Cannot get Intel 7260 to work properly on Ubuntu 20.04. Disconnects intermittently; need to reboot to restart it

Asked 2 years ago Modified 1 month ago Viewed 3k times

7

3

I am running Ubuntu 20.04 with the low latency kernel for Ubuntu Studio. (To be really specific I actually installed Kubuntu 20.04, then install Ubuntu Studio, and got the low latency kernel that way; I doubt any of that matters, just trying to be thorough.)

I have the Intel 7260.HMW Dual Band Wireless-AC 7260 Network Adapter PCI Express Half Mini Card 802.11 b/a/g/n/ac - Here's a link to the specific card I bought on Amazon: https://www.amazon.com/gp/product/B00MV3N7UO/ref=ppx_yo_dt_b_asin_title_o04_s00?ie=UTF8&psc=1

The problem I am having is that every few hours my wifi would get disconnected/deactivated and I would have no way to restart it except to restart my system. Just loggin in/out would not work. I tried all sort of terminal command to restart NetworkManager, reload kernel moduls, etc. but nothing worked.

A slight positive, whenever I get disconnected I am able to plug in a usb wifi adaptor and almost instantly am once again connected to the internet through that, but I still cannot use my internal wifi card until I restart my system.

Another thing I noticed, after much fiddling with NetworkManager, (and adding the Gnome NetworkManager, which seemed somewhat better than the default KDE/Plasma version) was that before wifi became completely unresponsive other smaller issues would show up in NetworkManager. The first thing was that all of the networks, except the one I was connected to, would disappear from the list. After that, certain options would be grayed out, like "Connect to hidden wifi network," "Enable networking," "Enable wifi;" not sure about other options. Soon after, wifi would just become completely unresponsive. The system would be completely unable to even detect that I had a wifi device installed.

I also noticed that if I ran `nmcli device wifi list` when the non-connected networks would disappear from the list in the NetworkManager gui, they would often re-appear, and this seemed to stave off the problem of wifi turning off -- but it would still turn off eventually, so this was not a viable solution. (I even made a cron job to run `nmcli device wifi list` every 20 seconds, so I wouldn't have to keep checking and running it manually. This did not solve the problem either.)

In "/etc/NetworkManager/conf.d" I had one file, "default-wifi-powersave-on.conf" with the following contents:

```
[connection]
wifi.powersave = 3
```

I deleted that file, and replaced it with one called "wifi-powersave.conf" with the follwing (eventually adding the two extra settings, one for iwlwifi and one for wlp3s0, just because I was trying to be thorough, and try everything I could think of):

```
[connection]
# Values are 0 (use default), 1 (ignore/don't touch), 2 (disable) or 3 (enable).
wifi.powersave = 2                  [This did not seem to have an effect.]
iwlwifi.powersave = 2               [This may not have an effect.]
wlp3s0.powersave = 2                [This might do it!]
```

My original kernel was 5.4, but got upgraded during a normal suggested upgrade to 5.42. After reading a few things that seemed to work for others, I switched kernels, first to 5.7.10 then 5.6.7., all of them low latency. That didn't help

Then tried adding the following to etc/modprobe.d/blacklist.conf: (Because of a post about someone else with a 7260, on a different distro, who added that line, plus a line "blacklist acer_wmi" and it worked. I don't have and Acer, but a ThinkPad T430, so I just added the one line.)

```
blacklist btusb
```

Didn't fix the problem.

Then I removed the blacklist btusb, and instead tried modifying the iwlwifi.conf, in the same folder, adding the following options.

```
options iwlwifi bt_coex_active=0
options iwlwifi power_save=0
options iwlmvm power_scheme=1
options iwlwifi d0i3_disable=1
options iwlwifi uapsd_disable=1
```

Didn't fix the problem.

