

MESHCOMMANDS REFERENCE MANUAL For MIMO/AC

Updated 2018 to support abg/n/ac radios and Ventana/Cavium Boards.

Contents

| | |
|---|----|
| 1. Radio Level Information..... | 4 |
| 1.1 ath5k..... | 4 |
| 1.1.1 beacon..... | 4 |
| 1.1.2 antenna | 4 |
| 1.1.3 bwmode..... | 4 |
| 1.1.4 frameerrros..... | 5 |
| 1.1.5 misc..... | 5 |
| 1.1.6 queue | 5 |
| 1.1.7 registers..... | 6 |
| 1.1.8 debug | 6 |
| 1.2 ath10k | 8 |
| 1.2.1 fw_stats..... | 8 |
| 1.2.2 wmi_services | 9 |
| 1.3 Wireless Interfaces..... | 10 |
| 2 Transient Radio Commands | 10 |
| 2.3 Freq..... | 10 |
| 2.4 Txpower..... | 10 |
| 3 Node Level Information..... | 11 |
| 3.1 Sta-List | 11 |
| 3.2 Downlink Stat | 11 |
| 3.3 Vlan-Stat..... | 12 |
| 3.4 Adhoc..... | 13 |
| 3.5 Reboot | 13 |
| 3.6 Coord..... | 14 |
| 3.7 Speed..... | 14 |
| 3.8 Temp..... | 14 |
| 3.9 Voltage..... | 14 |
| 3.10 Kap..... | 14 |
| 3.10.1 Information format for Stationary nodes..... | 14 |

- 3.10.2 Information format for mobile nodes.....16
- 4 Mobile Node Configuration Options.....17
 - 4.1 Sliding Scan Window and "winning" Parent Nodes.....17
 - 4.2 Default Preference and Selection Criteria.....17
 - 4.2.1 Usage Consideration.....21
 - 4.1 Mobwin.....22
- 5 Miscellaneous Commands.....23
 - 5.1 Perf.....23
 - 5.1 Kick.....24
 - 5.2 fwupdate.....24
 - 5.3 gpsd.....24
 - 5.4 Meminfo.....24
 - 5.5 telnetd.....24
- 6 Alconfset and Alconfget Commands.....25
 - 6.1 Introduction.....25
 - 6.2 Usage.....25
 - 6.3 Supported Parameters.....25
 - 6.4 Detail use of Parameters.....25
- 7 Debug Commands for Mesh.....43
 - 7.1 Using log information.....43
 - 7.2 Printing important internal mesh data structures.....43
 - 7.3 Printing queue stats for mesh modules.....44
 - 7.4 Printing thread stats for mesh modules.....45
 - 7.5 Printing drop stats for mesh modules.....45
 - 7.6 Dumping all stats.....46
- 8 Procedure for Sysupgrade.....47

1. Radio Level Information

Radio level information is available for each radio card present in the unit. The information is available through the following command:

```
For ath5k chipset  
cat /sys/kernel/debug/ieee80211/phyX/ath5k/...
```

```
For ath10k chipset  
cat /sys/kernel/debug/ieee80211/phyX/ath10k/
```

1.1 ath5k

1.1.1 beacon

Beacon related all the information will be is shown here.

```
cat /sys/kernel/debug/ieee80211/phyX/ath5k/beacon
```

Example output of above command

```
AR5K_BEACON 0x00000019 intval: 25 TIM: 0x0  
AR5K_LAST_TSTP 0x59f9bfc7  
AR5K_BEACON_CNT 0x00000000
```

```
AR5K_TIMER0 (TBTT) 0x00003563 TU: 00003563  
AR5K_TIMER1 (DMA) 0x0001ab08 TU: 00003561  
AR5K_TIMER2 (SWBA) 0x00e1ab90 TU: 001c3572  
AR5K_TIMER3 (ATIM) 0x00003564 TU: 00003564  
TSF 0x0000000070d5766a TU: 001c355d
```

1.1.2 antenna

Antenna related information is provided here.

```
cat /sys/kernel/debug/ieee80211/phyX/ath5k/antenna
```

The information is formatted as shown below:

```
antenna mode 0  
default antenna 0  
tx antenna 0
```

```
ANTENNA RX TX  
[antenna 1]3640 2034  
[antenna 2]9 0  
[antenna 3]0 0  
[antenna 4]0 0  
[invalid] 0 0
```

```
AR5K_DEFAULT_ANTENNA 0x00000000  
AR5K_STA_ID1_DEFAULT_ANTENNA 0  
AR5K_STA_ID1_DESC_ANTENNA 0  
AR5K_STA_ID1_RTS_DEF_ANTENNA 0  
AR5K_STA_ID1_SELFGEN_DEF_ANT 0
```

```
AR5K_PHY_AGCCTL_OFDM_DIV_DIS 0  
AR5K_PHY_RESTART_DIV_GC 4  
AR5K_PHY_FAST_ANT_DIV_EN 1
```

```
AR5K_PHY_ANT_SWITCH_TABLE_00x061a6981  
AR5K_PHY_ANT_SWITCH_TABLE_10x05165942
```

1.1.3 bwmode

Band width related information is provided here. It will provide current operating BW information.

```
cat /sys/kernel/debug/ieee80211/phyX/ath5k/bwmode
```

The information is formatted as shown below:

```
5 10 [20] 40
```

1.1.4 frameerrros

frameerrros information will be provided here.

```
cat /sys/kernel/debug/ieee80211/phy0/ath5k/frameerrors
```

The information is formatted as shown below:

RX

```
-----  
CRC2812 (24%)  
PHY0 (0%)  
FIFO 0 (0%)  
decrypt 2 (0%)  
MIC0 (0%)  
process 0 (0%)  
jumbo 3 (0%)  
[RX all 11597]  
RX-all-bytes 1412050
```

TX

```
-----  
retry 0 (0%)  
FIFO 0 (0%)  
filter 0 (0%)
```

```
TX-all-bytes 435337
```

1.1.5 misc

BSSID mask, filter flags and current operating mode of interface information will be provided here.

```
cat /sys/kernel/debug/ieee80211/phyX/ath5k/misc
```

The information is formatted as shown below:

```
bssid-mask: ff:ff:ff:ff:ff:f3  
filter-flags: 0x97 UCAST MCAST BCAST BEACON PROBEREQ RADARERR-5211  
opmode: AP (3)
```

1.1.6 queue

Available queues and queues being used and other information will be provided here.

```
cat /sys/kernel/debug/ieee80211/phyX/ath5k/queue
```

The information is formatted as shown below:

```
available txbuffers: 198
```

```
00: setup  
   len: 1 bufs: 1  
   stuck: 0  
01: setup  
   len: 0 bufs: 0  
   stuck: 0  
02: setup  
   len: 1 bufs: 1  
   stuck: 0  
03: setup  
   len: 0 bufs: 0  
   stuck: 0  
04: not setup  
05: not setup  
06: not setup
```

```
07: not setup
08: setup
   len: 0 bufs: 0
   stuck: 0
09: not setup
```

1.1.7 registers

Some of register information is provided here.

```
cat /sys/kernel/debug/ieee80211/phyX/ath5k/registers
The information is formatted as shown below:
```

```
AR5K_CR 0x00000004
AR5K_RXDP 0x2e8be180
AR5K_CFG0x00000100
AR5K_IER0x00000001
AR5K_BCR0x00000000
AR5K_RTSD0 0x00000000
AR5K_RTSD1 0x00000000
AR5K_TXCFG 0x00000015
AR5K_RXCFG 0x00000005
AR5K_RXJLA 0x2e8be170
AR5K_MIBC 0x00000000
AR5K_TOPS 0x00000008
AR5K_RXNOFRM 0x00000000
AR5K_TXNOFRM 0x00000000
AR5K_RPGTO 0x00000000
AR5K_RFCNT 0x0000001f
AR5K_MISC 0x00000000
AR5K_QCUDCU_CLKGT 0x00dfd770
AR5K_ISR0x00000000
AR5K_PISR 0x00100000
AR5K_SISR0 0x00000000
AR5K_SISR1 0x00000000
AR5K_SISR2 0x88000000
AR5K_SISR3 0x03000000
AR5K_SISR4 0x00000300
AR5K_IMR0x00000000
AR5K_PIMR 0x800914b5
AR5K_SIMR0 0x030f0000
AR5K_SIMR1 0x010f0000
AR5K_SIMR2 0x00070000
AR5K_SIMR3 0x00000000
AR5K_SIMR4 0x00000000
AR5K_DCM_ADDR 0x00000061
AR5K_DCCFG 0x00000000
AR5K_CCFG 0x00000000
AR5K_CPC0 0x00000000
AR5K_CPC1 0x00000000
AR5K_CPC2 0x00000000
AR5K_CPC3 0x00000000
AR5K_CPCOVF 0x00000000
AR5K_RESET_CTL 0x00000000
AR5K_SLEEP_CTL 0x00000000
AR5K_INTPEND 0x00000000
AR5K_SFR0x00000000
AR5K_PCICFG 0x00000014
AR5K_GPIOCR 0x00008000
AR5K_GPIODO 0x00000001
AR5K_SREV 0x000000a5
```

1.1.8 debug

debug lists all the available debug levels and display current debug level, as well as allow to change current debug level.

```
cat /sys/kernel/debug/ieee80211/phyX/ath5k/debug
```

