data.length = sizeof(RT_SIGNAL_STRUC);
        wrq.u.data.pointer = data;
        wrq.u.data.flags = RT_QUERY_SIGNAL_CONTEXT;
        ret = ioctl(socket_id, RT_PRIV_IOCTL, &wrq);
        if(ret != 0)
        {
            printf("\nrtuser::error::get SIGNAL_CONTEXT\n\n");
            goto rtuser_exit;
        }
        sp = (RT_SIGNAL_STRUC *)wrq.u.data.pointer;
        printf("\n===== SIGNAL_CONTEXT =====\n\n");
        printf("Sequence    = 0x%04x\n", sp->Sequence);
        printf("Mac.Addr    = %02x:%02x:%02x:%02x:%02x:%02x\n",
                        sp->MacAddr[0], sp->MacAddr[1],
                        sp->MacAddr[2], sp->MacAddr[3],
                        sp->MacAddr[4], sp->MacAddr[5]);
        printf("CurrAP.Addr = %02x:%02x:%02x:%02x:%02x:%02x\n",
                        sp->CurrAPAddr[0], sp->CurrAPAddr[1],
                        sp->CurrAPAddr[2], sp->CurrAPAddr[3],
                        sp->CurrAPAddr[4], sp->CurrAPAddr[5]);
        printf("Sig         = %d\n\n", sp->Sig);
    }

    //SSID, remove "set" string ------------------------------------------------
    memset(data, 0x00, 255);
    strcpy(data, "SSID=rtuser");
    strcpy(wrq.ifr_name, name);
    wrq.u.data.length = strlen(data)+1;
    wrq.u.data.pointer = data;
    wrq.u.data.flags = 0;
    ret = ioctl(socket_id, RTPRIV_IOCTL_SET, &wrq);
    if(ret != 0)
    {
        printf("\nrtuser::error::set SSID\n\n");
        goto rtuser_exit;
    }

rtuser_exit:
    if (socket_id >= 0)
        close(socket_id);

    if(ret)
        return ret;
    else
        return 0;
}
```

16. Porting Guide

This source code package can be use in Linux version after RedHat Linux 7.3

16.1. Source code package file list and description

assoc.c	# association service
auth.c	# authentication service
auth_rsp.c	# authentication response service
connect.c	# prepare/update beacon frame payload
mlme.c	# Mac sublayer management entity
mlme.h	# header file for Mac sublayer management entity
sanity.c	# payload sanity check
sync.c	# synchronization service
dls.c	# dynamic link setup (QoS)
eeprom.c	# E2PROM access method
md5.c	# Message Digest 5
md5.h	# header file for Message Digest 5
rtmp_data.c	# interrupt handler and Rx/Tx handler
rtmp_def.h	# constant definition
rtmp_info.c	# interface to user layer(include iwconfig/iwpriv)
rtmp_init.c	# module initialize
rtmp_main.c	# driver entry
rtmp_task.c	# tasklet
rtmp_tkip.c	# Temporal Key Integrity Protocol
rtmp_type.h	# data type definition
rtmp_wep.c	# Wire equivalent privacy
soft_ap.c	# soft ap functionality implementation
link_list.h	# link list used structure and inline routine
netif_block.c	# net interface block + link list
netif_block.h	# net interface block + link list
rtl865x_fast.c	# fast path used routine
rtl865x_fast.h	# fast path used definition
rtmp_apcli.c	# AP client function
rtmp_apcli.h	# AP client definition
rtmp_apcli_iparp.c	# AP client support IP/ARP
rtmp_apcli_pppoe.c	# AP client support PPPoE
sta_assoc.c	# AP client - association
sta_auth.c	# AP client - authentication
sta_ctrl.c	# AP client state machine
sta_mlme.c	# AP client management entity
sta_mlme.h	# AP client management entity definition
sta_sync.c	# AP client sync service
rtmp_igmpSnoop.c	# IGMP snooping function
rtmp_igmpSnoop.h	# IGMP snooping definition
wpa.c	# Wi-Fi Protected Access
wpa.h	# header file for Wi-Fi Protected Access
wsc.c	# simple config main function
wsc.h	# simple config main header

wsc_tlv.c	# type length value function
wsc_tlv.h	# type length value header
sha2.c	# secure hash algorithm function
sha2.h	# secure hash algorithm header
hmac.c	# hash message authentication code function
hmac.h	# hash message authentication code header
dh_key.c	# dynamic hash function
dh_key.h	# dynamic hash header
evp_enc.c	# digital "envelopes" encode function
evp_enc.h	# digital "envelopes" encode header
load	# script to install module
unload	# script to uninstall module
bridge_setup	# script about bridge setup procedure
config.mk	# initial config file
Configure	# script for auto-probe kernel version
Makefile	# makefile for kernel v2.4.x
Makefile.6	# makefile for kernel v2.6.x
Makefile.OpenRG.IXP	# makefile for IXP4XX
Makefile.RDC	# makefile for RDC 3210 or 8610
Makefile.RTL865X	# makefile for Realtek 865X
Makefile.RTL865X_FASTPATH	# same as above with fast path
Makefile.RTMPEMBEDDED	# makefile for ARM based
Makefile.SnapGear.IXP	# makefile for Snapgear based IXP platform
countrycode_vs_channel.txt	# country code vs channel cross reference
RT61AP.dat	**# initial profile stored in /etc/Wireless/RT61AP/**
RT2561.bin	**# Firmware for RT2561, need locate in specific directory**
RT2561S.bin	**# Firmware for RT2561S, need locate in specific directory**
RT2661.bin	**# Firmware for RT2661, need locate in specific directory**

16.2. Compile Flags

Add compile flags(CFLAGS) in Makefile to support specific driver code

1. -DDBG # turn on driver debug message
2. -DRALINK_ATE # turn on ATE functionality
3. -DBIG_ENDIAN # turn on BigEndian platform's code
4. -DMBSS_SUPPORT # turn on multiple BSSID support
5. -DAGGREGATION_SUPPORT # turn on packet aggregation
6. -DWDS_SUPPORT # turn on WDS support
7. -DWMM_SUPPORT # turn on WMM support
8. -DAPCLI_SUPPORT # turn on ApClient support
9. -DIGMP_SNOOP_SUPPORT # turn on IGMP Snooping support
10. -DBLOCK_NET_IF # turn on Block Net Interface
11. -DCARRIER_DETECTION_SUPPORT # turn on carrier detectiob support
12. -DWSC_SUPPORT # turn on WSC support
13. -DLLTD_SUPPORT # turn on LLTD support
14. -DSNMP # turn on SNMP MIB support
15. -DWIFI_TEST # turn on Wi-Fi test support
16. -DWPA_SUPPLICANT_SUPPORT # turn on WPA_SUPPLICANT support
17. -DTHREAD_ISR # turn on TASKLET support
18. -DRTL865X_SOC # turn on RTL865x SoC support
19. -DNONCOPY_RX # turn on Rx non-Copy support

16.3. Porting Note List

1. In single processor system, macro like NdisAllocateSpinLock, NdisReleaseSpinLock and NdisAcquireSpinLock in rtmp.h can be re-implement as semaphore lock to improve proformance.
2. This module provide several interfaces for user layer process to communicate with module, like iwconfig/iwpriv or proprietary ioctl. You can remove interface-code you don't need to minimize code size.
3. In embedded system, it is prefered to modify the "NdisMoveMemory" routine in rtmp_init.c as kernel's memcpy routine to enhance performance.
4. When performance can not reach to reasonable value, tuning DRAM timing(clock) maybe have some effort.
5. Make sure RT61(RT2561.bin/ RT2561S.bin / RT2661.bin) firmware is copied to /etc/Wireless/RT61AP/
6. For embedded device application, add "PACKED" to data structure that is related to:
   a、Hardware – MAC: PCI device, Little-Endian, 32-bit alignment
   b、802.11 header – Little-Endian

16.4. RT61 Notes for Embedded Device Applications

1. PCI's byte order is Little-Endian.

2. 802.11's header is Little-Endian.

3. RT61 is PCI based device:
   a、Bus Master
   b、DMA Based
   c、Physical Memory Access
   d、Non-Cacheable(Data-Cache)
   e、Effect to Descriptor and Data Buffer

4. Hardware is referred to Fixed Offset, no padding and apply PACKED to
   a、Data Structure

b、802.11 Header

5. Spinlock_xxx:
   - **a**、spin_lock_irqsave(&flags)
   - **b**、spin_unlock_irqrestore(flags)

6. Big-Endian:
   - **a**、Bit is Reverse relative to Little-Endian
   - **b**、After data swap to fit data structure
   - **c**、If reference only, needn't to write back
   - **d**、If modified, need to write back

7. Security Setting:
   - **a**、1st: Set SSID
   - **b**、2nd: Set Pass-Parse
   - **c**、3rd: Set SSID to update capability information.

8. TxRate fixed at 11Mbps
   - **a**、Check assoc.c on build association connection, data rate is fixed on each associated station.
   - **b**、After data rate changed, station have to de-associate then re-associate to take in effect on rate change.

9. B/G Protection = ON:
   - **a**、Would trigger CTS-To-Self mechanism
   - **b**、Performance would downgrade around 25% to 33%
   - **c**、Check below factors:
     - **a、)** Slot time is short or long ?
     - **b、)** Short retry or long retry ?
     - **c、)** SIF time's setting ?

10. MCU not ready.
    Do delay loop to wait MCU ready.

11. Clear Beacon's Tx valid bit before setup Beacon frame on AP initial stage.

12. Default 8-bit to load firmware, depends on platform may change to 32-bit and/or have to do byte-swap.

17. Integration Guide for Intel IXP4XX Platform

17.1. Introduction

This document briefly describes the step-by-step procedure to integrate Ralink SoftAP driver with MontaVista Linux Professional Edition 3.1 and 4.0 (they would be abbreviated as MVL PE 3.1 and MVL PE 4.0 respectively later in this document) on Intel IXP4XX platforms. The readers may follow this document to generate the executable image files of Ralink SoftAP driver, Linux kernel, and get starting to perform the SoftAP functionality.

17.2. Prerequisites

The readers whose want to perform the integration instructions shall have an X86-based computer with RedHat Linux installed as the host development system. The experience on compiling the Linux kernel is nice to have.

17.3. Source Code Installation

It has been verified that the Ralink SoftAP driver can run together with the kernel of MVL PE 3.1 and 4.0 together on the Intel IXP4XX platforms. The procedures for building PE 3.1 and PE 4.0 are slightly different. Both procedures would be illustrated in this document.
Before the installation, it has to create a working directory for putting the source trees of MontaVista Linux and Ralink SoftAP driver. Use the commands shown in the following block to create a working directory:

