sales@candelatech.com
support@candelatech.com
+1 (360) 380-1618 [PST, GMT -8]

Ath10k Candela Technologies (CT) Firmware User Guide

This documents some of the features of the Ath10k-CT Firmware and driver.

Disable firmware debugging.

The ath10k-ct firmware and driver print out firmware debuglog messages in the kernel logs (dmesg). It looks something like this:

```
[3538979.051973] ath10k_pci 0000:07:00.0: ath10k_pci ATH10K_DBG_BUFFER:
[3538979.051998] ath10k: [0000]: 07F90DB9 0BFC4C21 00000BAC 00000002 07F9100C 0BFC4C21 00000BF2 00000002
[3538979.052001] ath10k_pci 0000:07:00.0: ATH10K_END
```

This can be used to debug firmware issues, but in systems that are not having any trouble, users may want to disable it. To do so, set flag 0x10000000 in the ath10k debug_level setting. This can be done by echoing a value to the /sys/kernel/debug/ieee80211/phy0/ath10k/debug_level variable debugfs file, or it can be set as a modprobe variable.

Disable/tune firmware station-kickout.

The ath10k-ct driver allows adjusting the firmware's station-kickout logic. This should work on both standard and CT firmware, but it has only been tested on ath10k-ct firmware as far as we know. You may want to disable the station-kickout code and let higher levels make the decision.

```
# Disable station kick-out logic in firmware
echo 0x100100000000 > /sys/kernel/debug/ieee80211/phy0/ath10k/ct_special

# Set station kick-out back to default (20 * 16, or 320 decimal)
echo 0x100100000140 > /sys/kernel/debug/ieee80211/phy0/ath10k/ct_special
```

Set multicast, broadcast, beacon tx rates.

By default, multicast, broadcast, and beacons are sent at 6Mbps fixed encoding rate. The ath10k-ct driver allows configuring these rates. This probably works on all firmware, but has only been tested on ath10k-ct wave-1 firmware in our lab to date.

```
# See help info on this option
]# cat /sys/kernel/debug/ieee80211/wiphy0/ath10k/set_rates

This is to set fixed bcast, mcast, and beacon rates.  Normal rate-ctrl
is handled through normal API using 'iw', etc.
To set a value, you specify the dev-name, type, band and rate-code:

types: bcast, mcast, beacon
bands: 2, 5, 60
rate-codes: 0x43 1M, 0x42 2M, 0x41 5.5M, 0x40 11M, 0x3 6M, 0x7 9M, 0x2 12M, 0x6 18M, 0x1 24M, 0x5 36M, 0x0 48M, 0x4

# Set beacons to use 18Mbps encoding
echo "wlan0 beacon 2 0x6" >  /sys/kernel/debug/ieee80211/wiphy0/ath10k/set_rates
```

For the more adventurous, you can figure out HT and VHT rate codes by looking at how the ath10k driver builds/decodes them. I am not sure HT or VHT rates can be used successfully for these frames currently, however.

Setting specific encoding rates.

Users may wish to set specific rates for a variety of reasons, including regulatory testing. CT firmware has some additional features to give you more control. Stock firmware will support some of this as well. These commands were tested on a recent LEDE image running latest ath10k-CT driver and firmware as of October 10, 2017. An additional patch was added to the LEDE mac80211 backports package to enable setting a single rate without generating an error return value.

Enable/disable specific bandwidths for TX frames.