And here's the result of `sudo lshw -C network` done at various points of my wifi functionality:

When fully functional

```
*-network
       description: Wireless interface
       product: Wireless 7260
       vendor: Intel Corporation
       physical id: 0
       bus info: pci@0000:03:00.0
```

```
        logical name: wlp3s0
        version: bb
        serial: 7c:5c:f8:dc:f4:f1
        width: 64 bits
        clock: 33MHz
        capabilities: pm msi pciexpress bus_master cap_list ethernet physical wireless
        configuration: broadcast=yes driver=iwlwifi driverversion=5.4.0-40-lowlatency
firmware=17.3216344376.0 ip=172.20.20.20 latency=0 link=yes multicast=yes
wireless=IEEE 802.11
        resources: irq:34 memory:f1c00000-f1c01fff
```

When starting to fail - Notice "bus_master" is missing from capabilities

```
*-network
        description: Wireless interface
        product: Wireless 7260
        vendor: Intel Corporation
        physical id: 0
        bus info: pci@0000:03:00.0
        logical name: wlp3s0
        version: bb
        serial: 7c:5c:f8:dc:f4:f1
        width: 64 bits
        clock: 33MHz
        capabilities: pm msi pciexpress cap_list ethernet physical wireless
        configuration: broadcast=yes driver=iwlwifi driverversion=5.4.0-40-lowlatency
firmware=17.3216344376.0 latency=0 link=no multicast=yes wireless=IEEE 802.11
        resources: irq:33 memory:f1c00000-f1c01fff
```

When not working - Now even more missing from capabilities; also description is now "Network controller" not "Wireless interface"

```
*-network UNCLAIMED
        description: Network controller
        product: Wireless 7260
        vendor: Intel Corporation
        physical id: 0
        bus info: pci@0000:03:00.0
        version: bb
        width: 64 bits
        clock: 33MHz
        capabilities: pm msi pciexpress cap_list
        configuration: latency=0
        resources: memory:f1c00000-f1c01fff
```

That's about it. Of course I tried all the simple stuff like trying to restart NetworkManger, in both of the following ways:

```
sudo systemctl restart NetworkManager

sudo service network-manager restart
```

I also tried removing and reloading the wifi drivers:

```
sudo modprobe -r iwlwifi && sudo modprobe iwlwifi
```

So, that is everything that I've tried. I also found this tidbit here: https://wireless.wiki.kernel.org/en/users/drivers/iwlwifi#d_3165_and_3168_support

On this page it states:

> **7260, 3160, 7265, 7265D, 3165 and 3168 support**
>
> Those devices will not be supported by the newest firmware versions: the last firmware that was released for 3160, 7260 and 7265 is -17.ucode. Bug fixes will be ported to -17.ucode.

Now, does this mean that these devices will no longer be usable in linux?

By the way, I am able to easily change wifi cards in my laptop. If there is a card out there, half pci size, that would be excellent for use in my laptop, and will actually work, I'd like to know about that! I do plan on using this computer with Kali as well, so I'd like the card to be really fast and good for normal stuff, but also have all the features needed by Kali. I thought this one would fit the bill, if I could just get it to work. (Would ndiswrapper work, perhaps?)

I've noticed a LOT of other similar posts, both with the identical Intel card and with other Intels, so if we can find a fix, this would probably also help many others as well.

EDIT: Since posting the above I have tried more things, and will share them here now.

While I was switched to 5.6.7-lowlatency I tried to install backport-iwlwifi-dkms but it did not install properly (I am sorry I forget the error message). So, I then reverted back to 5.4.0-42-lowlatency, deleted the other higher kernels I had tried, and then installed backport-iwlwifi-dkms successfully without error. This however did nothing to solve my wifi problem, so I later uninstalled it.

At this point I had reverted all of my changes back to what the system was originally, except for the changes I made to iwlwifi.com, which didn't seem to help or hurt, or really affect anything in any way.

I was now at a loss for what to do to proceed, as I had tried just about everything. I was thinking of trying ndiswrapper, but some info I had found seemed to indicate that my wifi chipset would not work with that, so I am putting that idea on the shelf for a while.