```
$ cd ~
$ mkdir ./work
$ cd ./work
$
```

17.3.1. MontaVista Linux

The source tree of Linux kernel, the appropriate revision to Intel IXP4XX platform, is to be installed at this step:

- **For MVL PE 3.1**
  At first, the MontaVista Linux Professional Edition 3.1 General Availability release has to be installed. Then obtain development LSP which is appropriate to IXP4XX platform and install it.
  - ✓ For the Intel IXDP425 Big-Endian target platform, obtain the 0400824 or later Big-Endian LSP update from MontaVista. The 0400824 revision is dated August 30, 2004.
  - ✓ For the Intel IXDP465 Big-Endian target platform, obtain the 0401718 or later Big-Endian LSP from MontaVista. The 0401718 revision is dated January 27, 2005.

```
$ cp –a {path_to_lsp}/linux-2.4.20_mvl31 ./
```

- **For MVL PE 4.0**
  The MontaVista Linux Professional Edition 4.0 General Availability release shall be installed at first. Then Install the development platform LSP 0501140 revision that comes along with the MontaVista Linux Professional Edition 4.0 General Availability release CDs. This revision provides LSP for both Intel® IXDP425 Development Platform and Intel® IXDP465 Development Platform.

```
$ cp –a {path_to_lsp}/linux-2.6.10_dev ./
```

As the copying is done, a directory named *linux-2.4.20_mvl31* or *linux-2.6.10_dev* (according to which edition of MVL is installed) has located in the working directory. This directory is the root of source tree of the Linux kernel. It will be abbreviated to *<linux>* in the later examples.

17.3.2. Ralink SoftAP Driver

At first, the reader has to obtain the source code of Ralink SoftAP driver through contacting with Ralink Technology (http://www.ralinktech.com/). The driver version 1.1.1.0 is used as an example in the procedure throughout this document. The source code files are packed in a tgz file which is named *2007_0622_RT61_Linux_SoftAP_Drv1.1.1.0.tgz.* Use the commands shown in the following block to extract Ralink's source code.

```
$ tar xzvf {path_to_file}/2007_0622_RT61_Linux_SoftAP_Drv1.1.1.0.tgz
Archive:   2007_0622_RT61_Linux_SoftAP_Drv1.1.1.0.tgz
   inflating: 2007_0622_RT61-Linux-AP/assoc.c
   .. ..
   .. ..
$ ls
2007_0622_RT61_Linux_SoftAP_Drv1.1.1.0   linux-2.6.10_dev
$ cd 2007_0622_RT61_Linux_SoftAP_Drv1.1.1.0/
$ ls
8021X   Module
```

After the source tree (which named *2007_0622_RT61_Linux_SoftAP_Drv1.1.1.0* as shown above) has been extracted, two subdirectories are inside of it. One of the subdirectory named *Module*

contains the source code of Ralink SoftAP driver.

17.3.3. Source Tree Integration

17.3.3.1. Link Driver Directory to Linux Source Tree

The subdirectory *Module* shall be linked to a specific location where inside the Linux kernel source tree. Refer to the following command for making the linkage.

```
$ cd ~/work/<linux>/drivers/net/wireless
$ ln -s ~/work/2007_0622_RT61_Linux_SoftAP_Drv1.1.1.0/Module rt61ap
```

17.3.3.2. Modify the Linux Source Tree's Makefile

Modify (one of) the Makefile in the Linux source tree in order to include this module directory into the building components.

```
$ cd <linux>/drivers/net/wireless/
$ vi Makefile
```

- **For MVL PE 3.1**
  One text line shall be inserted to the thirty first row of the file. Refer to below block (the new line is in the bold type face of red color):

```
obj-$(CONFIG_HOSTAP_PLX)        += hostap_plx.o
obj-$(CONFIG_HOSTAP_PCI)        += hostap_pci.o

subdir-m += rt61ap/

include $(TOPDIR)/Rules.make
```

- **For MVL PE 4.0**
  One text line shall be inserted to end of the file. Refer to below block (the new line is in the bold type face of red color):

```
# 16-bit wireless PCMCIA client drivers
obj-$(CONFIG_PCMCIA_RAYCS)      += ray_cs.o
obj-$(CONFIG_PCMCIA_WL3501)     += wl3501_cs.o

obj-m += rt61ap/
```

17.3.3.3. Modify the Makefile in Ralink Driver Directory

Modify the Makefile in the Ralink driver directory in order to align the rules for being a subdirectory inside Linux source tree.

```
$ cd rt61ap/
$ vi Makefile
```

Please refer to below blocks for modifying the Makefile (the copyright banner in the file is skipped):

- **For MVL PE 3.1**

```
# Support AP client function
#HAS_APCLIENT=y

# Support IGMP snooping function
#HAS_IGMPSNOOPING=y

# Support Block Net-If during TxSw queue full.
#HAS_BLOCK_NET_IF=y

WFLAGS :=  -DMBSS_SUPPORT -DAGGREGATION_SUPPORT \
           -DWDS_SUPPORT -DWMM_SUPPORT

#WFLAGS += -DDBG
#WFLAGS += -DSNMP

ifeq ($(HAS_APCLIENT),y)
WFLAGS += -DAPCLI_SUPPORT
endif

ifeq ($(HAS_IGMPSNOOPING),y)
WFLAGS += -DIGMP_SNOOP_SUPPORT
endif

ifeq ($(HAS_BLOCK_NET_IF),y)
WFLAGS += -DBLOCK_NET_IF
endif

EXTRA_CFLAGS := $(WFLAGS) -DBIG_ENDIAN

OBJ = rt61ap.o
O_TARGET := $(OBJ)
obj-m     := $(OBJ)

obj-y :=  \
          rtmp_main.o     \
          rtmp_task.o     \
          mlme.o          \
          connect.o       \
          sync.o          \
          assoc.o         \
          auth.o          \
          auth_rsp.o      \
          rtmp_data.o     \
          rtmp_init.o     \
          sanity.o        \
          rtmp_wep.o      \
          rtmp_info.o     \
          eeprom.o        \
          rtmp_tkip.o     \
          wpa.o           \
          md5.o           \
          soft_ap.o       \
          dls.o

ifeq ($(HAS_APCLIENT),y)
obj-y +=  sta_auth.o      \
          sta_assoc.o     \
          sta_sync.o      \
```

```
          sta_ctrl.o      \
          rtmp_apcli.o    \
          rtmp_apcli_iparp.o    \
          rtmp_apcli_pppoe.o    \
          sta_mlme.o
endif

ifeq ($(HAS_IGMPSNOOPING),y)
obj-y += rtmp_igmpSnoop.o
endif

ifeq ($(HAS_BLOCK_NET_IF),y)
obj-y += netif_block.o
endif

include $(TOPDIR)/Rules.make
```

- **For MVL PE 4.0**

```
# Support AP client function
#HAS_APCLIENT=y

# Support IGMP snooping function
#HAS_IGMPSNOOPING=y

# Support Block Net-If during TxSw queue full.
#HAS_BLOCK_NET_IF=y

WFLAGS :=  -DMBSS_SUPPORT -DAGGREGATION_SUPPORT \
           -DWDS_SUPPORT -DWMM_SUPPORT -DRALINK_ATE

#WFLAGS += -DDBG
#WFLAGS += -DSNMP

ifeq ($(HAS_APCLIENT),y)
WFLAGS += -DAPCLI_SUPPORT
endif

ifeq ($(HAS_IGMPSNOOPING),y)
WFLAGS += -DIGMP_SNOOP_SUPPORT
endif

ifeq ($(HAS_BLOCK_NET_IF),y)
WFLAGS += -DBLOCK_NET_IF
endif

EXTRA_CFLAGS := $(WFLAGS) -DBIG_ENDIAN

OBJ = rt61ap.o

obj-m := $(OBJ)

rt61ap-objs := \
               rtmp_main.o   \
               rtmp_task.o   \
               mlme.o        \
               connect.o     \
```

```
                sync.o          \
                assoc.o         \
                auth.o          \
                auth_rsp.o      \
                rtmp_data.o     \
                rtmp_init.o     \
                sanity.o        \
                rtmp_wep.o      \
                rtmp_info.o     \
                eeprom.o        \
                rtmp_tkip.o     \
                wpa.o           \
                md5.o           \
                soft_ap.o       \
                dls.o

ifeq ($(HAS_APCLIENT),y)
rt61ap-objs += \
                sta_auth.o      \
                sta_assoc.o     \
                sta_sync.o      \
                sta_ctrl.o      \
                rtmp_apcli.o    \
                rtmp_apcli_iparp.o  \
                rtmp_apcli_pppoe.o  \
                sta_mlme.o
endif

ifeq ($(HAS_IGMPSNOOPING),y)
rt61ap-objs += rtmp_igmpSnoop.o
endif

ifeq ($(HAS_BLOCK_NET_IF),y)
rt61ap-objs += netif_block.o
endif
```

17.4. Building the Target Files

In the first step, add the path to the tool chain (which for MVL PE 3.1 or 4.0) to the environment variable "PATH".

• For Bourne-compatible shells:
PATH=/opt/montavista/pro/devkit/arm/xscale_be/bin:$PATH

• For C-Shell compatible shells:
setenv PATH /opt/montavista/pro/devkit/arm/xscale_be/bin:$PATH

17.4.1. Building a Bootable Kernel Image

- **For MVL PE 3.1**
  The commands are executed in the top-level Linux source tree.

  $ cd <linux>

  If using the IXDP425 with a Big-Endian LSP:

```
$ make ixdp425_config

If using the IXDP465 with a Big-Endian LSP:
$ make ixdp465_config

Then All platforms follow the remaining steps:
$ make oldconfig
$ make dep
$ make zImage
```

This last step creates a file named *<linux>/arch/arm/boot/zImage*, and can be booted using the RedBoot bootloader. Copy zImage to the tftpboot directory, typically */tftpboot*.

- **For MVL PE 4.0**
  The commands are executed in the top-level Linux source tree.