The information is formatted as shown below.

```
DEBUG LEVEL: 0x00000000
```

```
reset 0x00000001 - reset and initialization
intr 0x00000002 - interrupt handling
mode 0x00000004 - mode init/setup
xmit 0x00000008 - basic xmit operation
beacon 0x00000010 - beacon handling
calib 0x00000020 - periodic calibration
txpower 0x00000040 - transmit power setting
led 0x00000080 - LED management
dumpbands 0x00000400 - dump bands
dma 0x00000800 - dma start/stop
ani 0x00002000 - adaptive noise immunity
desc 0x00004000 - descriptor chains
all 0xffffffff - show all debug levels
```

1.2 ath10k

1.2.1 fw_stats

fw_stats information is provided here.

```
cat /sys/kernel/debug/ieee80211/phyX/ath10k/fw_stats
```

The information is formatted as shown below:

```
ath10k PDEV stats
```

```
=====
```

```
Channel noise floor  -105
Channel TX power    44
TX frame count     13739057
RX frame count     13446628
RX clear count     28636859
Cycle count       2702159744
PHY error count    0
RTS bad count      0
  RTS good count   0
FCS bad count      0
No beacon count    2417
MIB int count      0
```

```
ath10k PDEV TX stats
```

```
=====
```

```
HTT cookies queued 0
HTT cookies disp.  0
  MSDU queued      44815
  MPDU queued      44815
MSDUs dropped      0
Local enqueued     44815
Local freed        44815
HW queued          44815
PPDUs reaped       44815
Num underruns      0
PPDUs cleaned      0
MPDUs requed       0
Excessive retries  0
  HW rate          3
Sched self tiggers 0
Dropped due to SW retries 0
  Illegal rate phy errors 0
  Pdev continous xretry 0
TX timeout         0
  PDEV resets      0
PHY underrun       0
  MPDU is more than txop limit 0
```

```
ath10k PDEV RX stats
```

```
=====
```

```
Mid PPDU route change 0
Tot. number of statuses 46671
Extra frags on rings 0 0
Extra frags on rings 1 0
Extra frags on rings 2 0
Extra frags on rings 3 0
MSDUs delivered to HTT 46671
```



```
MPDUs delivered to HTT 46671
MSDUs delivered to stack 46620
MPDUs delivered to stack 46620
Oversized AMSUs 0
PHY errors 0
PHY errors drops 0
MPDU errors (FCS, MIC, ENC) 51
```

```
ath10k PEER stats (1)
```

```
=====
```

```
Peer MAC address 03:00:00:00:04:f0
Peer RSSI 46166
Peer TX rate 0
Peer RX rate 6000
```

1.2.2 wmi_services

wmi_services information is provided here.

```
cat /sys/kernel/debug/ieee80211/phyX/ath10k/wmi_services
```

The information is formatted as shown below:

```
0x00 - BEACON_OFFLOAD - disabled
0x01 - SCAN_OFFLOAD - enabled
0x02 - ROAM_OFFLOAD - enabled
0x03 -BCN_MISS_OFFLOAD - enabled
0x04 - STA_PWRSERVE - enabled
0x05 - STA_ADVANCED_PWRSERVE - enabled
0x06 - AP_UAPSD - enabled
0x07 - AP_DFS - disabled
0x08 -11AC - enabled
0x09 - BLOCKACK - enabled
0x0a - PHYERR - enabled
0x0b - BCN_FILTER - enabled
0x0c - RTT - disabled
0x0d - RATECTRL - disabled
0x0e - WOW - enabled
0x0f - RATECTRL CACHE - enabled
0x10 - IRAM TIDS - enabled
0x11 - ARPNS_OFFLOAD - disabled
0x12 - NLO - disabled
0x13 - GTK_OFFLOAD - disabled
0x14 - SCAN_SCH - disabled
0x15 - CSA_OFFLOAD - disabled
0x16 - CHATTER - disabled
0x17 - COEX_FREQAVOID - disabled
0x18 - PACKET_POWER_SAVE - disabled
0x19 - FORCE FW HANG - disabled
0x1a -GPIO - disabled
0x1b - MODULATED DTIM - disabled
0x1c - BASIC UAPSD - disabled
0x1d - VAR UAPSD - disabled
0x1e - STA KEEP ALIVE - disabled
0x1f - TX ENCAP - disabled
```

1.3 Wireless Interfaces

Basic information related to all the wireless interfaces are provided in below command

```
cat /proc/net/wireless
```

```
Inter-| sta-| Quality | Discarded packets | Missed | WE
face | tus | link level noise | nwid crypt frag retry misc | beacon | 22

wlan0: 0000 0 0 0 0 0 0 0 0 0 0
wlan1: 0000 35. -75. -256 0 0 0 6 0 0
wlan2: 0000 0 0 0 0 0 0 0 0 0 0
wlan3: 0000 0 0 0 0 0 0 0 0 0 0

v_wlan0:0000 0 0 0 0 0 0 0 0 0 0
mon3: 0000 0 0 0 0 0 0 0 0 0 0
```

2 Transient Radio Commands

The transient radio commands in this section have immediate effect on the radio card, but these changes are not persistent across reboots.

2.3 Freq

This command sets the frequency of the radio card. The full command for this information is:

```
iwconfig wlanX freq <<frequency in MHZ>>M
```

2.4 Txpower

This command sets the power of the radio card. The full command for this information is:

```
iwconfig wlanX txpower <<index>> The index is a number between 0 and 63.
```

Or

```
iw dev <devname> set txpower <auto|fixed|limit> [<tx power in mBm>]
```

3 Node Level Information

3.1 Sta-List

This entry provides details of all the connected client devices and child mesh nodes. The full command for this information is:

```
cat /proc/net/meshap/access-point/sta-list
```

The format of the information is show below:

```
-----
----- STATION ADDR      |IFNAME|VTAG|SIG|RATE|LAST PACKET
RX|REFCOUNT|ACL|BUF|COMP| KI|
-----
---
00:07:ba:a3:7e:33| wlan2|0000|034|0036|00000000060460|00000002|No |No | No
|000|
00:1d:e0:ae:df:e5| wlan2|0000|022|0036|00000000009180|00000002|No |No | No
|000|
00:0b:85:55:9c:3e| wlan2|0000|003|0001|00000000545690|00000002|No |No | No
|000|
```

The description of various fields is given below:

| | |
|----------------|--|
| STATION ADDR | The MAC address of the connected device |
| IFNAME | The radio interface on which the device is |
| VTAG | The VLAN tag associated with the device |
| SIG | The average RSSI of the packets received from |
| RATE | The transmit rate of the last packet received |
| LAST PACKET RX | The milli-seconds since the last received packet |
| ACL | Set to 'YES' if the device was connected using a |
| KI | The encryption key index if security is enabled |

Use the sta-list entry to troubleshoot client device connectivity issues. Using the SIG and RATE fields, one can determine the signal and rate of the packets received from the device.

The VTAG field can be used to determine if the client device is being assigned a VLAN tag. The LAST PACKET RX field can be used to determine if any packets are being received from the client.

3.2 Downlink Stat

This entry provides details and statistics for all the downlinks on the node. The full command for this information is:

```
cat /proc/net/meshap/access-point/downlink-
stat
```

The format of the information is show below:

```

-----
----- IFNAME|MCC|DCC|UBCAST      |TBCAST
              |UBCASTR |
-----
-----
wlan0|000|000|000000155958|000000128036|0000000
              00000|
wlan2|000|004|000000165455|000000000000|0000000
              00000|
wlan3|000|000|000000059544|000000000000|0000000
              00000|
ixp1 |000|000|000000173245|000000164105|000000000
              0000|

```

The description of various fields is given below:

| | |
|---------|--|
| IFNAME | The interface name of the downlink |
| MCC | Number of mesh child nodes connected to the |
| DCC | Number of client devices connected to the |
| UBCAST | Number of untagged broadcast/multicast packets transmitted by the |
| TBCAST | Number of tagged broadcast/multicast packets transmitted by the |
| UBCASTR | Number of untagged broadcast/multicast packets transmitted by the downlink using the MESH IGNORE bit turned on. <i>MESH IGNORE bit is used to send broadcast/multicast packets to client devices when a corresponding tagged version of the same packet was broadcast for child mesh nodes. When the MESH IGNORE bit is set, the child mesh nodes</i> |

Use the downlink-stat entry to troubleshoot applications that make use of broadcast/multicast packets.

E.g. Spectralink push-to-talk uses multicast packets so that all phones in the network will hear the voice.

The UBCASTR field is relevant if such applications are used over a VLAN.