I then tried running 'tail -f /var/log/syslog' just so I could keep tabs of what was happening when wifi failed. Here is some of the output:

```
Jul 30 11:03:34 ENCOM-T430 systemd[1]: Starting Network Manager Script Dispatcher
Service...
Jul 30 11:03:34 ENCOM-T430 dbus-daemon[582]: [system] Successfully activated service
'org.freedesktop.nm_dispatcher'
Jul 30 11:03:34 ENCOM-T430 systemd[1]: Started Network Manager Script Dispatcher
Service.
Jul 30 11:03:43 ENCOM-T430 systemd[1]: NetworkManager-dispatcher.service: Succeeded.
Jul 30 11:05:14 ENCOM-T430 kernel: [ 2511.456439] iwlwifi 0000:03:00.0: Failed to
wake NIC for hcmd
Jul 30 11:05:14 ENCOM-T430 kernel: [ 2511.457393] iwlwifi 0000:03:00.0: Error sending
STATISTICS_CMD: enqueue_hcmd failed: -5
Jul 30 11:05:15 ENCOM-T430 kernel: [ 2512.960634] iwlwifi 0000:03:00.0: Failed to
wake NIC for hcmd
Jul 30 11:05:15 ENCOM-T430 kernel: [ 2512.961409] iwlwifi 0000:03:00.0: Error sending
STATISTICS_CMD: enqueue_hcmd failed: -5
Jul 30 11:05:16 ENCOM-T430 kernel: [ 2512.981002] iwlwifi 0000:03:00.0: Failed to
wake NIC for hcmd
Jul 30 11:05:16 ENCOM-T430 kernel: [ 2512.981039] iwlwifi 0000:03:00.0: Error sending
STATISTICS_CMD: enqueue_hcmd failed: -5
Jul 30 11:05:19 ENCOM-T430 kernel: [ 2516.454834] iwlwifi 0000:03:00.0: Failed to
wake NIC for hcmd
Jul 30 11:05:19 ENCOM-T430 kernel: [ 2516.454900] iwlwifi 0000:03:00.0: Error sending
STATISTICS_CMD: enqueue_hcmd failed: -5
Jul 30 11:05:21 ENCOM-T430 kernel: [ 2518.956560] iwlwifi 0000:03:00.0: Failed to
wake NIC for hcmd
Jul 30 11:05:22 ENCOM-T430 kernel: [ 2518.956655] iwlwifi 0000:03:00.0: Error sending
STATISTICS_CMD: enqueue_hcmd failed: -5
Jul 30 11:05:22 ENCOM-T430 kernel: [ 2518.975830] iwlwifi 0000:03:00.0: Failed to
wake NIC for hcmd
Jul 30 11:05:22 ENCOM-T430 kernel: [ 2518.975945] iwlwifi 0000:03:00.0: Error sending
STATISTICS_CMD: enqueue_hcmd failed: -5
Jul 30 11:05:22 ENCOM-T430 kernel: [ 2519.549253] ------------[ cut here ]----------
-
Jul 30 11:05:22 ENCOM-T430 kernel: [ 2519.549258] Timeout waiting for hardware access
(CSR_GP_CNTRL 0xffffffff)
Jul 30 11:05:22 ENCOM-T430 kernel: [ 2519.549317] WARNING: CPU: 2 PID: 24 at
drivers/net/wireless/intel/iwlwifi/pcie/trans.c:2066
iwl_trans_pcie_grab_nic_access+0x1f9/0x230 [iwlwifi]
```

Now it's back to searching for something based upon this new information. If anyone out there has any thoughts, ideas, or words of kind encouragement at this point, I would really appreciate it. Also, if anyone can recommend a really good half-pci wifi card I can replace this one with, with all the features necessary for Kali as well, I would welcome that too!

20.04 network-manager iwlwifi

Share Improve this question Follow

edited Jul 30, 2020 at 16:28

asked Jul 28, 2020 at 1:50

Giovanni_Visione

121 1 8

After having experience the issue several more times, without seeing the "Starting Rotate log file..." line before the problem, it seem that this really had nothing to do with my actual issue. Just a clarification.

– Giovanni_Visione Jul 29, 2020 at 23:33

2 Answers

Sorted by:

Highest score (default)

4

Intel-7260-Wifi-Fix

Note: Both from clues found on the Internet, as well as personal experience, it seems that there are certain Intel 7260 WIFI PCI cards which actually work fine, and others which have the problems addressed herein. A *much* better fix than the method described below is to simply purchase the right card, because even with the fix below your wifi connection will still occassionally be going on and off, which is certainly not ideal, even though the fix below does make it automatically reconnect.

I originally bought this wifi card: https://www.amazon.com/gp/product/B00MV3N7UO/ref=ppx_yo_dt_b_asin_title_o08_s00?ie=UTF8&psc=1 If you look at the picture of the card, you can see that the Model is 7260HMW BN. Once I got the card it worked great *when* it worked, and for the times it stopped working I devised the fix described below.

After a few weeks I then purchased this card: https://www.amazon.com/gp/product/B01E85QIFI/ref=ppx_yo_dt_b_asin_title_o01_s00?ie=UTF8&psc=1 If you look at the picture of that card, you can see that the Model is 7260HMN. Once I got this card I removed the fix from my laptop, and just let it run to see what would happen. It worked PERFECTLY!