```
$ cd <linux>
$ make prepare-all
(You may be prompted to select specific kernel options. Press [Enter] to accept the
default values for any prompts. For additional clarification of kernel configuration
options, contact MontaVista)

$ make zImage
```

This last step creates a file named *<linux>/arch/arm/boot/zImage*, and can be booted using the RedBoot bootloader. Copy zImage to the tftpboot directory, typically */tftpboot*.

17.4.2. Building the Kernel Module for Ralink SoftAP Driver

The command is for building all kernel modules, including the one for Ralink SoftAP driver. It is also executed in the top-level Linux source tree:

```
$ make modules
```

This command is for both MVL PE 3.1 and PE 4.0. As this command is done, the kernel module of Ralink SoftAP driver is generated in the following path:

- **For MVL PE 3.1**
  <linux>/drivers/net/wireless/rt61ap/rt61ap.o
- **For MVL PE 4.0**
  <linux>/drivers/net/wireless/rt61ap/rt61ap.ko

Copy the *rt61ap.o* or *rt61ap.ko* file into the embedded target file system on the host machine which typical is, for example: *<TARGET_FILE_SYSTEM>/lib/modules*

Note:
For MVL PE 3.1 and 4.0 for a Big-Endian Intel IXP4XX platform, the default target path (TARGET_FILE_SYSTEM) is */opt/montavista/pro/devkit/arm/xscale_be/target*

Also copy the three files, *RT2561.bin*, *RT2561S.bin*, and *RT61AP.dat*, from *<linux>/drivers/net/wireless/rt61ap/* into *<TARGET_FILE_SYSTEM>/etc/Wireless/RT61AP*

```
$ mkdir –p <TARGET_FILE_SYSTEM>/etc/Wireless/RT61AP
```

```
$ cd <linux>/drivers/net/wireless/rt61ap
$ cp RT2561.bin <TARGET_FILE_SYSTEM>/etc/Wireless/RT61AP/
$ cp RT2561S.bin <TARGET_FILE_SYSTEM>/etc/Wireless/RT61AP/
$ cp RT61AP.dat <TARGET_FILE_SYSTEM>/etc/Wireless/RT61AP/
```

The user can change the configuration for the wireless AP by editing the *RT61AP.dat* file. Refer to Ralink's Release Note for more information.

17.5. Loading and Using Ralink SoftAP Driver on MVL

This procedure assumes that the target platform has RedBoot pre-programmed into flash. Versions prior to v1.94 do not support Little-Endian target configurations. For more information on how to program the flash images or how to configure RedBoot to connect to the host platform, refer to the appropriate IXP400 software RedBoot software release notes (The readers can get the document from http://www.intel.com/design/network/products/npfamily/docs/ixp4xx.htm)

1. It is recommended that the MAC addresses for the target platform be set at this point. This process is documented in the appropriate IXP400 software RedBoot software release notes.
2. Load and then execute the kernel using the RedBoot commands:

```
Redboot> load -v –r –b %{FREEMEMLO} zImage
Redboot> exec
```

After the zImage above is booted on the target platform, follow the procedure shown below to load the kernel modules for the Ralink SoftAP driver.

- **For MVL PE 3.1**

```
192.168.0.100 login: root
root@192.168.0.100:~# cd /lib/modules
root@192.168.0.100:~# insmod rt61ap.o
root@192.168.0.100:~# ifconfig ra0 up
```

- **For MVL PE 4.0**

```
192.168.0.100 login: root
root@192.168.0.100:~# cd /lib/modules
root@192.168.0.100:~# insmod rt61ap.ko
root@192.168.0.100:~# ifconfig ra0 up
```

The final command activates the soft AP's functionality. The user can then perform the wireless communication via the wireless AP.

18. Make Files

18.1. Configure

```
#! /bin/bash
#
# Configure
#
# ************************************************************
# * Ralink Tech Inc.
# * 4F, No. 2 Technology 5th Rd.
# * Science-based Industrial Park
# * Hsin-chu, Taiwan, R.O.C.
# *
# * (c) Copyright 2005, Ralink Technology, Inc.
# *
# * All rights reserved. Ralink's source code is an unpublished work and the
# * use of a copyright notice does not imply otherwise. This source code
# * contains confidential trade secret material of Ralink Tech. Any attemp
# * or participation in deciphering, decoding, reverse engineering or in any
# * way altering the source code is stricitly prohibited, unless the prior
# * written consent of Ralink Technology, Inc. is obtained.
#

ECHO="/bin/echo -e "
fail ()
{
	$ECHO ""
	$ECHO "Configuration failed"
	$ECHO ""
	exit 1
}


PROMPT=y

#============================================

CONFIG=config.new
CONFIG_MK=config.mk
rm -f $CONFIG $CONFIG_MK $MODVER
cat << 'EOF' > $CONFIG
#
# Automatically generated by 'make config' -- don't edit!
#
EOF

write_str () {
	value=`eval $ECHO '$'$1`
	$ECHO "$1"=\"$value\" >> $CONFIG
	$ECHO "$1=$value" >> $CONFIG_MK
}


prompt () {
	eval $3=\"$2\"
	if [ "$PROMPT" = "y" ] ; then
	 $ECHO "$1 [$2]: \c"
	 read tmp
	 if [ -n "$tmp" ] ; then eval $3=\"$tmp\" ; fi
	else
	 $ECHO "$1 [$2]"
	fi
}

ask_str () {
	default=`eval $ECHO '$'$2`
	prompt "$1" "`$ECHO $default`" answer
	eval $2=\"$answer\"
	write_str $2
```

```
}


$ECHO ""
$ECHO ""
$ECHO "---------- Ralink RT61 SoftAP Configuration ---------- "
$ECHO ""

CUR_RELEASE=`uname -r`
LINUX_SRC=/usr/src/linux-$CUR_RELEASE

if [ ! -d $LINUX_SRC ] ; then
	ask_str "    Linux kernel source directory" LINUX_SRC
	$ECHO     " "
	if [ ! -d $LINUX_SRC ] ; then
		$ECHO     "Linux source tree '$LINUX_SRC' is incomplete or missing!"
		fail
	fi
fi

$ECHO     "    Linux kernel source directory : $LINUX_SRC"
$ECHO     " "
write_str LINUX_SRC


# What kernel are we compiling for?
version () {
	$ECHO ""
	expr $1 \* 65536 + $2 \* 256 + $3
}

for TAG in VERSION PATCHLEVEL SUBLEVEL EXTRAVERSION ; do
	eval `sed -ne "/^$TAG/s/[    ]//gp" $LINUX_SRC/Makefile`
done

VERSION_CODE=`version $VERSION $PATCHLEVEL $SUBLEVEL`
if [ $VERSION_CODE -lt `version 2 2 0` ] ; then
	$ECHO "This package requires at least a 2.2.x series kernel."
	fail
fi
```

18.2. config.mk

```
LINUX_SRC=/usr/src/linux-2.4.18-3
TARGET_MODDIR=/lib/modules/2.4.18-3
```

18.3. Makefile

18.3.1. Makefile for Little-Endian

18.3.1.1. Makefile for x86 Little-Endian

```
#########################################################################
# Ralink Technology, Inc.                                               #
# 4F, No. 2 Technology 5th Rd.                                          #
# Science-based Industrial Park                                         #
# Hsin-chu, Taiwan, R.O.C.                                              #
#                                                                       #
# (c) Copyright 2005, Ralink Technology, Inc.                           #
#                                                                       #
# All rights reserved. Ralink's source code is an unpublished work and the  #
# use of a copyright notice does not imply otherwise. This source code  #
# contains confidential trade secret material of Ralink Tech. Any attempt   #
# or participation in deciphering, decoding, reverse engineering or in any  #
# way altering the source code is stricitly prohibited, unless the prior    #
# written consent of Ralink Technology, Inc. is obtained.               #
#########################################################################

CC := cc

include ./config.mk

WFLAGS := -Wall -Wstrict-prototypes -Wno-trigraphs
CFLAGS := -D__KERNEL__                                  \
          -I$(LINUX_SRC)/include                        \
          -O2                                           \
          -fomit-frame-pointer                          \
          -fno-strict-aliasing                          \
          -fno-common                                   \
          -pipe                                         \
          -mpreferred-stack-boundary=2                  \
          -march=i686                                   \
          -DMODULE                                      \
          -DMODVERSIONS                                 \
          -include $(LINUX_SRC)/include/linux/modversions.h \
          $(WFLAGS)                                     \
          -DDBG