3.3 Vlan-Stat

This entry provides statistics for all the VLANs configured on the node. The full command for this information is:

```
cat /proc/net/meshap/access-point/vlan-stat
```

The format of the information is show below:

```

-----
VTAG|ENCS|ENCB| wlan0| wlan2| wlan3|      ixp1|
-----

```

The description of various fields is given below:

| | |
|-------|--|
| VTAG | The VLAN tag |
| ENCS | 'Yes' if security is enabled on the settings for |
| ENCB | 'Yes' if security is currently activated for |
| wlanX | The count of the number of associated client devices for this VLAN |
| ixpX | The count of the number of associated client devices for this VLAN |

3.4 Adhoc

This entry provides information for disjoint-adhoc mode operation. The full command for this information is:

```
cat /proc/net/meshap/mesh/adhoc
```

The format of the information is show below:

```
NODE OPERATING AS FFN STEP SCAN INDEX 0
```

```
-----
PARENT ADDR          |CHN|SIG|RATE|TRATE|FLAGS          |FNC|R-VAL|
-----
00:12:ce:00:00:6e|060|046|0054|00100|P - - - - -|FFR|08192|
00:12:ce:00:20:da|157|085|0006|00100|- - - - -|FFR|04096|
00:12:ce:00:00:b2|060|046|0048|00054|- - - - -|FFN|24576|
00:12:ce:00:00:02|165|033|0006|00054|- - - - -|FFN|00000|
00:12:ce:00:11:98|149|036|0006|00054|- - - - -|FFN|24576|
-----
```

The first line of the information provides the current operating mode for the node and also provides the current channel being used for the "STEP SCAN" process.

The description of various fields is given below:

| | |
|-------------|---|
| PARENT ADDR | The MAC address of the prospective parent |
| CHN | Channel |
| SIG | RSSI |
| RATE | The transmit rate |
| TRATE | The parent's RATE to its parent. A root node's |
| FLAGS | 'P': PREFERRED PARENT 'M': MOBILE NODE 'C': CHILD 'D': DISABLED 'L': LIMITED FUNCTIONALITY 'W': DISABLED FOR LFR ARBITRATION 'Q': QUESTIONABLE PARENT |
| FNC | The node functionality |
| R-VAL | DHCP random value |

3.5 Reboot

This entry displays the status of the last reboot operation.

The full command for this information is: `cat /proc/net/meshap/reboot`

The format of the information is show below:

```
Code: 5 Flag: 1
```

Process:
 Mem Used: 0 MB
 Temp: 30
 Voltage: 23
 PC REG: FFFFFFFF
 CALLER REG: FFFFFFFF
 STACK DUMP: 0

The table below lists the various Code values and their meaning.

| CODE | DESCRIPTION |
|------|---|
| 1 | One of the downlink radio cards was not detected |
| 2 | Out of primary packet buffers for a long time (improbable) |
| 3 | Out of radio hardware packets for a long time (improbable) |
| 4 | Out of radio hardware descriptors for a long time |
| 5 | Rebooted via software |
| 6 | Firmware crashed |
| 128 | Heartbeat thread was stalled |
| 129 | Root node ixp0 ETHERNET link was down and rebooted to |
| 130 | Relay node ixp0 ETHERNET link was up and rebooted to become |
| 131 | FIPS 140-2 POST failure for AES (improbable) |
| 132 | FIPS 140-2 POST failure for HMAC-SHA1 (improbable) |
| 133 | Routing tables corrupted (improbable) |
| 134 | FIPS 140-2 RNG repeated (improbable) |
| 135 | General watchdog wake-up (improbable) |

The STACK DUMP is present only when the Code value is 6.

3.6 Coord

This entry displays the status of the current GPS coordinates for the node. The full command for this information is: **cat /proc/brdinfo/coord**

3.7 Speed

This entry displays the speed in km/h
 The full command for this information is: **cat /proc/brdinfo/speed**

3.8 Temp

This entry displays the current temperature of the node.
 The full command for this information is: **cat /proc/brdinfo/temp**

3.9 Voltage

This entry displays the input voltage of the node.
 The full command for this information is: **cat /proc/brdinfo/voltage**

3.10 Kap

This entry provides information about all the known neighboring nodes. The full command for this information is:

cat /proc/net/meshap/mesh/kap

3.10.1 Information format for Stationary nodes

```
-----
PARENT ADDR      |CHN|SIG|RATE|TRATE|NEXTSAMPLE|DISC  |FLAGS  |SAMCNT|MAXSAM|
-----
00:12:ce:00:00:6e|060|036|0054|00100|0000012850|0000000|- - - -|000000|000000|
00:12:ce:00:20:da|157|032|0036|00100|0000056520|0000000|- - - -|000000|000000|
00:12:ce:00:00:14|052|014|0024|00054|0000000000|0000000|- M - - -|000020|000050|
-----
```

The command lists the known neighboring nodes in the order of their preference. The description of various fields is given below:

| | |
|-------------|---|
| PARENT_ADDR | The MAC address of the prospective parent |
| CHN | Channel |
| SIG | RSSI |
| RATE | The transmit rate |
| TRATE | The parent's RATE to its parent. A root node's |
| NEXTSAMPLE | The number of milli-seconds left until the parent is scheduled for link sampling |
| DISC | The number of times the node has been disconnected from the |
| FLAGS | 'P': PREFERRED PARENT 'M': MOBILE NODE 'C': CHILD 'D': DISABLED 'L': LIMITED FUNCTIONALITY 'W': DISABLED FOR LFR ARBITRATION 'Q': QUESTIONABLE PARENT (MESH-INIT or Rebooted) |

The criteria used by stationary nodes in choosing the "Best" parents is as follows:

- Lower of RATE and TRATE values are used for calculating score
- The DISC value is used to decrease the score.
- For two parent's with the same score
 - If one of them is the current parent, it gets preference
 - If one of them is mobile, then the stationary node gets preference
 - The one with the higher tree bit rate gets preference
 - If tree bit rates are the same, then the higher direct bit rate gets preference
 - If the direct bit rates are the same, then the higher SIG value gets preference
 - If the SIG values are the same the lower hop count gets preference.
- Child mesh nodes (direct and in-direct) cannot be considered as a parent
- "Disabled" and "Questionable" are not considered until their status changes.

3.10.2 Information format for mobile nodes

```

-----
PARENT ADDR          |CHN|SIG|RATE|TRATE|LAST SEEN |PATCNT |
-----
00:12:ce:00:00:6e|060|047|0054|00100|0000000650|0000012|
00:12:ce:00:20:da|157|086|0054|00100|0000000650|0000000|
00:12:ce:00:00:b2|060|048|0054|00054|0000000650|0000000|
00:12:ce:00:00:02|165|038|0054|00054|0000000650|0000000|
00:12:ce:00:11:98|149|037|0054|00054|0000000650|0000000|
00:12:ce:00:22:d8|052|015|0048|00018|0000005270|0000000|
00:12:ce:00:25:ea|052|013|0006|00100|0000012200|0000000|

```

The command lists the known neighboring nodes in the order of their preference.

The description of various fields is given below:

| PARENT ADDR | The MAC address of the prospective parent |
|-------------|---|
| CHN | Channel |
| SIG | RSSI |
| RATE | The transmit rate |
| TRATE | The parent's RATE to its parent. A root node's |
| LAST SEEN | The number of milli-seconds since the parent's |
| PATCNT | The number of consecutive times the parent was the 'best parent' in |

4 Mobile Node Configuration Options

4.1 Sliding Scan Window and "winning" Parent Nodes

Sliding Scan Window: To make switching decisions, the scanner radio of a mobile node measures the beacons received from the downlink or AP radios of potential parent nodes.

Mobile nodes maintain a **sliding window** of scan results. The default size of the **sliding window** is 12 *scan intervals*. An individual *scan interval* lasts for 250ms (by default). This provides the mobile node with a sliding history of the last 3 seconds (12 scans x 250ms/scan = 3 seconds).

During the *scan interval*, equal time is spent on the each channel in the Scan Channel List of the mobile node's scanner radio. For example, if there are 5 channels in the Scan Channel List, then the scanner radio will listen on each channel for 50ms.

Winning Parent Nodes:

- After each scan, a "snapshot" will be taken of the prior 12 scans.
- Within this snapshot, there will be a list of "winners" for the 12 scans (for example, parent "A" wins 6 times, parent "B" wins 4 times, and parent "C" wins 2 times ...6 + 4 + 2 = 12). A "winner" is defined by the parameters in section 4.2. A parent with a missed beacon will lose for the interval.
- The parent node that has the most number of wins in the snapshot will be the **overall winner** of the snapshot.
- If a particular parent is the **overall winner** of four consecutive snapshots, this will become the new actual parent.

4.2 Default Preference and Selection Criteria

Mobile Node's Default Preference: By default, mobile nodes prefer *stationary* nodes as parents over *mobile* nodes. If the mobile node sees both stationary and mobile nodes as potential parents, it will associate to stationary nodes even if the signal strength is considerably lower than those of nearby mobile nodes. A mesh command provides the flexibility to modify this behavior. This is covered in the next section.

Mobile Node's Best Parent Selection Criteria: The criteria used for selecting the winning parent for each scan is based on the following criteria (in order of priority):

- Strongest signal is preferred.
- Two parents with signal strengths within a DAMPING FACTOR of each other are considered to have the same **signal value**. The default value of the DAMPING FACTOR is 6dB. (for example, if the signal from parent "A" is -60dB and the signal from parent "B" is -66dB, they have the same **signal value**).
- If two parents have the same signal value, and one of them is the current parent, then the current parent remains preferred.
- If two potential parents have equal signal values, the potential parent with the higher *connectivity* is preferred.
- If the connectivity values are the same, the parent with the lower hop count is preferred.