This is a CT specific feature using the 'ct-special' API. This only works with 10.1 (wave-1) CT firmware currently. If you want to use this option, configure the bandwidth first, and then set a rate. **The bandwidth constraint will not take effect until you set the rate using the 'iw' commands below**:

```
# Configure for 20Mhz only (disable 80, 40):
echo 0xE00000006 > /sys/kernel/debug/ieee80211/phy0/ath10k/ct_special
```

```
# Configure for 40Mhz only (disable 80, 20).  Please note than legacy
# rates cannot ever use 40Mhz.
echo 0xE00000005 > /sys/kernel/debug/ieee80211/phy0/ath10k/ct_special
```

```
# Configure for 80Mhz only (disable 40, 20).  Please note that HT
# and legacy rates cannot ever use 80Mhz.
echo 0xE00000003 > /sys/kernel/debug/ieee80211/phy0/ath10k/ct_special
```

```
# Configure for any available Mhz (default)
echo 0xE00000000 > /sys/kernel/debug/ieee80211/phy0/ath10k/ct_special
```

Configure a specific TX rate.

These commands probably work fine on stock firmware, and on 10.4 (wave-2) CT firmware as well.

```
# NOTE:  VHT rates are not normally available on the 2.4Ghz band without additional
#        kernel modifications.
# Set for vht-rateset, MCS 0, NSS 1:
iw dev wlan0 set bitrates legacy-5 ht-mcs-5 vht-mcs-5 1:0
# Set for vht-rateset, MCS 0, NSS 3:
iw dev wlan0 set bitrates legacy-5 ht-mcs-5 vht-mcs-5 3:0
# Set for vht-rateset, MCS 9, NSS 3:
iw dev wlan0 set bitrates legacy-5 ht-mcs-5 vht-mcs-5 3:9
# The ath10k 9880 3x3 NIC supports up to MCS 9, NSS 3.
```

```
# Set for ht-rateset, MCS 0, nss 1:
iw dev wlan0 set bitrates legacy-5 ht-mcs-5 0 vht-mcs-5
# For HT MCS 8, nss2:
iw dev wlan0 set bitrates legacy-5 ht-mcs-5 8 vht-mcs-5
#  The ath10k 9880 3x3 NIC supports ht-mcs 0-23 settings.
```

```
# Set for legacy (a/g) 6Mbps
iw dev wlan0 set bitrates legacy-5 6 ht-mcs-5 vht-mcs-5
# For legacy 54Mbps
iw dev wlan0 set bitrates legacy-5 54 ht-mcs-5 vht-mcs-5
#  Available legacy rates for the 2.4Ghz band are: 1, 2, 5.5, 11, 6, 9, 12, 18, 24, 36, 48, 54
#  Available legacy rates for the 5Ghz band are: 6, 9, 12, 18, 24, 36, 48, 54
```

```
# Set back to default rates.
iw dev wlan0 set bitrates
```

Verify tx/rx rates.

In order to verify that the rate configuration is working as expected, you can use an RF sniffer, but you may also be able to just dump the station info to see the current rateset. The TX rate should match the expected values.

```
iw dev wlan0 station dump
```

WMI firmware keepalive (10.1 wave-1 firmware only at this time)

The ath10k-CT driver and firmware support a WMI keepalive message. Once the firmware receives a keepalive message, then it will assert in the future if it does not receive a keepalive within about 8 seconds. Often this crash is better than an indefinite hang. But, some systems may not be able to reliably send keepalives often enough, so you may wish to disable the keepalive, or set it to a larger value. You can do this with the 'ct_special' debugfs file. This setting will be saved through firmware restarts, but a reboot or driver reload will initialize the setting back to defaults.

```
# Disable WMI timeout assert in the firmware:
echo 0xCFFFFFFFF > /sys/kernel/debug/ieee80211/phy0/ath10k/ct_special

# Set to 8 second timeout
echo 0xC00001F40 > /sys/kernel/debug/ieee80211/phy0/ath10k/ct_special
```