OBJ = rt61ap.o

all: $(OBJ)
     @touch config.mk

rt61.o:                         \
          rtmp_main.o           \
          mlme.o                \
          connect.o             \
          sync.o                \
          assoc.o               \
          auth.o                \
          auth_rsp.o            \
          rtmp_data.o           \
          rtmp_init.o           \
          sanity.o              \
          rtmp_wep.o            \
          rtmp_info.o           \
          eeprom.o              \
          rtmp_tkip.o           \
          wpa.o                 \
          md5.o                 \
          soft_ap.o
     $(LD) -r $^ -o $@

clean:
     rm -f *.o *~ core

config:
```

```
@touch config.mk
@./Configure
```

18.3.1.2. Makefile.RTMPEMBEDDED for Little-Endian

```
#############################################################################
 # Ralink Technology, Inc.                                                   #
 # 4F, No. 2 Technology 5th Rd.                                              #
 # Science-based Industrial Park                                             #
 # Hsin-chu, Taiwan, R.O.C.                                                  #
 #                                                                           #
 # (c) Copyright 2005, Ralink Technology, Inc.                               #
 #                                                                           #
 # All rights reserved. Ralink's source code is an unpublished work and the  #
 # use of a copyright notice does not imply otherwise. This source code      #
 # contains confidential trade secret material of Ralink Tech. Any attempt   #
 # or participation in deciphering, decoding, reverse engineering or in any  #
 # way altering the source code is stricitly prohibited, unless the prior    #
 # written consent of Ralink Technology, Inc. is obtained.                   #
#############################################################################

export RALINK_CFLAGS

RALINK_WFLAGS := -Wno-unused -O2
RALINK_CFLAGS :=    -fomit-frame-pointer            \
                    -DRTMP_EMBEDDED                 \
                    -DMBSS_SUPPORT                  \
                    -DAGGREGATION_SUPPORT           \
                    -DWDS                           \
                    -DWMM_SUPPORT                   \
                    -DLinux
RALINK_CFLAGS += -mstructure-size-boundary=8

EXTRA_CFLAGS += $(RALINK_WFLAGS) $(RALINK_CFLAGS)

O_TARGET := rt61ap.o

obj-y +=                        \
            rtmp_main.o         \
            mlme.o              \
            connect.o           \
            sync.o              \
            assoc.o             \
            auth.o              \
            auth_rsp.o          \
            rtmp_data.o         \
            rtmp_init.o         \
            sanity.o            \
            rtmp_wep.o          \
            rtmp_info.o         \
            eeprom.o            \
            rtmp_tkip.o         \
            wpa.o               \
            md5.o               \
            soft_ap.o

include $(TOPDIR)/Rules.make
```

18.3.1.3. Makefile.RDC for Little-Endian

```
#############################################################################
 # Ralink Technology, Inc.                                                 #
 # 4F, No. 2 Technology 5th Rd.                                            #
 # Science-based Industrial Park                                           #
 # Hsin-chu, Taiwan, R.O.C.                                                #
 #                                                                         #
 # (c) Copyright 2005, Ralink Technology, Inc.                             #
 #                                                                         #
 # All rights reserved. Ralink's source code is an unpublished work and the #
 # use of a copyright notice does not imply otherwise. This source code    #
 # contains confidential trade secret material of Ralink Tech. Any attempt #
 # or participation in deciphering, decoding, reverse engineering or in any #
 # way altering the source code is stricitly prohibited, unless the prior  #
 # written consent of Ralink Technology, Inc. is obtained.                 #
#############################################################################

export RALINK_CFLAGS

RALINK_WFLAGS :=                        \
                -Wno-unused             \
                -O2                     \
                -DMBSS_SUPPORT          \
                -DAGGREGATION_SUPPORT   \
                -DWDS_SUPPORT           \
                -DWMM_SUPPORT

RALINK_CFLAGS :=                        \
                -fomit-frame-pointer    \
                -DRTMP_EMBEDDED         \
                -DRALINK_ATE -DLinux

EXTRA_CFLAGS += $(RALINK_WFLAGS) $(RALINK_CFLAGS)

O_TARGET := rt61ap.o

obj-y +=                        \
        rtmp_main.o             \
        mlme.o                  \
        connect.o               \
        sync.o                  \
        assoc.o                 \
        auth.o                  \
        auth_rsp.o              \
        rtmp_data.o             \
        rtmp_init.o             \
        sanity.o                \
        rtmp_wep.o              \
        rtmp_info.o             \
        eeprom.o                \
        rtmp_tkip.o             \
        wpa.o                   \
        md5.o                   \
        soft_ap.o

include $(TOPDIR)/Rules.make
```

18.3.1.4. Makefile.S3C2510 for Little-Endian

```
#############################################################################
 # Ralink Technology, Inc.                                                 #
 # 4F, No. 2 Technology 5th Rd.                                            #
 # Science-based Industrial Park                                           #
 # Hsin-chu, Taiwan, R.O.C.                                                #
 #                                                                         #
 # (c) Copyright 2005, Ralink Technology, Inc.                             #
 #                                                                         #
 # All rights reserved. Ralink's source code is an unpublished work and the #
 # use of a copyright notice does not imply otherwise. This source code    #
 # contains confidential trade secret material of Ralink Tech. Any attempt #
 # or participation in deciphering, decoding, reverse engineering or in any #
 # way altering the source code is stricitly prohibited, unless the prior  #
 # written consent of Ralink Technology, Inc. is obtained.                 #
#############################################################################

export RALINK_CFLAGS

RALINK_WFLAGS :=                        \
                -Wno-unused             \
                -O2                     \
                -DDBG
RALINK_CFLAGS :=                        \
                -fomit-frame-pointer    \
                -DRTMP_EMBEDDED         \
                -DMBSS_SUPPORT          \
                -DAGGREGATION_SUPPORT   \
                -DWDS_SUPPORT           \
                -DWMM_SUPPORT           \
                -DLinux

RALINK_CFLAGS += -mstructure-size-boundary=8

EXTRA_CFLAGS += $(RALINK_WFLAGS) $(RALINK_CFLAGS)

O_TARGET := rt61ap.o

obj-y +=                        \
        rtmp_main.o             \
        mlme.o                  \
        connect.o               \
        sync.o                  \
        assoc.o                 \
        auth.o                  \
        auth_rsp.o              \
        rtmp_data.o             \
        rtmp_init.o             \
        sanity.o                \
        rtmp_wep.o              \
        rtmp_info.o             \
        eeprom.o                \
        rtmp_tkip.o             \
        wpa.o                   \
        md5.o                   \
        soft_ap.o

include $(TOPDIR)/Rules.make
```

18.3.1.5. Makefile.Micrel for Little-Endian

```
#####################################################################
 # Ralink Technology, Inc.                                           #
 # 4F, No. 2 Technology 5th Rd.                                      #
 # Science-based Industrial Park                                     #
 # Hsin-chu, Taiwan, R.O.C.                                          #
 #                                                                   #
 # (c) Copyright 2005, Ralink Technology, Inc.                       #
 #                                                                   #
 # All rights reserved. Ralink's source code is an unpublished work and the   #
 # use of a copyright notice does not imply otherwise. This source code      #
 # contains confidential trade secret material of Ralink Tech. Any attempt   #
 # or participation in deciphering, decoding, reverse engineering or in any  #
 # way altering the source code is stricitly prohibited, unless the prior    #
 # written consent of Ralink Technology, Inc. is obtained.                   #
#####################################################################

MK=arm-config.mk
include $(MK)
CC=$(CROSS)gcc
LD=$(CROSS)ld

WFLAGS := -Wall -Wstrict-prototypes -Wno-trigraphs -Wno-unused
CFLAGS := -D__KERNEL__ -I$(LINUX_SRC)/include -O2 -fomit-frame-pointer -fno-strict-aliasing -fno-common
CFLAGS += -pipe -DMODULE -DMODVERSIONS -include $(LINUX_SRC)/include/linux/modversions.h
CFLAGS += $(WFLAGS)
CFLAGS += -DKS8695P_ARM -DDBG

OBJ = rt61ap.o

all: $(OBJ)
	@touch arm-config.mk

rt61ap.o: rtmp_main.o		\
		mlme.o		\
		connect.o	\
		sync.o		\
		assoc.o		\
		auth.o		\
		auth_rsp.o	\
		rtmp_data.o	\
		rtmp_init.o	\
		sanity.o	\
		rtmp_wep.o	\
		rtmp_info.o	\
		eeprom.o	\
		rtmp_tkip.o	\
		wpa.o		\
		md5.o		\
		ks8695p_funcs.o	\
		ks8695p_ioctl.o	\
		soft_ap.o

	$(LD) -r $^ -o $@

clean:
	rm -f *.o *~ core tags .*.swp

install: $(OBJ)
	@touch $(MK)
	@echo $(TARGET_MODDIR)
	install -m 644 $(OBJ) $(TARGET_MODDIR)

config:
	@touch $(MK)
	@./Configure.arm
```

18.3.1.5.1. arm-config.mk

```
LINUX_SRC=/home/kendin/linux
LINUX_TOOLS=/usr/local/arm
MODDIR=/home/kendin/Soho/ramdisk/lib/modules/net
CROSS=/usr/local/arm/bin/arm-linux-
TARGET_MODDIR=/home/kendin/Soho/ramdisk/lib/modules/net

# LINUX_SRC=../../../linux
# MODDIR=../../ramdisk/lib/modules/net
# CROSS=arm-linux-
# TARGET_MODDIR=../../ramdisk/lib/modules/net
```

18.3.1.5.2. Configure.arm