- Child mesh nodes (direct and indirect) cannot be considered as parents. This prevents loops from being formed within the mesh.
- "Disabled" and "Questionable" nodes (see section 3.10.1 under "FLAGS") are not considered until their status changes.
- Parents with a LAST SEEN time (see section 3.10.2) that is outside of the current scan window are not considered. .

MobParams

The following mesh command is used to modify the parent-selection process:

```
alconfset mobparams patcnt[4] patmax[12] sigdamp[6] [static_thres=0] [scanint=250] [prefstatic=1 (0/1)] [downscan=2] [adv_root_mobile=1]
```

The first three parameters are mandatory:

1. **patcnt**: The number of consecutive times a parent node has to be the **overall winner** of the sliding-window snapshot (see section 4.1) in order to become the **actual parent**. The default **patcnt** value is 4, and valid values are ≥ 1 .
2. **patmax**: The length of the sliding window. The default value is 12, and valid values are ≥ 2 .
3. **sigdamp**: The signal damping factor when a mobile node is measuring the relative beacons of two parents. Signal strengths of two parent nodes that are within the damping factor are considered to have equal **signal values** (for example, if the signal damping factor is 6dB, and the signal from parent "A" is -60dB and the signal from parent "B" is -66dB, they have the same **signal value**). The default value is 6, and valid values are ≥ 0 .

The next five parameters are optional:

4. **static_thres**: A signal strength in dB above the noise floor (for example, if a signal is -86dB, and the noise floor is -96dB, the **static_thres** is 10dB). For **static_thres** values > 0 , a static (non-mobile) node is preferred as a parent over a mobile node as long as its signal is greater than [noise floor + static_thres]. The default value is 0, and valid values are ≥ 0 .

- Scenario 1: Static Parent Node Signal = -80dB, Mobile Parent Node Signal = -50dB, noise floor = -96dB, static_thres = 0: The Static Parent Node is preferred even though the Mobile Parent Node has a better signal strength.

- Scenario 2: Static Parent Node Signal = -80dB, Mobile Parent Node Signal = -50dB, noise floor = -96dB, static_thres = 20dB: The Mobile Parent Node is preferred since the Parent Node Signal is less than [noise floor + static_thres = -76dB].

5. **scanint**: The duration in milliseconds a scanner radio spends listening on the channels in its Scan Channel List. Each channel gets the same amount of scan time (for example, if the **scanint** = 250ms, and there are 5 channels in the Scan Channel List, then each channel gets 50ms of scan time). The default value is 250ms, and valid values are ≥ 1 .

Configuring the **scanint** such that each channel gets less than 50ms of scan time is not recommended. This will reduce the dwell time on a channel, and therefore, increase the probability of a beacon miss.

It is important to remember that increasing the number of channels in the Scan Channel List of a mobile node's scanner radio will decrease the scan time for each channel. In this case, two things can be done to counter this effect: the **scanint** can be increased, or, the beacon interval of the parent nodes' downlinks/APs can be decreased. To decrease the beacon interval of the parent nodes' downlink/AP radios, the **alconfset beacint wlan<<0/1/2/3>> << integer >>** command can be used.

6. **prefstatic**: The mobile node will prefer *static* parent nodes over *mobile* parent nodes. The default value is 1 (a value of "1" means that the mobile node prefers a *static* parent node over a *mobile* parent node), and valid values are 0 or 1. *****If** the value is 1, then static and mobile nodes are evaluated by the criteria described by the **static_thres** setting.

[Note: regardless of the Prefstatic setting, the static_thres setting > 0 19

takes precedence. If the value is greater than 0 then the *prefstatic* setting is ignored. Consider the following four cases:

- *Static_thres* =0 and *PrefStatic* =1. Static parent nodes will always be preferred over mobile parent nodes regardless of the signal strength from either.
- *Static_thres* =0 and *PrefStatic* =0. Static parent nodes will not always be preferred over mobile parent nodes. If mobile parent nodes have a stronger signal strength, they will be preferred.
- *Static_thres* =20 and *PrefStatic* =1. Static nodes will be preferred as long their signal strength is stronger than $-96\text{dB}+20\text{dB} = -76\text{dB}$. Else, both static parent nodes and mobile parent nodes will be evaluated on the same criteria.

7. downscan: Controls how the mobile node changes channels dynamically on its downlinks:

0: No scanning during operation (downlink will continue use channel selected upon boot-up).

1: Scan on every parent shift. When the mobile node's uplink makes a change in association to another parent node, its downlink will select another channel on which to transmit (if needed) such that its uplink and downlink are transmitting on different channels.

2: Scan on every scan interval. This is the default behavior -which provides the most dynamic reaction to changing interference from external sources.

8. adv_root_mobile: When a mobile node is a *root* node, it will advertise itself as a *static* node by default. This can effect the *preference of association* for its surrounding nodes (as per the "**prefstatic**" setting). In certain applications, such as convoys, it is often desired that none of the nodes in the mesh advertise themselves as being static. The default setting is 1, where a mobile root will advertise itself as being static, and valid values 0/1.

4.2.1 Usage Consideration

A. The order of optional parameters is important. For example, if it is desired that the **static_thres** option be configured, the values for the preceding parameters of **patcnt**, **patmax**, and **sigdamp** must be included in the command. The succeeding parameters of **scanint**, **prefstatic**, **downscan**, **adv_root_mobile** do not have to be included. Examples:

- **alconfset mobparams 3 12 3** Å Change patcnt to 3 overall wins, change damping factor to 3 db. Optional settings are unchanged.
- **alconfset mobparams 4 12 6 0 500 0** Å scan interval changed to 500ms, do not prefer static nodes, remaining optional settings are unchanged.
- **alconfset mobparams 4 12 6 20 250 1** Å prefer static nodes, but only ones that are 20dB above the noise floor, else, no node preference. Remaining optional settings are unchanged.
- **alconfset mobparams 4 12 6 0 250 1 0 1** Å prefer static nodes, no scanning on the downlink(s) during operation, advertise mobile-root node as "static".

B. A node whose last beacon heard was outside of the sliding scan window [**scan_interval** x **patmax**] is not considered as a possible parent by the mobile node. To see the current sliding window, use the command:
cat/proc/net/meshap/mesh/mobwin.

C. If the sliding scan window is set (by default) at 250ms and there are 5 channels being scanned (5 channels in the scan list) then each channel is scanned for 50 ms. If the beacon is at 100 ms then the probability of hearing the beacon is 50% (note that it is not necessary for a beacon from a particular parent to be heard on every scan interval, but the beacon must be heard a sufficient number of times in the sliding scan window). To increase the probability, consider:

- Reducing all prospective parent downlink beacon intervals by using **alconfset beacint wlan <<0/1/2/3>> << integer >>**

- Reducing the number of channels being scanned. The number of channels being scanned is controlled by populating, or depopulating the Scan Channel List of the mobile node's scanner radio. These channels correlate to the channels being provided by the downlinks of the parent nodes in the mesh.

- Increasing the **patmax** (number of scans in the sliding window).

- Increasing the **scanint** ("scan interval" -the amount of time spent scanning the channels in the Scan Channel List of mobile node's scanner radio).

These options are also helpful for noisy/dynamic RF environments, where the probability of detecting beacons is reduced due to external RF interference.

D. To see when last beacon from a specific parent was received, execute the following command: **cat/proc/net/meshap/mesh/kap.**

This will show the time (in 'ms') since the last beacon as heard, and also, the number of consecutive wins the node has in the sliding scan window.

E. When mobile nodes display preference to static nodes over mobile nodes, change static node preferences (**pref_static**), or adjust the **static_thresh** setting.

F. All "**alconfset mobparams**" settings will be set back to default after rebooting the node. Typing in the following command after each "**alconfset mobparams**" command is entered can avert this: **alconfset save** For example:

Type "**alconfset mobparams 4 12 5 0 250 1 0 1**", click "**Enter**" Type "**alconfset save**", click "**Enter**"

4.1 Mobwin

This entry displays the current status of the Sliding Scan Window for mobile nodes.

The full command for this information is:

```
cat /proc/net/meshap/mesh/mobwin
```

The format of the information is show below:

```
-----  
PARENT ADDR      |CHN|SIG|ITEM ID |CNT|  
-----  
00:12:ce:00:01:6e|060|046|C2AA3FC0|012|  
00:12:ce:00:02:6e|060|046|C29091E0|012|  
00:12:ce:00:03:6e|060|036|C2909200|011|  
00:12:ce:00:04:6e|060|026|C2909220|012|  
00:12:ce:00:05:6e|060|016|C2909240|013|
```

The evaluation results of each scan in the current window are listed in the output.

| | |
|-------------|--|
| PARENT ADDR | The MAC address of the best parent for the scan |
| CHN | Channel |
| SIG | RSSI |
| CNT | The number of consecutive times the parent was th 'best parent' in the current scan window. |

5 Miscellaneous Commands

5.1 Perf

This command sets the preferred parent for the node without changing its settings. Since the settings are not changed, the preferred parent will not be maintained after a node is rebooted.

The full command is: `meshd pref <<enable>> <<mac_address>>`

Where

enable: Set to 1 to enable the feature and 0 to disable the feature.

mac_address: Set to the desired preferred parent's downlink mac address

For disabling the feature the `mac_address` value needs to match the one provided when enabling. **The command takes a minimum of 3 heartbeat intervals to take effect.**

5.1 Kick

This command disassociates a client device or a mesh node and forces the node to execute all procedures when a client device or a mesh node disconnects.

