**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 as 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:

- **Dump 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 dump logs, you need to change the ath10k debug level. You can adjust this at runtime through `debugfs`, and `0xc0000032` is a good value to start with:

```
cat /sys/kernel/debug/ieee88211/wiphy1/ath10k/debug_level
```

Current debug level: `0xc0000032`

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

<table>
<thead>
<tr>
<th>PCI</th>
<th>WMI</th>
<th>HTC</th>
<th>HTT</th>
<th>MAC</th>
<th>BOOT</th>
<th>PCI-DUMP</th>
<th>HTT-DUMP</th>
<th>MDM</th>
<th>DATA</th>
<th>WMI</th>
<th>REGULATORY</th>
<th>TESTMODE</th>
<th>WMI-PRINT</th>
<th>PCI-PS</th>
<th>AHB</th>
<th>NO-FW-DBGLOG</th>
<th>MAC2</th>
<th>INFO-AS-DBG</th>
<th>FW</th>
<th>ALL</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x1</td>
<td>0x2</td>
<td>0x4</td>
<td>0x8</td>
<td>0x10</td>
<td>0x20</td>
<td>0x40</td>
<td>0x80</td>
<td>0x100</td>
<td>0x200</td>
<td>0x400</td>
<td>0x800</td>
<td>0x1000</td>
<td>0x2000</td>
<td>0x4000</td>
<td>0x8000</td>
<td>0x28800000</td>
<td>0x48800000</td>
<td>0x88800000</td>
<td>0x88800000</td>
<td>0xF0000000</td>
</tr>
</tbody>
</table>

To change the logging level at run-time:

```
echo `0xc0000032` > /sys/kernel/debug/ieee88211/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 `modprobe`. 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
```

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

  ```
  [root@ben-ota-2 lanforge]# ethtool -i wlan0
driver: ath10k_pci
version: 4.9.29+
firmware-version: 10.1.x-Bx__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. It works in one config but not in others, that is good info. If only one of the systems fails, that is more good info. If it 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 `ht-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 isolate 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 "rcx" variants typically will not load without a customized `twcg` file). For instance:

```
cp /tmp/firmware-2-ht-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
modprobe ath10k_pci
```
Use 'dmesg' to verify the new firmware version was properly loaded.

dmesg | grep ath10k
973 [158681.71132] ath10k driver, optimized for CT firmware, probing pci device: 0x3c.
974 [158681.711725] ath10k_pci 0000:07:00.0: pci irq msi oper irq node 2 irq node 0 reset node 0
975 [158681.779561] ath10k_pci 0000:05:00.0: Direct firmware load for ath10k/pre-cal-pci-0000:05:00.0.bin failed with error -2
976 [158681.779579] ath10k_pci 0000:05:00.0: Direct firmware load for ath10k/cal-pci-0000:05:00.0.bin failed with error -2
977 [158681.779606] ath10k_pci 0000:05:00.0: funcfg key: vdevs val: 64
978 [158681.779611] ath10k_pci 0000:05:00.0: funcfg key: peers val: 128
979 [158681.779633] ath10k_pci 0000:05:00.0: funcfg key: active_peers val: 128
980 [158681.779641] ath10k_pci 0000:05:00.0: funcfg key: stations val: 127
981 [158681.779665] ath10k_pci 0000:05:00.0: funcfg key: rate_ctrl_objs val: 16
982 [158681.779667] ath10k_pci 0000:05:00.0: funcfg key: regdom val: 840
983 [158681.779668] ath10k_pci 0000:05:00.0: funcfg key: faname val: firmware-2.bin
984 [158681.779689] ath10k_pci 0000:05:00.0: funcfg key: fwver val: 2
985 [158681.779690] ath10k_pci 0000:05:00.0: funcfg key: nohwcrypt val: 1
986 [158681.779691] ath10k_pci 0000:05:00.0: funcfg key: tx_desc val: 680
987 [158681.779692] ath10k_pci 0000:05:00.0: funcfg key: tids val: 256
988 [158681.779694] ath10k_pci 0000:05:00.0: funcfg key: skid_limit val: 368
989 [158681.779695] ath10k_pci 0000:05:00.0: funcfg key: max_amsdu val: 3
990 [158681.779697] ath10k_pci 0000:05:00.0: qca988x hw2.0 target 0x4100016c chip_id 0x043202ff sub 8000:0000
991 [158681.779698] ath10k_pci 0000:05:00.0: kconfig debug 1 debugfs 1 tracing 1 dfs 1 testmode
992 [158681.779856] 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.x-ct,
993 [158681.808058] ath10k_pci 0000:05:00.0: Direct firmware load for ath10k/QCA988X/hw2.0/board-2.bin failed with error -2
994 [158681.808059] ath10k_pci 0000:05:00.0: board_file api 1 be1_id N/A crc32 7dec7b0

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 bisection. The method is to first try the very first firmware. If it fails, then try the very last firmware. If it fails, then pick a firmware committ that is halfway between the good and bad, and test it. Repeat until you find the exact commit that introduced the failure. Please double-check at that point to make sure that you have found the correct commit.

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

firmware-5-full-http-ng-t-commit-970-b2b0348a8.bin

This firmware is commit number 970, with git hash b2b0348a8.

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 debug 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 drive memory, so save it properly the first time! Attach the /tmp/crash-ath10k-dump bin to bug reports if you can:

```
[root@ben-ota-2 lanforge] cat /sys/kernel/debug/ieee80211/ath10k/fw_crash_dump > /tmp/crash-ath10k-ct.bin
```

Copyright Technologies, Inc., 247 East Street, Suite 301, Dedham, MA 02026, USA
www.candestech.com | sales@candestech.com | +1.860.281.4138