Enable AMSDU for IBSS connections

Wave-1 (9880, etc) hardware appears to have bugs with AMSDU frames in IBSS interfaces: It zeros the BSSID field, which causes the linux stack to discard the frames on receipt. I am not sure about wave-2 (9984, etc), but some brief testing shows that tx throughput is higher when I disable AMSDU on it as well. Possibly due to other causes, however. Either way, I have disabled AMSDU on wave-2 by default now (Feb 21, 2018) as well. Users can re-enable AMSDU with the command below. I would like to hear of your results if you try this and it works well for you.

```
# Enable AMSDU for IBSS connections on wave-1 and wave-2.
echo 0x500000001 > /sys/kernel/debug/ieee80211/phy0/ath10k/ct_special

# Disable AMSDU for IBSS connections on wave-1 and wave-2 (default)
echo 0x500000000 > /sys/kernel/debug/ieee80211/phy0/ath10k/ct_special
```

Enable IBSS CCA hack

Wave-2 firmware configures some CCA logic differently for IBSS tx queues. At one point, I suspected that this decreased performance, but futher testing was inconclusive. If you want to enable this feature (in wave-2 firmware built after March 16, 2018), you can use the commands below to do so.

```
# Treat IBSS non-beacon txqueues same as normal queues
echo 0x1200000001 > /sys/kernel/debug/ieee80211/phy0/ath10k/ct_special

# Treat IBSS beacon txqueues same as normal queues
echo 0x1200000002 > /sys/kernel/debug/ieee80211/phy0/ath10k/ct_special

# Treate all IBSS txqueues same as normal queues
echo 0x1200000003 > /sys/kernel/debug/ieee80211/phy0/ath10k/ct_special
```

Enable TX-HANG fixup to try COLD resets

Wave-2 firmware now (Feb 22, 2018) supports attempting cold resets in the tx-hang work-around code if there is more than one WARM reset requested within 15 seconds. This feature is disabled by default (only WARM resets will be used). Possibly cold resets will help in some cases?

```
# Enable COLD resets in tx-hang work-around.
echo 0x1100000001 > /sys/kernel/debug/ieee80211/phy0/ath10k/ct_special

# Disable COLD resets in tx-hang work-around.
echo 0x1100000000 > /sys/kernel/debug/ieee80211/phy0/ath10k/ct_special
```

Enable getting CFR data (at least for probe-response ACKs) (9984, 9888 only it seems)

For recent wave-2 firmware for 9984 and (maybe) 9888, you may configure it to return CFR data for probe-response frames. This does not work with 4019 or 9980 chipsets it seems. The data is delivered in multiple chunks over WMI. See ath10k_wmi_event_csi_mesg() in the ath10k-ct driver.

```
# Enable CFR data reporting.
echo 0xD00000001 > /sys/kernel/debug/ieee80211/phy1/ath10k/ct_special

# Disable CFR data reporting.
echo 0xD00000000 > /sys/kernel/debug/ieee80211/phy1/ath10k/ct_special
```

Disable beamforming features with module options (10.4 wave-2 firmware only at this time)

The ath10k-CT driver now (Feb 21, 2018) supports disabling advertisement of the beamform feature flags. This may be used for testing purposes and/or to work-around bugs related to txbf. This should work with ath10k-ct firmware or stock firmware since it is only a driver change.

```
# Disable MU txbf features.  Add option like this in your modprobe.d file(s)
options ath10k_core nobeamform_mu=1

# Disable SU txbf features.  Add option like this in your modprobe.d file(s)
options ath10k_core nobeamform_su=1
```

Enable receiving TXBF frames in the driver (wave-2 only)

For wave-2 firmware built after Nov 7, 2017, you can enable the receipt of txbf frames when you put the radio into monitor mode and also when you just use the ct_special command to enable the txbf reporting logic. Frames that are consumed by the firmware (ones directed at local peers), will not be delivered in this way. They can only be received using the txbf_cv WMI message, which is also enabled/disabled with this same command. See ath10k_wmi_event_txbf_cv_mesg in the ath10k-ct driver for details.

```
# Enable txbf frames and txbf_cv WMI messages to be sent to the driver.
echo 0xF00000001 > /sys/kernel/debug/ieee80211/phy1/ath10k/ct_special

# Disable txbf frames and txbf_cv WMI messages from being sent to the driver (default).
echo 0xF00000000 > /sys/kernel/debug/ieee80211/phy1/ath10k/ct_special
```