The full command is: `meshd kick <<mac_address>>`

5.2 fwupdate

This command is used to determine whether a unit's firmware can be upgraded via the NMS. The full command is: `cat /proc/mtd | grep fwupdate` If the node's firmware can be upgraded from the NMS, the output of this command will be something like:

```
mtd7: 00140000 00020000 "fwupdate"
```

5.3 gpsd

This command is used to determine whether the GPS feature on the node has been activated. The full command is: `pidof gpsd`. If the GPS feature is not active, the output will be a blank line.

5.4 Meminfo

This command is used to determine the amount of free SDRAM on the unit.

The full command is: `cat /proc/meminfo`

Typically, the free memory is between 14 and 19 MB. If the node's firmware has been upgraded, and it has not been rebooted, the amount of free memory is less than 10 MB.

5.5 telnetd

This command enables the Telnet daemon on the node.

The full command is: `cp /bin/otelnetd`

`/bin/telnetd` Enabling the telnet daemon on the unit exposes the node to security risks, hence, only enable telnet for troubleshooting purposes. After usage, reboot the node to disable telnet access.

6 Alconfset and Alconfget Commands

6.1 Introduction

The alconfset command is used to manually configure mesh related parameters on the node in the absence of NMS controller. The configured parameter will be written to the respective .conf files and for it to take effect the node needs to be rebooted.

6.2 Usage

The usage and supported options for alconfset and alconfget command are available as follows.

```
alconfset [-v] [-h] [-f configfile] fieldname [fieldparams]
```

```
alconfget [-v] [-h] [-f configfile] fieldname [fieldparams]
```

-v: to know the version.

-h: for help

-f: specify config file (/etc/meshap.conf is default)

6.3 Supported Parameters

Below are the supported parameters for alconfset and alconfget command which can be used to set and get the values respectively.

Generic fields:

mesh_id, mesh_imcp_key, name, globdca, swc, aging, regdom, country, medtype, subtype, usetype, channel, essid, rts, frag, beacint, dca, dcalist, antport, txrate, preamble.slot, linkopt, acktime, hidessid, txpower, prefpar, hbint, igmp, adhoc, dhcp, forcedroot, fips, dfs, mobindex, mobmode, gps, logmon, location, acwmin, acwmax, aifsn, backoff, burst, autostart, usevirt, option, save, savefw, failOverEthernet, server_ip_addr, mgmt_gw_addr, mgmt_gw_enable, mgmt_gw_certificates, disable_backhaul_security,

802.11N supported fields:

ldpc, smps, tx_stbc, rx_stbc, delayed_ba, gfmodes.

802.11ac supported fields:

max_mpdulen, supported_channel_width, rx_ldpc, gi_80, gi_160, vtx_stbc, vrx_stbc, su_beamformer_cap, su_beamformee_cap, beamformee_sts_count, sounding_dimensions, mu_beamformer_cap, mu_beamformee_cap, vht_txop_ps, htc_vht_cap, rx_ant_pattern_consistency, tx_ant_pattern_consistency, vht_oper_bandwidth, seg0_center_freq, seg1_center_freq.

6.4 Detail use of Parameters

Generic fields:

mesh_id:

Description: Assign a node to a specific mesh network. An alphanumerical string which helps to connect the nodes into Mesh-network.

Usage alconfset: alconfset mesh_id <<value>>

Usage alconfget: alconfget mesh_id

mesh_imcp_key:

Description: Set the encryption key for imcp packets.

Usage alconfset: alconfset mesh_imcp_key <<self_mac_address>> <<value>>

Usage alconfget: alconfget mesh_imcp_key

name:

Description: Node name parameter is used to set the name of the node.

Usage alconfset: alconfset name <<value>>

Usage alconfget: alconfget name

globdca:

Description: To set the global dca value. It either can be 0 or 1.

Usage alconfset: alconfset globdca <<value=0 or 1>>

Usage alconfget: alconfget globdca

sac:

Description: stay awake count for the mesh node.

Usage alconfset: alconfset sac <<value (non-zero positive)>>

Usage alconfget: alconfget sac

aging:

Description: Bridge aging timeout value. This is the inactivity time for stations to disconnect from AP.

Usage alconfset: alconfset aging <<value (non-zero possitive)>>

Usage alconfget: alconfget aging

regdom:

Description: To set the regulatory domain code

Usage alconfset: alconfset regdom <<0=NONE, 1=FCC, 2=ETSI, 3=CUSTOM>>

Usage alconfget: alconfget regdom

country:

Description: To set the country code. Set as needed to indicate country in which device is operating. This can limit available channels and transmit power.

Usage alconfset: alconfset country <<value (non-zero positive)>>

Usage alconfget: alconfget country

medtype:

Description: To set the medium type of a particular interface to either ethernet or wireless.

Usage alconfset: alconfset medtype <<if-name>> <<value 0=ethernet,1=802.11>>

Usage alconfget: alconfget medtype <<if-name>>

subtype:

Description: To set the subtype for a particular interface. It can be any supported wireless protocol.

Usage alconfset: alconfset subtype <<if-name>>
<<value, a,b,g,bg,psq,psh,psf,n_2_4G,n_5G,ac,bgn,an,anac

Usage alconfget: alconfget subtype <<if-name>>

usetype:

Description: To set the particular interface to act as downlink(wm), uplink(ds), Passive-Monitoring(pmon) or Active-monitoring(amon).

Usage alconfset: alconfset usetype <<if-name>> <<value ds,wm,pmon,amon>>

Usage alconfget: alconfget usetype <<if-name>>

channel:

Description: To set the channel value on which interface has to work.

Usage alconfset: alconfset channel <<if-name>> <<channel>>

Usage alconfget: alconfget channel <<if-name>>

essid:

Description: The Extended Service Set Identification (ESSID) is used to set the name for interface which will be visible to all other nodes in same network.

Usage alconfset: alconfset essid <<if-name>> <<essid>>

Usage alconfget: alconfget essid <<if-name>>

rts:

Description: To set the packet size at which packet transmission is governed by the RTS/CTS transaction.

Usage alconfset: alconfset rts <<if-name>> <<rts>>

Usage alconfget: alconfget rts <<if-name>>

frag:

Description: To set the Fragmentation Threshold to a maximum length of the frame, beyond which payload must be broken up (fragmented) into two or more frames.

Usage alconfset: alconfset frag <<if-name>> <<frag>>

Usage alconfget: alconfget frag <<if-name>>

beacint:

Description: To set the beacon interval .

Usage alconfset: alconfset beacint <<if-name>> <<interval>>

Usage alconfget: alconfget beacint <<if-name>>

Dca:

Description: To set the Dynamic Channel Assignment(dca) value per interface(Can be either 1 or 0). If dca is not present than globdca value will be used for that particular interface.

Usage alconfset: alconfset dca <<if-name>> <<0-1>>

Usage alconfget: alconfget dca <<if-name>>

Dcalist:

Description: dcalist is nothing but the channel list on which configured interface(mostly AP) has to operate on.

Usagealconfset:alconfset dcalist <<if-name>> <<count>> <<channels>>

Usagealconfget:alconfget dcalist <<if-name>>

Antport:

Description: To set the antenna port for interface.

Usagealconfset:alconfset antport <<if-name>> <<1-14>> [1 for RX on all, 0 otherwise]

Usagealconfget:alconfget antport <<if-name>>

Txrate:

Description: To set the maximum transmission rate through which packet will be sent out.

Usagealconfset:alconfset txrate <<if-name>> <<mbps/0=auto>>

Usagealconfget:alconfget txrate <<if-name>>

Preamble:

Description: preamble allows the receiver to acquire the wireless signal and synchronize itself with the transmitter. It can be either set to short(1) or long(0).

Usagealconfset:alconfset preamble <<if-name>> <<long or short>>

Usagealconfget:alconfget preamble <<if-name>>

slot:

Description:

Usagealconfset:alconfset slot <<if-name>> <<long or short>>

Usagealconfget:alconfget slot <<if-name>>

linkopt:

Description:

Usagealconfset:alconfset linkopt <<if-name>> <<all/backhaul/client>>

Usagealconfget:alconfget linkopt <<if-name>>

acktime:

Description: To set the acknowledgement time in micro second.

Usage alconfset: alconfset acktime <<if-name>> <<µs>>

Usage alconfget: alconfget acktime <<if-name>>

hidessid:

Description: Set to 1 to disable broadcasting device SSID to the network.

Usage alconfset: alconfset hidessid <<if-name>> <<0 or 1>>

Usage alconfget: alconfget hidessid <<if-name>>

txpower:

Description: To set the transmission power of an AP.

Usage alconfset: alconfset txpower <<if-name>> <<0 to 100>>

Usage alconfget: alconfget txpower <<if-name>>

Prefpar:

Description: To set the preferred parent by selecting the MAC address of the parent you want to connect with.

Usage alconfset: alconfset prefpar <<MAC-ID>>

Usage alconfget: alconfget prefpar

Hbint:

Description: Heartbeat Interval is set in order to wake up the lock at scheduled intervals. By default it is set to 15.

