

## Ath10k Candela Technologies (CT) Firmware Bug Reports

Please report ath10k-ct driver and/or firmware bugs on the [ath10k-ct github project web page issue tracker](#). A good bug report should contain a description of how the bug is reproduced, the network configuration used, hardware used (especially the NIC chipset), how often the bug is reproduced, firmware version, and appropriate logs, and so forth. Please put some effort into good bug reports. It will help me fix the problem efficiently, and poor bug reports will most likely be ignored or we will waste lots of time while I ask enough questions to update the bug report. Here is how you can get some of the useful information:

- **dmesg logs can often show firmware crash dumps and related kernel issues.** It is a good place to start. Some problems are not actually firmware related, and you may need to enable driver logs to make progress. To enable more verbose dmesg logs, you may need to change the ath10k debug\_level. Firmware related bugs should be sure to disable this flag: NO-FW-DBGLOG. You can adjust this at runtime through debugfs, and `0xc0000032` is a good value to start with:

```
cat /sys/kernel/debug/ieee80211/wiphy1/ath10k/debug_level
Current debug level: 0xc0000030
```

To change debug level, set value adding up desired flags:

```
PCI:          0x1
WMI:          0x2
HTC:          0x4
HTT:          0x8
MAC:          0x10
BOOT:         0x20
PCI-DUMP:     0x40
HTT-DUMP:     0x80
MGMT:         0x100
DATA:         0x200
BMI:          0x400
REGULATORY:   0x800
TESTMODE:     0x1000
WMI-PRINT:    0x2000
PCI-PS:       0x4000
AHB:          0x8000
NO-FW-DBGLOG:0x10000000
MAC2:         0x20000000
INFO-AS-DBG:  0x40000000
FW:           0x80000000
ALL:          0xFFFFFFFF
```

To change the logging level at run-time:

```
echo 0xc0000032 > /sys/kernel/debug/ieee80211/phy0/ath10k/debug_level
```

You can change this at module load time by setting the appropriate value in modprobe.conf files and/or using options to 'insmod'. Your particular OS distribution may use different means, but normally something like this works:

```
$ cat /etc/modprobe.d/ath10k.conf
# Boot & MAC
#options ath10k_core debug_mask=0xc0000030

# Boot, MAC, and WMI
options ath10k_core debug_mask=0xc0000032
```

If your performance goes down a lot when you enable debugging, it may be because all of the logs are spammed to the serial console instead of just showing up in dmesg. You can try this to fix that problem:

```
sysctl "kernel.printk=0 4 1 7"
```

- **Provide details about your network configuration.** For instance:

"I have two archer c7 systems running LEDE with the QCA9880 chipset. They are using ath10k-ct firmware version 10.1-ct-8x-\_\_xTH-019-7f130e1. I set up an encrypted adhoc network between them and attempt to ping. Ping shows total path loss. This same setup works fine if I disable encryption or if I use AP/STA instead of IBSS. My LEDE config files are as follows..."

You can get the firmware version in a number of ways. It should be printed in 'dmesg' right after the system boots. You can also get it from ethtool if that is on your platform:

```
[root@ben-ota-2 lanforge]# ethtool -i wlan0
driver: ath10k_pci
version: 4.9.29+
firmware-version: 10.1-ct-8x-__xTH-019-7f130e1
expansion-rom-version:
bus-info: 0000:05:00.0
```

Please include details on your system configuration, including how you set up your adhoc network, for instance. If it works in one config but not in others, that is good info. If only some of the systems fail, that is more good info. If other NICs and/or hardware platforms work, that is also good to know.

- **Try some different firmware.** Ath10k-CT firmware is compiled in various different configurations. You could first try the latest beta builds, and you could try the htt-mgt firmware if you are using the default 'wmi-mgt' firmware. You could also try older releases. If you can determine that old firmware worked and new does not, then it probably means a regression was added, and we can work together to bisect the problem (I have scripts to build a binary for every commit, for instance).