Enable rate-ctrl txbf probing (wave-2 only)

If you want to enable rate-ctrl txbf probing (which is needed to get a 2x2 AP to send txbf probes to a 2x2 station, for instance), enable it like this. This requires wave-2 CT firmware and kernel built after Dec 7, 2017.

```
# Enable rc-txbf-probe
echo 0x100400000001 > /sys/kernel/debug/ieee80211/wiphy1/ath10k/ct_special

# Disable rc-txbf-probe
echo 0x100400000000 > /sys/kernel/debug/ieee80211/wiphy1/ath10k/ct_special
```

As of April 5, 2018: To force NDP probes to go out even when the firmware otherwise thinks that txbf is not necessarily optimal, set the bit 0x2 for the rc-txbf-probe command:

```
# Enable rc-txbf-probe with extra logic to force NDP to go out more often
echo 0x100400000003 > /sys/kernel/debug/ieee80211/wiphy1/ath10k/ct_special
```

Enable receiving all management frames on the host (wave-2 only)

With ath10k-ct wave-2 firmware from Nov 8, 2017 or later, you can set a flag to cause all received management frames to be sent to the host driver. The one big exception is txbf messages: They are

consumed by the txbf_cv logic and so the only way you can get notification of them is by using the txbf_cv WMI message (see above). RX filters are not modifed as part of this setting.

```
# Enable receiving all mgt frames
echo 0x1000000001 > /sys/kernel/debug/ieee80211/phy1/ath10k/ct_special

# Disable receiving all mgt frames
echo 0x1000000000 > /sys/kernel/debug/ieee80211/phy1/ath10k/ct_special
```

Set the SU or MU Sounding timer in ms (wave-2 only)

With ath10k-ct driver code from Nov 7, 2017 or later, you can now set the SU and MU sounding frame timer. Minimum is 0, maximum is 500. This should work on stock firmware as well as CT firmware, but only tested on CT firmware. The default SU timer is 100ms, and the default MU timer is 40ms.

```
# Set MU sounding timer to 16ms
echo 0x100300000010 > /sys/kernel/debug/ieee80211/phy1/ath10k/ct_special

# Set SU sounding timer to 32ms
echo 0x100200000020 > /sys/kernel/debug/ieee80211/phy1/ath10k/ct_special

# Set values to 0 and restart the firmware (or just reload the entire driver),
# and values will be automatically set back to firmware defaults.
```

Force sending sounding frames (wave-2 only)

With ath10k-ct driver and firmware code from Nov 16, 2017 or later, you can now use the ct_special API to make generic 'fwtest' configuration commands. And, one of them can be used to force sending a sounding frame to a specific peer. The API is a bit complicated: First, the 'ID' will have '0xFF0000' set as a mask to identify this as being a 'fwtest' command. The low bits of the ID will be the fwtest ID. The low 32 bits of the ct_special command are the value to set. To send a sounding frame, you need ID 74, and the value is the 'aid' of the peer. The driver should print the aid when a station associates, and there are other ways to find it as well. If you have a single station, then aid == 1. In addition, the ID=74 command in stock firmware will not actually send an sounding frame. In order to be somewhat backwards compatible, you have to set bit 0x80000000 in the value to get the new behaviour. So, to force a sounding frame to the first station, you can use this command:

```
# Force sounding frame to station 1.
echo 0xFF004A80000001 > /sys/kernel/debug/ieee80211/phy1/ath10k/ct_special
```

While testing this, I noticed that the TX logic in the NIC/Firmware would hang if I ran this command in a fast loop. So, the firmwre now has protection where it will ignore any of these commands that are closer than 25ms apart. Possibly it will still be unstable in long runs even with 25ms spacing...time and testing will tell.

