# Linux 802.11n CSI Tool

GitHub | Main Page | FAQ | Get Help

## Installation Instructions

These instructions are currently expected to work on Linux operating systems that are based on an upstream Linux kernel version between 3.2 (e.g. Ubuntu 12.04) and 4.2 (e.g. Ubuntu 14.04.4).

An older version of this page, used with the original testbed at the University of Washington, is archived here.

### 1. Prerequisites

Install the build tools, the Linux development headers, and the Git client. On Ubuntu, run:

```
sudo apt-get install gcc make linux-headers-$(uname -r) git-core
```

> Tip: To maintain greater control over the wireless interface during your experiments, you will likely want to configure it using command-line utilities such as iw and iproute2. If so, you first need to disable NetworkManager from controlling the wireless interface (if it is installed). On Ubuntu, run:
>
> ```
> sudo apt-get install iw
> echo iface wlan0 inet manual | sudo tee -a /etc/network/interfaces
> sudo restart network-manager
> ```
>
> Tip: If you want to prevent the wireless driver from automatically loading during boot (particularly if you are developing your own modifications to the driver), run:
>
> ```
> echo blacklist iwldvm | sudo tee -a /etc/modprobe.d/csitool.conf
> echo blacklist iwlwifi | sudo tee -a /etc/modprobe.d/csitool.conf
> ```

### 2. Build and Install the Modified Wireless Driver

Obtain the CSI Tool Linux source tree, which includes modifications to the wireless driver, and check out the appropriate tag in the repository for your upstream kernel version:

```
CSITOOL_KERNEL_TAG=csitool-$(uname -r | cut -d . -f 1-2)
git clone https://github.com/dhalperi/linux-80211n-csitool.git
cd linux-80211n-csitool
git checkout ${CSITOOL_KERNEL_TAG}
```

> Tip: You can then merge these driver modifications into the Linux source tree for your distribution-provided kernel version, in order to improve compatibility. On Ubuntu, run:
>
> ```
> UBUNTU_KERNEL_TAG=Ubuntu-3.13.0-32.57
> # Modify the line above with your Ubuntu kernel tag. First, determine your full kernel
> # version by reading /proc/version_signature; then, look up the Ubuntu kernel tag at:
> # http://people.canonical.com/~kernel/info/kernel-version-map.html
>
> . /etc/lsb-release
> git remote add ubuntu git://kernel.ubuntu.com/ubuntu/ubuntu-${DISTRIB_CODENAME}.git
> git pull --no-edit ubuntu ${UBUNTU_KERNEL_TAG}
> ```

Build the modified driver for the existing kernel:

```
make -C /lib/modules/$(uname -r)/build M=$(pwd)/drivers/net/wireless/iwlwifi modules
```

Install the modified driver into the module updates directory:

```
sudo make -C /lib/modules/$(uname -r)/build M=$(pwd)/drivers/net/wireless/iwlwifi INSTALL_MOD_DIR=updates \
    modules_install
```

```
    sudo depmod
    cd ..
```

> Tip: If the message "Can't read private key" appears, then the module could not be signed. This will not cause a problem unless the kernel enforces module signature verification.

## 3. Install the Modified Firmware

Obtain the CSI Tool supplementary material:

```
    git clone https://github.com/dhalperi/linux-80211n-csitool-supplementary.git
```

Relocate any existing firmware for Intel Wi-Fi Link 5000 Series adapters:

```
    for file in /lib/firmware/iwlwifi-5000-*.ucode; do sudo mv $file $file.orig; done
```

Install the modified firmware:

```
    sudo cp linux-80211n-csitool-supplementary/firmware/iwlwifi-5000-2.ucode.sigcomm2010 /lib/firmware/
    sudo ln -s iwlwifi-5000-2.ucode.sigcomm2010 /lib/firmware/iwlwifi-5000-2.ucode
```

## 4. Build the Userspace Logging Tool

Build log_to_file, a command line tool that writes CSI obtained via the driver to a file:

```
    make -C linux-80211n-csitool-supplementary/netlink
```

## 5. Enable Logging and Test

Unload the driver:

```
    sudo modprobe -r iwlwifi mac80211
```

> Tip: The message "FATAL: Module iwlwifi is in use." may appear if the iwldvm module needs to be explicitly unloaded first (on distributions other than Ubuntu). If so, use this instead:
>
> ```
>     sudo modprobe -r iwldvm iwlwifi mac80211
> ```

Once the driver is unloaded, reload it with CSI logging enabled:

```
    sudo modprobe iwlwifi connector_log=0x1
```

Connect to an 802.11n access point, by using (for example) the iw and iproute2 utilities (or if NetworkManager is enabled for the wireless interface, using nm-cli or the graphical applet). Alternatively, you can let your system function as an 802.11n access point by installing and configuring hostapd (see the supplementary material for configuration examples); in this case, start hostapd on this system and then connect another 802.11n station to it.

Either way, once you are connected, begin logging CSI to a file:

```
    sudo linux-80211n-csitool-supplementary/netlink/log_to_file csi.dat
```

At a different terminal, run the ping command with the IP address of the connected access point (or connected station). If replies are sent back at High Throughput (802.11n) bitrates, log_to_file will write the CSI for each received packet to the file, and also print a message to the terminal. Note that depending on the rate selection algorithm, the first few packets are typically sent at Legacy (802.11a/b/g) bitrates instead of High Throughput bitrates; these will not provide CSI.

Make sure to see the FAQ for more details about collecting CSI.

Back to the main page