In general, for the ath10k 10.1 (wave-1) firmware, you will download the firmware binary to your platform, and then copy over the existing firmware-2.bin file. There are several different builds of the firmware with different compile options. If unsure, the 'full community' version is a good starting point. It may be worth trying others (though the 'nrcc' variants typically will not load without a customized fwcfg file). For instance:

```
cp /tmp/firmware-2-htt-mgt.bin /usr/lib/firmware/ath10k/QCA988X/hw2.0/firmware-2.bin
```

To reload the firmware, reload the driver (or just reboot):

```
rmmod ath10k_pci ath10k_core
modprobe ath10k_pci
```

Use 'dmesg' to verify the new firmware version was properly loaded.

```
dmesg | grep ath10k
973 [516861.711132] ath10k driver, optimized for CT firmware, probing pci device: 0x3c.
974 [516861.711725] ath10k_pci 0000:07:00:0: pci irq msi oper_irq_mode 2 irq_mode 0 reset_mode 0
975 [516861.779562] ath10k_pci 0000:05:00:0: Direct firmware load for ath10k/pre-cal-pci-0000:05:00:0.bin failed with error -2
976 [516861.779570] ath10k_pci 0000:05:00:0: Direct firmware load for ath10k/cal-pci-0000:05:00:0.bin failed with error -2
977 [516861.779600] ath10k_pci 0000:05:00:0: fwcfg key: vdevs val: 64
978 [516861.779601] ath10k_pci 0000:05:00:0: fwcfg key: peers val: 128
979 [516861.779603] ath10k_pci 0000:05:00:0: fwcfg key: active_peers val: 128
980 [516861.779604] ath10k_pci 0000:05:00:0: fwcfg key: stations val: 127
981 [516861.779605] ath10k_pci 0000:05:00:0: fwcfg key: rate_ctrl_objs val: 16
982 [516861.779607] ath10k_pci 0000:05:00:0: fwcfg key: regdom val: 840
983 [516861.779608] ath10k_pci 0000:05:00:0: fwcfg key: fwname val: firmware-2.bin
984 [516861.779609] ath10k_pci 0000:05:00:0: fwcfg key: fwver val: 2
985 [516861.779610] ath10k_pci 0000:05:00:0: fwcfg key: nohwcrypt val: 1
986 [516861.779611] ath10k_pci 0000:05:00:0: fwcfg key: tx_desc val: 680
987 [516861.779613] ath10k_pci 0000:05:00:0: fwcfg key: tids val: 256
988 [516861.779614] ath10k_pci 0000:05:00:0: fwcfg key: skid_limit val: 360
989 [516861.779615] ath10k_pci 0000:05:00:0: fwcfg key: max_amsdus val: 3
990 [516861.779697] ath10k_pci 0000:05:00:0: qca988x hw2.0 target 0x4100016c chip_id 0x043202ff sub 0000:0000
991 [516861.779698] ath10k_pci 0000:05:00:0: kconfig debug 1 debugfs 1 tracing 1 dfs 1 testmode 1
992 [516861.779850] ath10k_pci 0000:05:00:0: firmware ver 10.1-ct-8x-__xtH-019-7f130e1 api 2 features wmi-10.x,txstatus-noack,wmi-10.
993 [516861.808040] ath10k_pci 0000:05:00:0: Direct firmware load for ath10k/QCA988X/hw2.0/board-2.bin failed with error -2
994 [516861.808059] ath10k_pci 0000:05:00:0: board_file api 1 bmi_id N/A crc32 bebc7c08
...
```

Wave-2 (9888, 9980, 9984, etc) are similar, but will be called 'firmware-5.bin' instead, and each chipset will have its own firmware directory in `/lib/firmware/ath10k/`, so please pay attention to that.

- **How to Bisect.** Sometimes a series of firmware images can be tested to find where the regression happened. You can search this with a binary bisect. The method is to first try the very first firmware. If it passes, then try the very last firmware. If it fails, then pick a firmware commit half way between the good and bad and test it. Repeat until you find the exact commit that introduces the failure. Please double-check at that point to make sure that you have found the correct commit.

Bisect images are named with a numeric commit-id followed by a git hash. For instance:

```
firmware-5-full-htt-mgt-community-commit-970-b2b0340a6.bin
```

This firmware is commit number 970, with git hash b2b0340a6.

Please double-check the firmware version your system is using by looking at the firmware version info in `dmesg` after loading the new firmware. The git-hash should be in the version string.

- **For firmware crashes, grab a binary dump if you can.** If you are experiencing a firmware crash, then some crash info will usually be in the `dmesg` output, but a more complete crash dump can be found in the `debugfs` files. In the case below, there is no crash, but if you had a crash, then you could save it like this. Each time you read the file, the crash will be erased from the driver memory, so save it properly the first time! Attach the `/tmp/crash-ath10k-ct.bin` to bug reports if you can:

```
[root@ben-ota-2 lanforge]# cat /sys/kernel/debug/ieee80211/wiphy0/ath10k/fw_crash_dump > /tmp/crash-ath10k-ct.bin
cat: /sys/kernel/debug/ieee80211/wiphy0/ath10k/fw_crash_dump: No data available
```