```
#! /bin/bash
#
# Configure.arm
# Modified for Micrel's KS8695P
#
# ****************************************************************************
# * Ralink Tech Inc.
# * 4F, No. 2 Technology 5th Rd.
# * Science-based Industrial Park
# * Hsin-chu, Taiwan, R.O.C.
# *
# * (c) Copyright 2002, Ralink Technology, Inc.
# *
# * All rights reserved. Ralink's source code is an unpublished work and the
# * use of a copyright notice does not imply otherwise. This source code
# * contains confidential trade secret material of Ralink Tech. Any attemp
# * or participation in deciphering, decoding, reverse engineering or in any
# * way altering the source code is stricitly prohibited, unless the prior
# * written consent of Ralink Technology, Inc. is obtained.
#

ECHO="/bin/echo -e "

fail ()
{
        $ECHO ""
        $ECHO "Configuration failed"
        $ECHO ""
        exit 1
}

PROMPT=y

#============================================================================
CONFIG=arm-config.new
CONFIG_MK=arm-config.mk
rm -f $CONFIG $CONFIG_MK $MODVER
cat << 'EOF' > $CONFIG
#
# Automatically generated by 'make config' -- don't edit!
#
EOF

write_str () {
        value=`eval $ECHO '$'$1`
        $ECHO "$1"=\"$value\" >> $CONFIG
        $ECHO "$1=$value" >> $CONFIG_MK
}

prompt () {
        eval $3=\"$2\"
        if [ "$PROMPT" = "y" ] ; then
         $ECHO "$1 [$2]: \c"
         read tmp
         if [ -n "$tmp" ] ; then eval $3=\"$tmp\" ; fi
        else
         $ECHO "$1 [$2]"
```

```
        fi
}

ask_str () {
        default=`eval $ECHO '$'$2`
        prompt "$1" "`$ECHO $default`" answer
        eval $2=\"$answer\"
        write_str $2
}

$ECHO ""
$ECHO "------ Ralink RT61 Configuration for KS8695P ------"
$ECHO ""

CUR_RELEASE=2.4.16-rmk2
LINUX_SRC=/home/Kendin/linux
LINUX_TOOLS=/home/Kendin/tools
MODDIR=/home/Kendin/ramdisk/lib/modules/net

# What kernel are we compiling for?
version () {
        $ECHO ""
        expr $1 \* 65536 + $2 \* 256 + $3
}

for TAG in VERSION PATCHLEVEL SUBLEVEL EXTRAVERSION ; do
        eval `sed -ne "/^$TAG/s/[   ]//gp" $LINUX_SRC/Makefile`
done

VERSION_CODE=`version $VERSION $PATCHLEVEL $SUBLEVEL`
if [ $VERSION_CODE -lt `version 2 2 0` ] ; then
        $ECHO "This package requires at least a 2.2.x series kernel."
        fail
fi

ask_str "    Linux Kernel Source path"    LINUX_SRC
ask_str "    Toolchain path" LINUX_TOOLS
ask_str "    Module install directory" MODDIR

CROSS=$LINUX_TOOLS/bin/arm-linux-
echo "CROSS=$CROSS" >> $CONFIG_MK

TARGET_MODDIR=$MODDIR
write_str TARGET_MODDIR

$ECHO ""
$ECHO "    ----- Configure settings -----"
cat $CONFIG_MK
```

18.3.2. Makefile for Big-Endian

18.3.2.1. Makefile for Big-Endian Generic

```
#############################################################################
 # Ralink Technology, Inc.                                                   #
 # 4F, No. 2 Technology 5th Rd.                                              #
 # Science-based Industrial Park                                             #
 # Hsin-chu, Taiwan, R.O.C.                                                  #
 #                                                                           #
 # (c) Copyright 2005, Ralink Technology, Inc.                               #
 #                                                                           #
 # All rights reserved. Ralink's source code is an unpublished work and the  #
 # use of a copyright notice does not imply otherwise. This source code      #
 # contains confidential trade secret material of Ralink Tech. Any attempt   #
 # or participation in deciphering, decoding, reverse engineering or in any  #
 # way altering the source code is stricitly prohibited, unless the prior    #
 # written consent of Ralink Technology, Inc. is obtained.                   #
#############################################################################

export RALINK_CFLAGS

RALINK_WFLAGS := -Wno-unused -O2
RALINK_CFLAGS :=    -fomit-frame-pointer            \
                    -DBIG_ENDIAN                    \
                    -DMBSS_SUPPORT                  \
                    -DAGGREGATION_SUPPORT           \
                    -DWDS_SUPPORT                   \
                    -DWMM_SUPPORT                   \
                    -DLinux
RALINK_CFLAGS += -mstructure-size-boundary=8

EXTRA_CFLAGS += $(RALINK_WFLAGS) $(RALINK_CFLAGS)

O_TARGET := rt61ap.o

obj-y +=                    \
        rtmp_main.o         \
        mlme.o              \
        connect.o           \
        sync.o              \
        assoc.o             \
        auth.o              \
        auth_rsp.o          \
        rtmp_data.o         \
        rtmp_init.o         \
        sanity.o            \
        rtmp_wep.o          \
        rtmp_info.o         \
        eeprom.o            \
        rtmp_tkip.o         \
        wpa.o               \
        md5.o               \
        soft_ap.o

include $(TOPDIR)/Rules.make
```

18.3.2.2. Makefile.OpenRG.IXP for Big-Endian

```
#####################################################################
 # Ralink Technology, Inc.                                          #
 # 4F, No. 2 Technology 5th Rd.                                     #
 # Science-based Industrial Park                                    #
 # Hsin-chu, Taiwan, R.O.C.                                         #
 #                                                                  #
 # (c) Copyright 2005, Ralink Technology, Inc.                      #
 #                                                                  #
 # All rights reserved. Ralink's source code is an unpublished work and the   #
 # use of a copyright notice does not imply otherwise. This source code      #
 # contains confidential trade secret material of Ralink Tech. Any attempt   #
 # or participation in deciphering, decoding, reverse engineering or in any  #
 # way altering the source code is stricitly prohibited, unless the prior    #
 # written consent of Ralink Technology, Inc. is obtained.                   #
#####################################################################

# Comment/uncomment the following line to enable/disable debugging
RGSRC = ../../../
include $(RGSRC)/envir.mak

RA_WFLAG=       -DMBSS_SUPPORT              \
                -DAGGREGATION_SUPPORT       \
                -DWDS_SUPPORT               \
                -DWMM_SUPPORT

MOD_CFLAGS= -O2                             \
                -I.                         \
                $(RA_WFLAG)                 \
                -DBIG_ENDIAN                \
                -DLinux                     \
                -DDBG                       \
                -DRALINK_ATE

MOD_TARGET = rt61ap.o

O_OBJS_$(MOD_TARGET)=           \
                rtmp_main.o     \
                mlme.o          \
                connect.o       \
                sync.o          \
                assoc.o         \
                auth.o          \
                auth_rsp.o      \
                rtmp_data.o     \
                rtmp_init.o     \
                sanity.o        \
                rtmp_wep.o      \
                rtmp_info.o     \
                eeprom.o        \
                soft_ap.o       \
                md5.o           \
                wpa.o           \
                rtmp_tkip.o


all: $(MOD_TARGET)
#Put the 8051 bin file to the ramdisk
RAMDISK_ETC_FILES=RT61AP.dat RT2561.bin RT2561S.bin RT2661.bin

include $(RGMK)
```

18.3.2.3. Makefile.SnapGear.IXP for Big-Endian

```
#####################################################################
 # Ralink Technology, Inc.                                          #
 # 4F, No. 2 Technology 5th Rd.                                     #
 # Science-based Industrial Park                                    #
 # Hsin-chu, Taiwan, R.O.C.                                         #
 #                                                                  #
 # (c) Copyright 2005, Ralink Technology, Inc.                      #
 #                                                                  #
 # All rights reserved. Ralink's source code is an unpublished work and the   #
 # use of a copyright notice does not imply otherwise. This source code      #
 # contains confidential trade secret material of Ralink Tech. Any attempt   #
 # or participation in deciphering, decoding, reverse engineering or in any  #
 # way altering the source code is stricitly prohibited, unless the prior    #
 # written consent of Ralink Technology, Inc. is obtained.                   #
#####################################################################

CC:=arm-linux-gcc
LD:=arm-linux-ld
include ./config.mk
LDFLAGS :=-EB
WFLAGS :=                                       \
                -DBIG_ENDIAN                    \
                -DRTMP_EMBEDDED                 \
                -DMBSS_SUPPORT                  \
                -DAGGREGATION_SUPPORT           \
                -DWDS_SUPPORT                   \
                -DWMM_SUPPORT                   \
                -Wall                           \
                -Wstrict-prototypes             \
                -Wno-trigraphs
CFLAGS :=                                       \
                -mbig-endian                    \
                -D__KERNEL__                    \
                -I$(LINUX_SRC)/include          \
                -O2                             \
                -fomit-frame-pointer            \
                -fno-strict-aliasing            \
                -Uarm                           \
                -fno-common                     \
                -pipe                           \
                -mapcs-32                       \
                -D__LINUX_ARM_ARCH__=5          \
                -mcpu=xscale                    \
                -mtune=xscale                   \
                -DMODULE                        \
                -DMODVERSIONS                   \
                -include $(LINUX_SRC)/include/linux/modversions.h $(WFLAGS)

OBJ = rt61ap.o

all: $(OBJ)
        @touch config.mk

rt61ap.o: rtmp_main.o           \
          mlme.o                \
          connect.o             \
          sync.o                \
          assoc.o               \
          auth.o                \
          auth_rsp.o            \
          rtmp_data.o           \
          rtmp_init.o           \
          sanity.o              \
          rtmp_wep.o            \
          rtmp_info.o           \
          eeprom.o              \
          rtmp_tkip.o           \
          wpa.o                 \
          md5.o                 \
          soft_ap.o
```