Usage alconfset: alconfset hbint <<value (non-zero possitive)>>

Usage alconfget: alconfget hbint

Igmp:

Description: When enabled(set to 1), IGMP Snooping monitors IGMP communications among devices and optimizes wireless multicast traffic.

Usage alconfset: alconfset igmp <<0 or 1>>

Usage alconfget: alconfget igmp

*Adhoc:

Description: to enable the adhoc mode

Usage alconfset: alconfset adhoc <<adhoc_mode=0/1>> <<begin_in_infra=0/1>>
<<sectored_usage=0/1>>

Usage alconfget: alconfget adhoc

dhcp:

Description: Dynamic host control protocol(DHCP) can be enabled by 1 and disable by 0. While enabling follow below alconfset usage for other command options.

Usage alconfset: alconfset dhcp <EN> [<MODE> <NET> <MASK> <GW> <DNS> <LT>]

EN = 1 for enabling, 0 for disabling

MODE = random or fixed

NET = Network in dotted decimal form

MASK = Subnet mask in dotted decimal form

GW = Gateway IP in dotted decimal form

DNS = DNS IP in dotted decimal form

LT = Lease time in seconds

Usage alconfget: alconfget dhcp

Forcedroot:

Description: used to make root node forcefully. Enabled by setting it to 1 and disable by 0.

Usage alconfset: alconfset forcedroot <<0 or 1>>

Usage alconfget: alconfget forcedroot

fips:

Description: Federal Information Processing Standard (FIPS) defines security and interoperability requirements . It can be enabled by setting to 1 and disable by 0.

Usage alconfset: alconfset fips <<0 or 1>>

Usage alconfget: alconfget fips

dfs:

Description: The concept of DFS is to have the unlicensed device detect the presence of a radar system on the channel they are using. It can be enabled by setting it to 1 and can be disabled by 0.

Usage alconfset: alconfset dfs <<0 or 1>>

Usage alconfget: alconfget dfs

*Mobindex:

Description:

Usage alconfset: alconfset mobindex <<0-20>>

Usage alconfget: alconfget mobindex

*mobmode:

Description: To enable mobility power save mode can be set to 1 or else set to 0.

Usage alconfset: alconfset mobmode <<0 or 1>>

Usage alconfget: alconfget mobmode

gps:

Description: Global Positioning System (GPS) can be enabled by setting it to 1 and disable by 0. While enabling follow the below alconfset usage for other command options.

Usage alconfset: alconfset gps <EN> [<device> <dest_ip> <dest_port> <tx_interval>]

EN = 1 for enabling, 0 for disabling

device = Serial port device e.g. /dev/ttyS0

dest_ip = Push destination IP address

dest_port = Push destination UDP port

tx_interval = Push destination transmit interval

*logmon:

Description: logging to remote server.

Usage alconfset: alconfset logmon <dest_ip> <dest_port>

dest_ip = Push destination IP address

dest_port = Push destination UDP port

Usage alconfget: alconfget logmon

location:

Description: Location can be enabled by setting it to 1 and disable by 0.

Usage alconfset: alconfset location <<0 or 1>>

Usage alconfget: alconfget location

acwmin:

Description: levels of priority are called as access categories which lies from 0 to 3 and minimum contention window size can be set from below mentioned values. In QoS, if the medium is busy the station will do an exponential backoff between contention window min and contention window max, after which the frame is transmitted.

Usagealconfset: `alconfset acwmin <<category [0-3]>>
<<value[0,1,3,7,15,31,63,...,1023]>>`

Usagealconfget: `alconfget acwmin`

acwmax:

Description: levels of priority are called as access categories which lies from 0 to 3 and maximum contention window size can be set from below mentioned values. In QoS, if the medium is busy the station will do an exponential backoff between contention window min and contention window max, after which the frame is transmitted.

Usagealconfset: `alconfset acwmax <<category [0-3]>>
<<value[0,1,3,7,15,31,63,...,1023]>>`

Usagealconfget: `alconfget acwmax`

aifsn:

Description: Arbitration Interframe Spacing Number(AIFSN) values are administrator configurable with proper categories. This is the minimum duration the frame to be transmitted will wait when the medium is free.

Usagealconfset: `alconfset aifsn <<category [0-3]>> <<value>>`

Usagealconfget: `alconfget aifsn`

Backoff:

Description: Backoff can be enabled by 1 and disable by 0 with appropriate category.

Usagealconfset: `alconfset backoff <<category [0-3]>> <<value = 0 or 1>>`

Usagealconfget: `alconfget backoff`

Burst:

Description: To set the burst time value with appropriate category.

Usagealconfset: `alconfset burst <<category [0-3]>> <<value>>`

Usagealconfget: `alconfget burst`

Autostart:

Description: Autostart is a list of below sub-commands.

<alconfset autostart list> will show you the number of blocks created and the respective task has to be done for each block.

<alconfset autostart create> This will create a new block.

<alconfset autostart add> new command can be added to newly created or already exist block by selecting block number.

<alconfset autostart del> This will delete block with provided block number.

Usage alconfset: alconfset autostart list

```
alconfset autostart create
```

```
alconfset autostart add <<command-block-number>> <<command>>
```

```
alconfset autostart del <<command-block-number>>
```

Usage alconfget: Not implemented.

usevirt:

Description: This is nothing but to create virtual interface from wlan0 if it set to 1. Note: In meshap.conf file, this can be find as use_virt_if.

Usage alconfset: alconfset usevirt <<value (0 or 1)>>

Usage alconfget: alconfget usevirt

* Option:

Description:

Usage alconfset: alconfset option list

```
alconfset option add option-key
```

```
alconfset option del option-index
```

Save:

Description: To save the configurations has been done using alconfset command.

Usage alconfset: alconfset save

Usage alconfget: Not implemented.

savefw:

Description: to save the firmware configurations has been done using `alconfset` command.

Usage `alconfset`: `alconfset savefw`

Usage `alconfget`: Not implemented.

***FailOverEthernet:**

Description: Enable LFRS mode in the node.

Usage `alconfset`: `alconfset failOverEthernet <<off/enable/enable_with_ping=0/1/2>>`
`<<power_on_default=0/1>>` `<<scan_freq_secs=0/1>>`

Usage `alconfget`: `alconfget failOverEthernet`

server_ip_addr:

Description: To set the server ip address.

Usage `alconfset`: `alconfset server_ip_addr <IP_ADDR>` eg: `alconfset server_ip_addr 10.10.10.10`

Usage `alconfget`: `alconfget server_ip_addr`

mgmt_gw_addr:

Description: To set the management gateway address. Used for running NMS in a remote site.

Usage `alconfset`: `alconfset mgmt_gw_addr <IP_ADDR>` eg: `alconfset mgmt_gw_addr <url:port>`

Usage `alconfget`: `alconfget mgmt_gw_addr`

mgmt_gw_enable:

Description: management gateway can be enabled by setting it to 1 and disable by 0.

Usage `alconfset`: `alconfset mgmt_gw_enable <0/1>`

Usage `alconfget`: `alconfget mgmt_gw_enable`

mgmt_gw_certificates:

Description: This helps to set the management gateway certificates to specific path with key.

Usage `alconfset`: `alconfset mgmt_gw_certificates <certificate_path>:<key_path>`
Example: `alconfset mgmt_gw_certificates </path/xyz.crt:/path/abc.key>`

Usage `alconfget`: `alconfget mgmt_gw_certificates`

disable_backhaul_security:

Description: To disable the backhaul security.

Usagealconfset: alconfset disable_backhaul_security

Usagealconfset: alconfset disable_backhaul_security

802.11n Supported fields.

Below are the fields/parameters are supported for 802.11n protocol.

Ldpc:

Description: Low density parity check(LDPC) is iterative error checking code.

It can be set either enable(1) or disable(0).

Usagealconfset: alconfset ldpc <<if-name>> <<enabled or disabled>>

Usagealconfget: alconfget ldpc <<if-name>>

Smps:

Description: Spatial Multiplexing Power Save (SMPS)mode is a mechanism to conserve power in 802.11n implementation. It can be set either to static, dynamic or disabled.

Usagealconfset: alconfset smps <<if-name>> <<disabled or static or dynamic>>

Usagealconfget: alconfget smps <<if-name>>

tx_stbc:

Description: STA capability of transmitting PPDU using STBC (Space Time Block Coding) . A technique used to send multiple copies of the data streams across a number of antennas and to exploit the various received versions of the data to improve the reliability of data-transfer.

It can be set either to enable or disable.

Usagealconfset: alconfset tx_stbc <<if-name>> <<enabled or disabled>>

Usagealconfget: alconfget tx_stbc <<if-name>>

rx_stbc:

Description: STA capability of receiving PPDU using STBC (Space Time Block Coding). A technique used to send multiple copies of the data streams across a number of antennas and to exploit the various received versions of the data to improve the reliability of data-transfer.

It can be set either to enable or disable.

Usagealconfset: alconfset rx_stbc <<if-name>> <<enabled or disabled>>

Usagealconfget: alconfget rx_stbc <<if-name>>

delayed_ba:

Description: indicate STA support of Delayed BlockAck. It can be set either to enable(1) or disabled(0).