Further testing shows this is fragile at best. It appears to fail entirely on 4019 radios similar to how it failed on 9984 at high speeds. This feature will be compiled out of any diet (trimmed) CT firmware builds, at least until the problems are understood and resolved.

Enable sw-rts (wave-2 only)

With ath10k-ct driver and firmware code from Feb 17, 2021 or later, you can now enable/disable software control of sending rts frames. Possibly this works on stock firmware too since it uses an upstream API.

```
# Enable sw-rts.
echo 0xFF00D480000001 > /sys/kernel/debug/ieee80211/phy1/ath10k/ct_special
```

Set MU RTS duration (wave-2 only)

With ath10k-ct driver and firmware code from Feb 17, 2021 or later, you can now set the MU RTS duration in usec units. Default is 0x300. Set to zero and restart firmware to go back to defaults. The low 16 bits of the number below is the duration. This *may* work, untested. See SU section below.

```
# Set MU RTS duration
echo 0xFF00868000007F > /sys/kernel/debug/ieee80211/phy1/ath10k/ct_special
```

Set SU RTS duration (wave-2 only)

With ath10k-ct driver and firmware code from Feb 17, 2021 or later, you can now set the SU RTS duration in usec units. Default is 0 (auto-calculate). The low 16 bits of the number below is the duration. This does not work as hoped, I think the hardware over-writes the duration before putting the CTS frame on air.

```
# Set SU RTS duration
echo 0xFF00E78000007F > /sys/kernel/debug/ieee80211/phy1/ath10k/ct_special
```

Set tx-antenna pattern on a per-peer basis (wave-2 firmware only at this time)

As of June 8, 2018, wave-2 firmware and 4.9 and higher kernels support setting the tx-antenna antenna field in the tx descriptor on a per peer basis. I am waiting on feedback from a customer about whether it actually works or not, but the idea is that it should twiddle some DIO pins on the chip as packets are transmitted, allowing smart antenna setups to steer packets to specific peers.

We use the ct_special API to set this. We need the peer-ID as well, and the drivers include a new debugfs file called 'peers' which list the peer-id for each peer. The value sent to ct_special is calculated as:

```
peer_id << 16 | antenna_pattern
```

Currently the antenna_pattern is limited to 5 bits, but that might change as needed.

```
[root@lf0313-6477 lanforge]# cat /debug/ieee80211/wiphy1/ath10k/peers
04:f0:21:95:25:b0  vdev-id: 0  peer-ids: 1
```

```
# Set peer-id 0x1 to use antenna pattern 0x1c
echo 0x130001001c > /sys/kernel/debug/ieee80211/phy0/ath10k/ct_special

# Set peer-id 0x1 to use the default antenna pattern (which is 0x5 on my radio)
echo 0x1300010000 > /sys/kernel/debug/ieee80211/phy0/ath10k/ct_special
```

Due to some register setting commands which I do not fully understand, it may be that once you set the antenna mask for any peer, the hardware will need to be reset to go back to the default behaviour. Also, the driver will not cache the value you set through ct_special, so you need to (re)set it each time the firmware restarts or the peer is (re)added.

Set board power ctl table from board.bin (cal data)

As of April 4, 2019, wave-1 9880/9882 firmware and 4.9 and higher kernels support overriding the eeprom power ctl table using the values loaded from the board.bin or calibration data. The reason this may be needed is that on chips with OTP enabled (WLE900VX, WLE600VX, etc) the OTP will over-ride the values from the board.bin file. With this new feature, users can specifically override the OTP values. Do NOT use this unless you are certain this is a good idea for your use case. The only way to get back to the defaults once you enable this command is to reload the driver (which will re-do the OTP).

```
# Enable power ctl table override
echo 0x100500000001 > /sys/kernel/debug/ieee80211/phy0/ath10k/ct_special
```