```
	$(LD) -r $(LDFLAGS) $^ -o $@

clean:
	rm -f *.o *~ core

config:
	@touch config.mk
	@./Configure
```

18.3.2.4. Makefile.RTL865X for Big-Endian

```
#####################################################################
 # Ralink Technology, Inc.                                          #
 # 4F, No. 2 Technology 5th Rd.                                     #
 # Science-based Industrial Park                                    #
 # Hsin-chu, Taiwan, R.O.C.                                         #
 #                                                                  #
 # (c) Copyright 2005, Ralink Technology, Inc.                      #
 #                                                                  #
 # All rights reserved. Ralink's source code is an unpublished work and the  #
 # use of a copyright notice does not imply otherwise. This source code      #
 # contains confidential trade secret material of Ralink Tech. Any attempt   #
 # or participation in deciphering, decoding, reverse engineering or in any  #
 # way altering the source code is stricitly prohibited, unless the prior    #
 # written consent of Ralink Technology, Inc. is obtained.                   #
#####################################################################

CFLAGS =        -Wall                                   \
                -O2                                     \
                -D__KERNEL__                            \
                -DMODULE                                \
                -D__MIPSEB__                            \
                -DBIG_ENDIAN                            \
                -DRTL865X_SOC                           \
                -DRALINK_ATE                            \
                -DMBSS_SUPPORT                          \
                -DAGGREGATION_SUPPORT                   \
                -DWDS_SUPPORT                           \
                -DWMM_SUPPORT                           \
                -fomit-frame-pointer                    \
                -fno-common                             \
                -mlong-calls                            \
                -mno-abicalls                           \
                -fno-pic                                \
                -pipe                                   \
                -I$(KERNELDIR)/include -I$(KERNELDIR)/drivers/net       \
                -include $(KERNELDIR)/include/linux/modversions.h

MOD_TARGET = rt61ap.o
CONF_FILE    = RT61AP.dat

OBJ = rt61ap.o

#all: $(MOD_TARGET)
all: $(OBJ)

rt61ap.o:                       \
                rtmp_data.o     \
                rtmp_main.o     \
                rtmp_init.o     \
                mlme.o          \
                soft_ap.o       \
                connect.o       \
                sync.o          \
                assoc.o         \
                auth.o          \
                auth_rsp.o      \
                sanity.o        \
                rtmp_wep.o      \
                rtmp_info.o     \
                eeprom.o        \
                rtmp_tkip.o     \
                wpa.o           \
                md5.o

        $(LD) -r $^ -o $@

#$(MOD_TARGET): $(OBJS)
#       $(LD) -r -o    $@ $(OBJS)

clean:
```

```
	rm -f *.o *.bak

romfs:
	cp -f $(MOD_TARGET) $(ROMFSDIR)/lib/modules/$(MOD_TARGET)
	cp -f $(CONF_FILE)    $(ROMFSDIR)/etc/$(CONF_FILE)
```

18.3.2.5. Makefile.BROADCOM for Big-Endian

```
#####################################################################
 # Ralink Technology, Inc.                                          #
 # 4F, No. 2 Technology 5th Rd.                                     #
 # Science-based Industrial Park                                    #
 # Hsin-chu, Taiwan, R.O.C.                                         #
 #                                                                  #
 # (c) Copyright 2005, Ralink Technology, Inc.                      #
 #                                                                  #
 # All rights reserved. Ralink's source code is an unpublished work and the   #
 # use of a copyright notice does not imply otherwise. This source code       #
 # contains confidential trade secret material of Ralink Tech. Any attempt    #
 # or participation in deciphering, decoding, reverse engineering or in any   #
 # way altering the source code is stricitly prohibited, unless the prior     #
 # written consent of Ralink Technology, Inc. is obtained.                    #
#####################################################################

CC = mips-uclibc-gcc
LD = mips-uclibc-ld

KERNELDIR = ./linux

CFLAGS =                                                \
        -Wall                                           \
        -O2                                             \
        -D__KERNEL__                                    \
        -DMODULE                                        \
        -D__MIPSEB__                                    \
        -DBIG_ENDIAN                                    \
        -DDBG                                           \
        -DRALINK_ATE                                    \
        -fomit-frame-pointer                            \
        -fno-common                                     \
        -mlong-calls                                    \
        -mno-abicalls                                   \
        -fno-pic                                        \
        -pipe                                           \
        -I$(KERNELDIR)/include                          \
        -I$(KERNELDIR)/drivers/net                      \
        -include $(KERNELDIR)/include/linux/modversions.h \
        -Wno-trigraphs                                  \
        -fno-strict-aliasing                            \
        -I$(KERNELDIR)/include/asm/gcc                  \
        -G 0                                            \
        -mno-abicalls                                   \
        -fno-pic                                        \
        -pipe                                           \
        -finline-limit=100000                           \
        -mabi=32                                        \
        -march=r4600                                    \
        -Wa,-32                                         \
        -Wa,-march=r4600                                \
        -Wa,-mips3                                      \
        -Wa,--trap                                      \
        -mlong-calls

MOD_TARGET = rt61ap.o
CONF_FILE    = RT61AP.dat

OBJ = rt61ap.o

all: $(OBJ)

rt61ap.o:                       \
        rtmp_data.o             \
        rtmp_main.o             \
        rtmp_init.o mlme.o      \
        soft_ap.o               \
        connect.o               \
        sync.o                  \
        assoc.o                 \
```

```
		auth.o			\
		auth_rsp.o		\
		sanity.o		\
		rtmp_wep.o		\
		rtmp_info.o		\
		eeprom.o		\
		rtmp_tkip.o		\
		wpa.o			\
		md5.o