Usage alconfset: alconfset delayed_ba <<if-name>> <<enabled or disabled>>

Usage alconfget: alconfget delayed_ba <<if-name>>

Intolerant:

Description: It can be set either to enabled(1) or disabled(0).

Usage alconfset: alconfset intolerant <<if-name>> <<enabled or disabled>>

Usage alconfget: alconfget intolerant <<if-name>>

lsig_txop:

Description: indicate support for Legacy-Signal (L-SIG) protection mechanism.

Can be set either to enabled(1) or disabled(0).

Usage alconfset: alconfset lsig_txop <<if-name>> <<enabled or disabled>>

Usage alconfget: alconfget lsig_txop <<if-name>>

gi_20:

Description: Guard Interval is intended to avoid signal loss from multipath effects which helps to overcome overlapping transmission. Guard interval with 20Mhz can be set to either long, short or auto.

Usage alconfset: alconfset gi_20 <<if-name>> <<long or auto or short>>

Usage alconfget: alconfget gi_20 <<if-name>>

gi_40:

Description: Guard Interval is intended to avoid signal loss from multipath effects which helps to overcome overlapping transmission. Guard interval with 40Mhz can be set to either long, short or auto.

Usage alconfset: alconfset gi_40 <<if-name>> <<long or auto or short>>

Usage alconfget: alconfget gi_40 <<if-name>>

dsss_cck_40:

Description: DSSS check mode in 40 MHz can be set either to allow or deny. Which says, the stream of information to be transmitted has to be divided or not.

Usage alconfset: alconfset dsss_cck_40 <<if-name>> <<allow or deny>>

Usage alconfget: alconfget dsss_cck_40 <<if-name>>

ht_bandwidth:

Description: Hardware threshold bandwidth can be set to 40+, 40- and 20 MHz channel.

Usage alconfset: alconfset ht_bandwidth <<if-name>> <<Bandwidth>>

Usage alconfget: alconfget ht_bandwidth <<if-name>>

max_amsdu_len:

Description: Aggregated Mac Service Data Unit length can be set either to 1(7935) and 0(3839).

Usage alconfset: alconfset max_amsdu_len <<if-name>> << 0=>3839 / 1=>7935>>

Usage alconfget: alconfget max_amsdu_len <<if-name>>

ampdu_enable:

Description: Aggregated Mac protocol data unit(AMPDU) can be enabled by setting it to 1 and disabled by 0.

Usage alconfset: alconfset ampdu_enable <<if-name>> <<ampdu_enable integer: 0 or 1>>

Usage alconfget: alconfget ampdu_enable <<if-name>>

max_ampdu_len:

Description: to set the Maximum length of Aggregated Mac protocol data unit(AMPDU).

Usage alconfset: alconfset max_ampdu_len <<if-name>> <<eg: 64KB>>

Usage alconfget: alconfget max_ampdu_len <<if-name>>

Gfmode:

Description: STA is capable of receiving HT Greenfield PPDU if it is set to 1 and vice versa for 0.

Usage alconfset: alconfset gfmode <<if-name>> <<enabled or disabled>>

Usage alconfget: alconfget gfmode <<if-name>>

802.11ac Supported fields:

max_mpdu_len:

Description: MAC frames in 802.11ac may have one of three lengths: 3,895 bytes, 7,991 bytes, or 11,454 bytes.

Usage alconfset: alconfset max_mpdu_len <<if-name>> <<max_mpdu_len integer value>>

Usage alconfget: alconfget max_mpdu_len <<if-name>>

supported_channel_width:

Description: Supported channel width can be set to either 0, 1 or 2.

Usage alconfset: alconfset supported_channel_width <<if-name>>
<<supported_channel_width integer value>>

Usage alconfget: alconfget supported_channel_width <<if-name>>

rx_ldpc:

Description: Transmitter can receive LDPC-encoded frames if it is enabled and vice versa.

Usage alconfset: alconfset rx_ldpc <<if-name>> <<enabled or disabled>>

Usage alconfget: alconfget rx_ldpc <<if-name>>

gi_80:

Description: Transmitter can receive frames transmitted using guard interval with 80MHz. It can be set to Auto, long or short.

Usage alconfset: alconfset gi_80 <<if-name>> <<long or auto or short>>

Usage alconfget: alconfget gi_80 <<if-name>>

gi_160:

Description: Transmitter can receive frames transmitted using guard interval with 160MHz. It can be set to Auto, long or short.

Usage alconfset: alconfset gi_160 <<if-name>> <<long or auto or short>>

Usage alconfget: alconfget gi_160 <<if-name>>

vtx_stbc:

Description: STA capability of transmitting PPDU using STBC (Space Time Block Coding) for interfaces with 802.11ac support. It can be set either to enable or disable

Usage alconfset: alconfset vtx_stbc <<if-name>> <<enabled or disabled>>

Usage alconfget: alconfget vtx_stbc <<if-name>>

vr_x_stbc:

Description: STA capability of receiving PPDU using STBC (Space Time Block Coding) for interfaces with 802.11ac support. It can be set either to enable or disable

Usagealconfset: `alconfset vrx_stbc <<if-name>> <<enabled or disabled>>`

Usagealconfget: `alconfget vrx_stbc <<if-name>>`

su_beamformer_cap:

Description: Single-User (SU) Beamformer is set to yes(1), when transmitter is capable of operating as a single-user beamformer. It can be set either to 1 or 0.

Usagealconfset: `alconfset su_beamformer_cap <<if-name>> <<yes or no>>`

Usagealconfget: `alconfget su_beamformer_cap <<if-name>>`

su_beamformee_cap:

Description: Single-User (SU) Beamformee is set to yes(1), when transmitter is capable of operating as a single-user beamformee. It can be set either to 1 or 0.

Usagealconfset: `alconfset su_beamformee_cap <<if-name>> <<yes or no>>`

Usagealconfget: `alconfget su_beamformee_cap <<if-name>>`

***beamformee_sts_count:**

Description:

Usagealconfset: `alconfset beamformee_sts_count <<if-name>> <<beamformee_sts_count integer value>>`

Usagealconfget: `alconfget beamformee_sts_count <<if-name>>`

***sounding_dimensions:**

Description:

Usagealconfset: `alconfset sounding_dimensions <<if-name>> <<sounding_dimensions integer value>>`

Usagealconfget: `alconfget sounding_dimensions <<if-name>>`

mu_beamformer_cap:

Description: Multi-user(MU) beamformer is set to yes(1), when transmitter is capable of operating as a multi-user beamformer.

Usagealconfset: `alconfset mu_beamformer_cap <<if-name>> <<yes or no>>`

Usagealconfget: `alconfget mu_beamformer_cap <<if-name>>`

mu_beamformee_cap:

Description: Multi-user(MU) beamformee is set to yes(1), when transmitter is capable of operating as a multi-user beamformee.

Usagealconfset: `alconfset mu_beamformee_cap <<if-name>> <<yes or no>>`

Usagealconfget: `alconfget mu_beamformee_cap <<if-name>>`

vht_txop_ps:

Description: An AP can set this bit to 1 to enable power save operations during a VHT transmission burst, or 0 to disable them. Stations associating with a network will set this bit to 1 to indicate the capability is enabled or 0 if it is disabled.

Usage alconfset: alconfset vht_txop_ps <<if-name>> <<yes or no>>

Usage alconfget: alconfget vht_txop_ps <<if-name>>

htc_vht_cap:

Description: This value is set to yes (1) to indicate that the transmitter is capable of receiving the VHT-variant HT Control field.

Usage alconfset: alconfset htc_vht_cap <<if-name>> <<yes or no>>

Usage alconfget: alconfget htc_vht_cap <<if-name>>

rx_ant_pattern_consistency:

Description: Set to yes(1), if rx antenna doesn't change and set to No(0), if rx antenna change during the lifetime of current association.

Usage alconfset: alconfset rx_ant_pattern_consistency <<if-name>> <<yes or no>>

Usage alconfget: alconfget rx_ant_pattern_consistency <<if-name>>

tx_ant_pattern_consistency:

Description: Set to yes(1), if tx antenna doesn't change and set to No(0), if tx antenna change during the lifetime of current association.

Usage alconfset: alconfset tx_ant_pattern_consistency <<if-name>> <<yes or no>>

Usage alconfget: alconfget tx_ant_pattern_consistency <<if-name>>

vht_oper_bandwidth:

Description: For either 20 MHz or 40 MHz operation, the Channel Width field is set to 0. 80 MHz operation sets this value to 1. Because it is necessary to distinguish the 160 MHz channel width (a value of 2) from the 80+80 MHz channel structure (a value of 3), they receive separate values.

Usage alconfset: alconfset vht_oper_bandwidth <<if-name>> <<supported_channel_width integer value>>

Usage alconfget: alconfget vht_oper_bandwidth <<if-name>>

seg0_center_freq:

Description: This fields are used only with 80 and 160 MHz operation, to transmit the center channel frequency of the BSS

Usage alconfset: alconfset seg0_center_freq <<if-name>> <<supported_channel_width integer value>>

Usage alconfget: alconfget seg0_center_freq <<if-name>>

seg1_center_freq:

Description: This field is used only with 80+80 MHz operation, and is used to transmit the center channel frequency of the second segment.