	$(LD) -m elf32btsmip -r $^ -o $@

clean:
	rm -f *.o *.bak

romfs:
	cp -f $(MOD_TARGET) $(ROMFSDIR)/lib/modules/$(MOD_TARGET)
```

19. Miscellaneous

19.1. Multiple BSSID

1. Before turn on multiple BSSID, make sure the byte5 of MAC address in EEPROM is a multiple of 2 or 4 and reserve multiple MAC address when manufacturing. example, 00:0A:0B:0C:0D:04; 00:0A:0B:0C:0D:88

2. When enable multiple BSSID function, the field 'BssidNum' shall larger than 1 and less than 4

3. BssidNum can only be modified with editing configure file.

   When change the ' BssidNum ' field, the driver must restart, and modify bridge_setup file to group virtual interface.

   Others parameters can pass through iwpriv according to their interface.

4. The parameter that support **multiple BSSID** is listed as followed,
   - **1.)** SSID,
   - **2.)** AuthMode,
   - **3.)** EncrypType,
   - **4.)** WPAPSK,
   - **5.)** DefaultKeyID,
   - **6.)** Key1Type,
   - **7.)** Key1Str,
   - **8.)** Key2Type,
   - **9.)** Key2Str,
   - **10.)** Key3Type,
   - **11.)** Key3Str,
   - **12.)** Key4Type,
   - **13.)** Key4Str,
   - **14.)** AccessPolicy,
   - **15.)** AccessControlList,
   - **16.)** NoForwarding,
   - **17.)** IEEE8021X,
   - **18.)** TxRate,
   - **19.)** HideSSID,
   - **20.)** PreAuth,
   - **21.)** WmmCapable

   Others are not supported.

5. Example of notation to represent multiple ssid's parameter:
   - **1.)** BssidNum=4
   - **2.)** SSID=SSID-A;SSID-B;SSID-C;SSID-D
   - **3.)** AuthMode=OPEN;SHARED;WPAPSK;WPA
   - **4.)** EncrypType=NONE;WEP;TKIP;AES

6. MBSSID and WDS.
   There 64 security key table in MAC(RT61 - RT2561/RT2661).
   Entry 0: For Multicast and Broadcast.
   Entry 1 - 59: For Associated STA.
   Entry 60 - 63: For WDS.

   Current driver defined WDS number to 4.

19.2. Concurrent A+G with two devices

Below table is brief example for two interface.

For example, Linux HotPlug system found new device would create one driver instance(create new space for driver image) for new device to hold private informations(memory consumed).

RT61 Interface Bring Up Sequence						
NIC#	Sequence	Normal	WDS(Virtual)			
			1	2	3	4
Two	ifconfig ra0 up	ra0	ra2	ra3	ra4	ra5
	ifconfig ra1 up	ra1	ra6	ra7	ra8	ra9

NIC#	Sequence	Normal	MBSSID (Physical)			WDS(Virtual)			
						1	2	3	4
Two	ifconfig ra0 up	ra0	ra2	ra3	ra4	ra5	ra6	ra7	ra8
	ifconfig ra1 up	ra1	ra9	ra10	ra11	ra12	ra13	ra14	ra15

Note:
1. WDS is Virtual Interface, No IOCTL function.

19.3. Site Survey

1. Site survey issue "iwpriv ra0 set SiteSurvey=1"
2. After 4 seconds (wait site survey process complete) then issue "iwpriv ra0 get_site_survey" command to get data.
3. We can use system("iwpriv ra0 get_site_survey > /etc/site_survey.dat") then it will write the site survey data to /etc/site_survey.dat.

19.4. OLBC

DisableOLBC=1 ➔ Disable Co-Channel OLBC AP/STA Detection.
DisableOLBC=0 ➔ Enable Co-Channel OLBC AP/STA Detection.

Overlapping Legacy BSS Condition (OLBC)		
BGProtection	DisableOLBC	
	1 (Disable)	0 (Enable)
AUTO	Condition to Turn ON CTS-To-Self Protection	
	1. Only Associated 11B Client(STA).	1. Associated 11B Client(STA) 2. Co-Channel with 11B only mode a. Other 11B's AP b. 11B's STA that associated to Other 11B's AP
ON	CTS-To-Self Protection Always ON	CTS-To-Self Protection Always ON
OFF	No CTS-To-Self Protection	No CTS-To-Self Protection

Note:
1. BGProtection only has CTS-To-Self.
2. If the condition of RTS-CTS Threshold be triggerred then RTS-CTS Protection will

turn on, no matter what setting of BGProtection.

Example 1:

Assume:

a. RTS Threshold = 500 Bytes.
b. Length of Data Packet = 600 bytes

Result:

a. Packet#1 → RTS
b. Packet#2 ← CTS
c. Packet#3 → Data Packet#1 (500 Bytes)
d. Packet#4 ← Ack
e. Packet#5 → Data Packet#2 (100 Bytes)
f. Packet#6 ← Ack

Example 2:

Assume:

a. RTS Threshold = 500 Bytes.
b. Length of Data Packet = 490 bytes

Result:

a. Packet#1 → Data Packet#1 (490 Bytes)
b. Packet#2 ← Ack

3. For OLBC, please refer to section 2.21 of "WiFi-802_11g-TestPlan_V2_2.pdf".

19.5. Tx Power

RT61 Tx Power Cross Reference				
	EEPROM	**RF[R3]**	**BBP[R94]**	**Description**
1.	0xFA = -6	0x00 = 0	0x00	Below BBP Lower Bound Per step = 1 = 1dB
2.	0xFB = -5	0x00 = 0	0x01	
3.	0xFC = -4	0x00 = 0	0x02	
4.	0xFD = -3	0x00 = 0	0x03	
5.	0xFE = -2	0x00 = 0	0x04	
6.	0xFF = -1	0x00 = 0	0x05	
7.	0x00 = 0	0x00 = 0	0x06	In normal BBP range Per Step = 1 = 0.5dB
8.	0x01 = 1	0x01 = 1	0x06	
9.	0x02 = 2	0x02 = 2	0x06	
10.	0x03 = 3	0x03 = 3	0x06	
11.	0x04 = 4	0x04 = 4	0x06	
12.	0x05 = 5	0x05 = 5	0x06	
13.	0x06 = 6	0x06 = 6	0x06	
14.	0x07 = 6	0x07 = 6	0x06	
15.	0x08 = 8	0x08 = 8	0x06	
16.	0x09 = 9	0x09 = 9	0x06	
17.	0x0A = 10	0x0A = 10	0x06	
18.	0x0B = 11	0x0B = 11	0x06	
19.	0x0C = 12	0x0C = 12	0x06	
20.	0x0D = 13	0x0D = 13	0x06	
21.	0x0E = 14	0x0E = 14	0x06	
22.	0x0F = 15	0x0F = 15	0x06	
23.	0x10 = 16	0x10 = 16	0x06	
24.	0x11 = 17	0x11 = 17	0x06	
25.	0x12 = 18	0x12 = 18	0x06	
26.	0x13 = 19	0x13 = 19	0x06	
27.	0x14 = 20	0x14 = 20	0x06	
28.	0x15 = 21	0x15 = 21	0x06	
29.	0x16 = 22	0x16 = 22	0x06	
30.	0x17 = 23	0x17 = 23	0x06	
31.	0x18 = 24	0x18 = 24	0x06	
32.	0x19 = 25	0x19 = 25	0x06	
33.	0x1A = 26	0x1A = 26	0x06	
34.	0x1B = 27	0x1B = 27	0x06	
35.	0x1C = 28	0x1C = 28	0x06	
36.	0x1D = 29	0x1D = 29	0x06	
37.	0x1E = 30	0x1E = 30	0x06	
38.	0x1F = 31	0x1F = 31	0x06	
39.	0x20 = 32	0x1F = 31	0x07	Over BBP Upper Bound Per Step = 1 = 1dB
40.	0x21 = 33	0x1F = 31	0x08	
41.	0x22 = 34	0x1F = 31	0x09	
42.	0x23 = 35	0x1F = 31	0x0A	
43.	0x24 = 36	0x1F = 31	0x0B	

TxPower=value
parameter :: TxPower

Vaule
100 ~ 90 use value in E2PROM as default
90 ~ 60 default value -2

60 ~ 30 default value -6
30 ~ 15 default value -12
15 ~ 9 default value -18
9 ~ 0 default value -24

Note:
1. Range: 1 ~ 100 (unit in percentage)
2. This value restricted by HW characteristic.

TxPower			
	percentage		
1.	100 ~ 90	Default value from E2PROM	
2.	90 ~ 60	default value -2	-1dB
3.	60 ~ 30	default value -6	-3dB
4.	30 ~ 15	default value -12	-6dB
5.	15 ~ 9	default value -18	-9dB
6.	9 ~ 0	default value -24	-12dB

19.6. Auto Channel Selection

19.6.1. Rules

- RT61AP driver will traverse all supported channels when system bootup.
- Driver will stay 0.5 sec in each channel and collect necessary information - Max RSSI.
- Driver implements a dirty rate for each channel to qualify which channel is suitable for selecting.
- If the Max RSSI is not equal to zero, the channel's dirty rate will plus 10.
- The upper and the lower 4 channel's dirty rate will plus one.

Finally,

RULE 1. pick up a good channel that no one used (dirtyness=0)
RULE 2. if not available, then co-use a channel that's no interference (dirtyness=10)
RULE 3. if not available, then co-use a channel that has minimum interferenc (dirtyness=11,12)
RULE 4. still not available, pick up the first channel

When AP scan through each channel (stay 0.5 sec) upon bootup. It'll maintain a max_rx_rssi for each channel, which value is actually acquired from each correctly received BEACON frames.

max_rx_rssi[ch] is used ony when this AP can't find a 100% clean channel (no neighbor AP within 5 channel apart) and there're more than 1 equal-dirty channels to choose from. In this case, this AP would choose the channel with smallest max_rx_rssi[ch] because this means the neighbor AP is more far away than the one in other channel.

The fundamental problem is -

Auto Channel Selection function decide channel dirtyness solely base on correcty received 802.11 BEACONs. All other signal/frame are not used (or not able to use) as an indication.

19.6.2. Practice

1. In the shielding room, the client can see 4 out side APs with very low power level. Channel_2 -91dB, Channel_3 -92dB, Channel_4 -91dB, Channel_6 -91dB. Set the channel to Auto and power on 5 times, the RT61AP goes to CH 1,1,1,1,1.

   ➔ If there are several outside APs and the signal are too weak and are actually invisible (no CRC-ok BEACON seen) at least during the RT61AP power-on period (e.g. theRSSI is -91dB). Therefore all 11 channels(assume country region is FCC) are clean, thus RT61AP just pickup the first clean channel which is channel 1.

2. In the shielding room, set one AP to Channel_1, and power on RT61AP 5 times, it goes to Channel 6, 6, 6, 6, 6.

   ➔ Now channel 1 is occupied, so does channel 2,3,4,5 become a little dirty (to avoid interference from AP_Channel_1), channel 6 is chosen because it's the first clean channel.

3. As item 2, now add another AP to Channel_6, and power on RT61AP 5 times, it goes to Channel 11, 11, 11, 11, 11.

   ➔ Then channel 6 also occupied, and channel 2,3,4,5,7,8,9,10 all dirty. Channel 11 is a correct decision.

4. As item 3, now add another AP to Channel_11, and power on RT61AP 5 times, it goes to Channel 1, 6, 6, 6, 6.

➔ Now channel 11 is occupied, and no clean channel at all. RT61AP decide to co-channel with other AP, but prefer that co-channel AP to be as far away as possible so it may choose channel 1, 6, or 11 depending which co-channel AP has smallest RSSI.

➔ Since all devices stay in shielding room, the RSSI may be very close. This explains why RT61AP sometimes choose channel 1, sometimes choose channel 6. You can check the distance of each AP to confirm that AP_Channel_1 and AP_Channel_6 is about the same distance to RT61AP, while AP_Channel_11 is closer.

5. Add 16M(Tx+Rx) traffic to AP in Channel_6, and power on RT61AP 5 times, it goes to Channel 1, 6, 6, 1, 6.

➔ Since RT61AP only count max_rx_rssi[ch] from correctly received BEACON. The extra traffic load won't affect the election result. RT61AP still picks up eiher Channel 1 or Channel 6 depends on the max_rx_rssi.

Maybe this algorithm is not perfect. But think about that data traffic is bursty by nature. So put weighing on this 0.5sec bootup-time traffic doesn't mean that much. AP_Channel_1 and AP_Channel_11 still may generate heavy loading later on.

As for

a. Channel 2,3,4,5, will interfere both AP_Channel_1 and AP_Channel_6, and
b. Channel 7,8,9,10 will intefere both AP_Channel_6 and AP_Channel_11.

So why picking up channel 3 or 8 is not a good choice.

19.7. The Difference of WPA1 and WPA2

19.7.1. WPA1

19.7.1.1. Wi-Fi WPA

1. Refer to "Wi-Fi 802.11g Interoperability Test Plan Version 2.4, Page 7":
   "The WPA protocol is defined by Wi-Fi document 'WPA for 802.11 Specification – Version 2.0, April 29, 2003'. The WPA Specification captures those clauses of the IEEE 802.11i Draft 3.0 that define Wi-Fi Protected Access."

19.7.1.2. IEEE 802.11i/D3.0 WPA

1. Pairwise key would be installed after 4-way handshake.
2. Group key would be installed before 2-way handshake.
3. Refer to "P802.11i/D3.0, November 2002, Page 80, Section 8.4.5 MPDU filtering, Figure 45—Sequence of Filtering-related Events" for detail information.

19.7.1.3. WPA1 Practice

```
*RT61*<7>AUTH_RSP-Rcv AUTH seq#1,Alg=0,Status=0 from 00:0c:43:26:61:25 to IF(ra0)
*RT61*<7>MacTableInsertEntry -IF(ra0) allocate entry #1, Total= 1
*RT61*<7>AUTH_RSP - IF(0) Send AUTH response (SUCCESS)...
*RT61*<7>ASSOC - receive ASSOC request from 00:0c:43:26:61:25
*RT61*<7>AssignAid (AID=1)
*RT61*<7>BuildAssoc-IF(0):AuthMode=4,WepStatus=6,GroupWepStatus=6,WpaState=7,AGGRE=1,PiggyBack=1,AP
SD=0
*RT61*<7>LOG#6 00:0c:43:26:61:25 successfully associated
*RT61*<7>Init entry init retry timer
*RT61*<7>assign AID=1 to 00:0c:43:26:61:25,MaxSupportedRate=54Mbps,CurrTxRate=54Mbps
*RT61*<7>RSNIE_Len=0x16,pEntry->RSNIE_Len=22,pEntry->PrivacyFilter=1
*RT61*<7>ASSOC - Send ASSOC response (Status=0) from IF(ra0)...