Usage alconfset: alconfset seg1_center_freq <<if-name>> <<supported_channel_width integer value>>

Usage alconfget: alconfget seg1_center_freq <<if-name>>

7 Debug Commands for Mesh

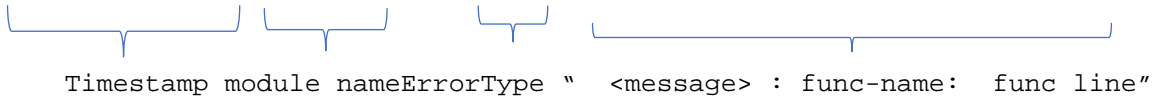
7.1 Using log information

Mesh code can print important log messages of events and this can be displayed by doing the following

- dmesg
- cat /overlay/md_syslog.txt

The format of the mesh logs are given below

```
[ 76.510000] MESH_AP: INFO: message-text <line#>
```



Example

```
[220250.070000] MESHAP: INFO: Number of parent bening added = 4:  
_add_parents_list_to_heartbeat: 98
```

7.2 Printing important internal mesh data structures

```
cat /proc/net/meshap/mesh/debug_ds_info
```

```
*****  
v_wlan0  
*****  
current_essid_length : 22  
current_essid: StructuredMesh_v_wlan0  
current_phy_mode: 802_11_A  
iw_mode: MASTER  
use_type : AP  
beacon_conf_complete : 1  
channel_count: 255  
beacon_interval : 100  
rts_threshold: 2347  
frag_threshold: 2346  
tx_power : 63  
sec_chan_offset : 0  
scan_state : -1  
country code : 0  
fsm_state: STOPPED  
mesh_state : RUNNING  
ht_capab_info: 0x0  
ht_opera : 0x0  
vht_capab_info: 0x0  
scan_start_time : 0  
scan_completion_time : 0  
beacon_vendor_info_length: 0  
beacon_vendor_info:  
current_bssid:  
connected: 0  
a_mpdu_params: 0x0  
W_Channel-number: 52  
W_Channel-frequency(HZ): 5260
```

```
W_Channel-state_flags: 0
W_Channel-max_power: 23
W_Channel-min_power: 0
vht_op_info_chwidth: 0
vht_op_info_chan_center_freq_seg0_idx: 0
vht_op_info_chan_center_freq_seg1_idx: 0
vht_basic_mcs_set : 0
```

<repeat the same for each wlan interface>

7.3 Printing queue stats for mesh modules

```
cat /proc/net/meshap/mesh/queue_packet_stats
```

```
Entry_pool_current_count : 2048
Core_packet_pool_current_count: 2048
Tx_packet_pool_current_count: 2048
dot1p_queue_total_packet_added_count : 130
dot1p_queue_total_packet_removed_count : 130
dot1p_queue_current_packet_count: 0
tx_queue_total_packet_added_count : 527
tx_queue_total_packet_removed_count: 527
tx_queue_current_packet_count : 0
imcp_queue_total_packet_added_count: 130
imcp_queue_total_packet_removed_count: 130
imcp_queue_current_packet_count : 0
skb_queue_total_packet_added_count: 130
skb_queue_total_packet_removed_count : 130
skb_queue_current_packet_count: 0
upstack_queue_total_packet_added_count : 13
upstack_queue_total_packet_removed_count: 13
upstack_queue_current_packet_count: 0
core_packet_type_alloc_heap: 2048
```

7.4 Printing thread stats for mesh modules

This stats basically checks if all threads are alive and kicking

```
cat /proc/net/meshap/mesh/thread_stats
Ap_thread [1]: 251
Tx_thread [1]: 659
process_skb_thread: 381
imcp_thread: 251
upstack_thread : 39
Ap_sta_monitor_thread : 1
Ap_dot11i_thread: 1
Md_watch_thread: 26
Mesh_Thread: 315
Mesh_hb_thread : 29
Uplink_scan_thread: 0
Lfr_thread: 0
Scan_thread: 157
Radar_dca_thread: 0
Dot1x_timer_thread: 875
dBmTestRx_thread: 1
dBmTestTx_thread: 1
Sip_cleanup_thread: 0
```

7.5 Printing drop stats for mesh modules

```
cat /proc/net/meshap/mesh/tx_rx_drop_pkt_stats

access_point_process_data_pkt: 130
access_point_process_pkt: 130
rx_up_stack_pkt: 13
tx_pkt: 23
tx_imcp_pkt : 78
mip_xmit_tx : 19
rx_master_mode_mgmt_probe_req_frame : 410
rx_infra_mode_mgmt_frame: 32790

*****Embeddedstudios and md-mac80211 packet drop stats***
packet_drop[27 ]_count: 19
packet_drop[34 ]_count: 23
packet_drop[55 ]_count: 3
packet_drop[59 ]_count: 13
packet_drop[102]_count: 28
packet_drop[104]_count: 33
packet_drop[115]_count: 8
packet_drop[123]_count: 316

mac80211 stats:
WM_mgmt_frame_processed_in_meshap_hook: 94
mgmt_frame_sent_to_userspace : 313
unusable_frame_free: 14
non_consume_data_frame_drop: 113
tx_frame_not_sent_to_monitor_iface: 441
free_tx_skb : 13108
iface_work_skb: 32705
rx_monitor_drop: 26408
```

```
v_wlan0 stats:  
rx_mgmt : 90
```

```
wlan3 stats:  
rx_mgmt : 7272
```

```
wlan2 stats:  
rx_mgmt : 226
```

```
wlan1 stats:  
rx_mgmt : 25518  
rx_data : 111
```

```
wlan0 stats:  
rx_mgmt : 94
```

7.6 Dumping all stats

Use a single command called `gather_info` which will dump all the above and other relevant stats for developers to analyze and debug issue

Usage: `gather_info`

8 Procedure for Sysupgrade

It is used to upgrade the firmware.

Note: Images with ".bin" file extension is used for sys upgrade.

Steps:

1. Build target for sys upgrade(Build .bin file).

IMX:

```
sudo make imx6_md_sysupgrade MAC=04:F0:21:33:0E:DD MD_CONFIG=MD6455-LLJAXX_NO_ETH1.conf RMAC1=04:F0:21:25:61:01 RMAC2=04:F0:21:25:61:02 RMAC3=04:F0:21:25:61:03 RMAC4=04:F0:21:25:61:04 RMAC5=04:F0:21:25:61:05 VLAN1=04:F0:21:25:61:06 VLAN2=04:F0:21:25:61:07 VLAN3=04:F0:21:25:61:08 BOARD=nor FLASHSIZE=16 VERSION=1.1.11
```

CNS:

```
sudo make cns3xxx_md_sysupgrade MAC=30:14:4A:EA:88:53 MD_CONFIG=MD4455-AAIA_NO_ETH1.conf RMAC1=30:14:4A:AE:11:01 RMAC2=30:14:4A:AE:11:02 RMAC3=30:14:4A:AE:11:03 RMAC4=30:14:4A:AE:11:04 RMAC5=30:14:4A:AE:11:05 VLAN1=30:14:4A:AE:11:06 VLAN2=30:14:4A:AE:11:07 VLAN3=30:14:4A:AE:11:08 BOARD=nor FLASHSIZE=16 VERSION=1.1.11
```

2. Copy image(.bin) to "/tmp/" folder of the device.

[Scp image to the device "/tmp/" folder]

To copy from local machine use:

```
scp <file name> username@<IP address>:/tmp/
```

example:

```
scp imx_md_1.1.11_04_F0_21_33_0E_DD.bin root@172.17.200.7:/tmp/
```

Note: Before doing scp make sure that password is set for the device console.

****How to set password in console****

```
root@OpenWrt123:/# passwd
Changing password for root
New password:
Bad password: too short
Retype password:
Password for root changed by root
```

3. In device console enter the following command.

```
sysupgrade /tmp/<filename>
```

example:

```
sysupgrade /tmp/imx_md_1.1.11_04_F0_21_33_0E_DD.bin
```

This will initiate sysupgrade and it will preserve all configuration files in **/etc/**

If you do not want to save configuration files over reflash, use "-n" option.

```
sysupgrade -n /tmp/imx_md_1.1.11_04_F0_21_33_0E_DD.bin
```

Below are the upgrade options for sysupgrade.

Usage: /tmp/sysupgrade [<upgrade-option>...] <image file or URL>

```
/tmp/sysupgrade [-q] [-i] <backup-command> <file>
```

upgrade-option:

```
-d <delay> add a delay before rebooting
-f <config> restore configuration from .tar.gz (file or url)
-i interactive mode
-c attempt to preserve all changed files in /etc/
-n do not save configuration over reflash
-p do not attempt to restore the partition table after flash.
-T | --test Verify image and config .tar.gz but do not actually flash.
-F | --forceFlash image even if image checks fail, this is dangerous!
-q less verbose
-v more verbose
-h | --help display this help
```

backup-command:

```
-b | --create-backup <file>
create .tar.gz of files specified in sysupgrade.conf
```

then exit. Does not flash an image.

If file is '-', i.e. stdout, verbosity is set to 0 (i.e. quiet).

```
-r | --restore-backup <file>
```

restore a .tar.gz created with sysupgrade -b then exit. Does not flash an image.

If file is '-', the archive is read from stdin.

```
-l | --list-backup
```

list the files that would be backed up when calling sysupgrade -b. Does not create a backup file.