WpaEAPOLStartAction ====>>
==>WPAStart4WayHS
STA from 00:0c:43:26:61:25
PMK = 99:61:62:c4-86:a8:8d:bf
pEntry->AuthMode == Ndis802_11AuthModeWPA/WPAPSK
WPA - RTMPToWirelessSta ====>> to IF(ra0)
<== WPAStart4WayHS:pEntry->WpaState=8, FrameLen=113
Receive EAPOL-Key frame, TYPE = 3, Length =0
WPAMsgTypeSubst (EAPType=3)
WpaEAPOLKeyAction ===>
PeerPairMsg2Action ===>
PTK-ed 32 1f e3 2a 6f c4 e9
ANonce1-d5 1c 3c 54 7b 91 cb fd
ANonce2-dc 39 f1 bc cc 2 5e 77
MIC VALID in Msg 2 of 4-way handshake!!
RSN_IE VALID in Msg 2 of 4-way handshake!!
RTMPToWirelessSta : ETHTYPE = 88 8e FrameLen = 137!
WPA - RTMPToWirelessSta ====>> to IF(ra0)
Send Msg3 and setup timeout timer
Receive EAPOL-Key frame, TYPE = 3, Length =0
WPAMsgTypeSubst (EAPType=3)
WpaEAPOLKeyAction ===>
WpaEAPOL Peer Pair Msg4 Action ===>
MIC valid in Msg 4 of 4-way handshake!!
WPA1(PairwiseKey) = 63:c5:5d:75-7e:8c:b6:08
WPA1(RxMic) = fc:7a:1c:5f-95:72:62:e2
WPA1(TxMic) = 83:35:1f:67-54:fe:a5:67
*RT61*<7>AsicAddPairwiseKeyEntry: #1 Alg=AES mac=00:0c:43:26:61:25 key=63-c5-5d-..
IF(ra0) WPA Group Key ID    = 1
c 37 cf 69 cd 7c 85 49
83 f9 e2 2c ad a8 cc e7
f0 7 d2 b9 62 9a bd 3e
e9 b5 c0 a2 1 f9 d6 17
*RT61*<7>AsicAddSharedKeyEntry(BssIndex=0): AES key #1
```

RT61<7> Key =0c:37:cf:69:cd:7c:85:49:83:f9:e2:2c:ad:a8:cc:e7:
RT61<7> Rx MIC Key = e9:b5:c0:a2:01:f9:d6:17:
RT61<7> Tx MIC Key = f0:07:d2:b9:62:9a:bd:3e:
<== IF(ra0) WPAHardTransmit - FrameLen = 137
WPA - RTMPToWirelessSta ====>> to IF(ra0)
IF(ra0) recv WpaEAPOL Peer PAIR Msg4 Action and send GROUP Msg1
Receive EAPOL-Key frame, TYPE = 3, Length =0
WPAMsgTypeSubst (EAPType=3)
WpaEAPOLKeyAction ===>
PeerGroupMsg2Action ===> from MAC(00:0c:43:26:61:25)
Replay Counter VALID in Msg 2 of GROUP 2-way handshake!!!
MIC Valid in Msg 2 of GROUP 2-way handshake.
===> AP SETKEYS DONE - (ra0) WPA1, AuthMode=4, WepStatus=6

19.7.2. WPA2

19.7.2.1. Wi-Fi WPA2

1. Wi-Fi 802.11 WPA2 Interoperability Test Plan Version 2.4.2, Page 7:
   "The WPA2 protocol is based upon the IEEE 802.11i specification."

19.7.2.2. IEEE 802.11i WPA

1. Group key would be installed after AP received message 2 before send message 3.
2. Pairwise key would be installed after AP received message 4.
3. Refer to "IEEE Std 802.11i-2004, Page 87, Section 8.5.3.3 4-Way Handshake Message 3" for detail information.

19.7.2.3. WPA2 Practice

```
*RT61*<7>ASSOC - receive DIS-ASSOC request from 00:0c:43:26:61:25
*RT61*<7>AUTH_RSP-Rcv AUTH seq#1,Alg=0,Status=0 from 00:0c:43:26:61:25 to IF(ra0)
*RT61*<7>MacTableInsertEntry -IF(ra0) allocate entry #1, Total= 1
*RT61*<7>AUTH_RSP - IF(0) Send AUTH response (SUCCESS)...
*RT61*<7>ASSOC - receive ASSOC request from 00:0c:43:26:61:25
*RT61*<7>AssignAid (AID=1)
*RT61*<7>BuildAssoc-IF(0):AuthMode=7,WepStatus=6,GroupWepStatus=6,WpaState=7,AGGRE=1,PiggyBack=1,AP
SD=0
*RT61*<7>LOG#8 00:0c:43:26:61:25 successfully associated
*RT61*<7>Init entry init retry timer
*RT61*<7>assign AID=1 to 00:0c:43:26:61:25,MaxSupportedRate=54Mbps,CurrTxRate=54Mbps
*RT61*<7>RSNIE_Len=0x14,pEntry->RSNIE_Len=20,pEntry->PrivacyFilter=1
*RT61*<7>ASSOC - Send ASSOC response (Status=0) from IF(ra0)...

WpaEAPOLStartAction ====>>
==>WPAStart4WayHS
STA from 00:0c:43:26:61:25
PMK = 99:61:62:c4-86:a8:8d:bf
pEntry->AuthMode == Ndis802_11AuthModeWPA2/WPA2PSK
WPA - RTMPToWirelessSta ====>> to IF(ra0)
<== WPAStart4WayHS:pEntry->WpaState=8, FrameLen=113
Receive EAPOL-Key frame, TYPE = 3, Length =0
WPAMsgTypeSubst (EAPType=3)
WpaEAPOLKeyAction ===>
PeerPairMsg2Action ===>
PTK-20 75 9f 5c 42 ac 7 cd
ANonce1-15 5c 19 72 8e 78 74 3
ANonce2-5a 7f c2 ef 86 c8 ee 6c
MIC VALID in Msg 2 of 4-way handshake!!
RSN_IE VALID in Msg 2 of 4-way handshake!!
WPA2 Group Key ID  = 1
G_Key :c 37 cf 69 cd 7c 85 49
       83 f9 e2 2c ad a8 cc e7
TX Mic:f0 7 d2 b9 62 9a bd 3e
RX Mic:e9 b5 c0 a2 1 f9 d6 17
*RT61*<7>AsicAddSharedKeyEntry(BssIndex=0): AES key #1
*RT61*<7>       Key =0c:37:cf:69:cd:7c:85:49:83:f9:e2:2c:ad:a8:cc:e7:
*RT61*<7>       Rx MIC Key = e9:b5:c0:a2:01:f9:d6:17:
*RT61*<7>       Tx MIC Key = f0:07:d2:b9:62:9a:bd:3e:
RTMPToWirelessSta : ETHTYPE = 88 8e FrameLen = 169!
WPA - RTMPToWirelessSta ====>> to IF(ra0)
Send Msg3 and setup timeout timer
Receive EAPOL-Key frame, TYPE = 3, Length =0
WPAMsgTypeSubst (EAPType=3)
WpaEAPOLKeyAction ===>
Wpa2PeerPairMsg4Action ===> from MAC:00:0c:43:26:61:25
Replay Counter VALID in Msg 4 of 4-way handshake!
MIC Valid in Msg 4 of 4-way handshake!!
*RT61*<7>AsicAddPairwiseKeyEntry: #1 Alg=AES mac=00:0c:43:26:61:25 key=df-53-f5-..
===> AP SETKEYS DONE (ra0) - WPA2, AuthMode=7, WepStatus=6
```

20. Q&A

1. Why **WPAPSK** can not work?
   Ans:
   a. Please make sure the parameter **"DefaultKeyID" is set to 2** in configuration file

2. How to switch driver to operate in A band?
   Ans:
   a. make sure RFIC support A band
   b. parameter "WirelessMode" is set to 3 to support A band
   c. Channel set to 36, 40…..

3. When I set channel as 1, but it will appear in channel 3. Why?
   Ans:
   a. Make sure the channel is match with CountryRegion or CountryRegionABand

4. How can I know the version of package
   Ans:
   a. can see the definition of DRIVER_VERSION in rt_config.h
   b. use command "iwpriv ra0 set DriverVersion=0", it will export to debug console

5. Linux SoftAP Driver does not support antenna diversity.
   If the setting in EEPROM turns on antenna diversity, you can set "TxAntenna" in config file as 1(Antenna A) or 2(Antenna B) to fix antenna.