My advice is that if you want an Intel 7260 WIFI PCI card in your machine, that you are careful to purchase the Model 7260HMW - not the 7260HMW BN, and probably not the 7260HMW NB or the 7260HMW AC. There is a comparison of these various cards, and the 3160HMW here: https://www.legitreviews.com/intel-7260hmwg-802-11ac-versus-intel-7260hmw-bn-802-11n_135541 As you can see, the 7260HMW has the best and most complete features, and it also happens to be the one that actually works perfectly on Linux!

If anyone comes across this post, please comment to share your experience with others, being very careful to note which card you have. If you can physically look at your card (which would require opening your machine) please report the Model printed on the card itself. Also, the output of `sudo lshw -C network` (the wifi part) could also be of use, in particular the "version."

Here is my output for the first card, the one with the problems:

```
*-network
    description: Wireless interface
    product: Wireless 7260
    vendor: Intel Corporation
    physical id: 0
    bus info: pci@0000:03:00.0
    logical name: wlp3s0
    version: bb
    serial: 7c:5c:f8:dc:f4:f1
```

```
    width: 64 bits
    clock: 33MHz
    capabilities: pm msi pciexpress bus_master cap_list ethernet physical wireless
    configuration: broadcast=yes driver=iwlwifi driverversion=5.4.0-40-lowlatency
firmware=17.3216344376.0 ip=172.20.20.20 latency=0 link=yes multicast=yes
wireless=IEEE 802.11
    resources: irq:34 memory:f1c00000-f1c01fff
```

Here is my output for the second card, the one that worked perfectly:

```
*-network
     description: Wireless interface
     product: Wireless 7260
     vendor: Intel Corporation
     physical id: 0
     bus info: pci@0000:03:00.0
     logical name: wlp3s0
     version: 73
     serial: a0:a8:cd:2c:f3:da
     width: 64 bits
     clock: 33MHz
     capabilities: pm msi pciexpress bus_master cap_list ethernet physical wireless
     configuration: broadcast=yes driver=iwlwifi driverversion=5.4.0-42-lowlatency
firmware=17.3216344376.0 ip=172.20.20.20 latency=0 link=yes multicast=yes
wireless=IEEE 802.11
     resources: irq:33 memory:f1c00000-f1c01fff
```

The only differences are the *version* and the *serial,* and I think it is actually the *version* which is pertinent here.

I have done a lot of the troubleshooting already. It would be nice to get some feedback so this problem can be definitively resolved for the community.

And...if you are stuck with a misbehaving Intel 7260 for now...here's the fix I came up with for that:

A fix for Intel 7260 WIFI PCI cards, which intermittently and unpredictably stop working on Linux.

(With a little bit of know how, this might easily be adapted to support other chipsets.)

The Intel 7260 WIFI PCI cards have fantasitic wifi capabilities, but on Linux are notorious for intermittently and unpredictably shutting down and becoming completely non-responsive, with no way of restarting the card except for rebooting the system.

After a LOT of searching I found a couple of scripts which could be run that would restart the card. While that was nice, the card would definitely still go down from time to time, and then require the user to manually run the script. This was an improvement, but not very convenient,

and I wanted a way to automate the process so I could simply forget about it and have it just work.

I took the script and modified it just a bit, and also added a few checks at the beginning of the script, which would check in various ways whether the the wifi card was working or not. (At first, the only checks I had were based upon `nmcli` and `ifconfig` but it seemed that there were failures which these would not catch. I then added another check based on the output of `lshw`, because while debugging and suffering with this problem I had noticed different outputs of `lshw` depending upon whether the card was working or not; specifically, when the card was working I would see that "bus_master" was listed under the capabilities for the device, but that this would go missing when it had failed, or even was just starting to fail.) Anyway, once these checks were in place, once the script was run, the following would happen:

(1) If the wifi was found to be WORKING, the the script would simply exit.

(2) If the wifi was found to be NOT WORKING, then the script would continue and perform the wifi reset.

I then set up some cron jobs which would run my modified script every 20 seconds. Once I had this all set up, my wifi problems were over!

How to set this up

The setup does take a few minutes and some preparation, but it it WELL worth it, and I will take you through step-by-step!

First, you need to have `ifconfig` installed on your system. I think it would be relatively easy to modify the script to use `ip` instead, or even to detect which of these were available on your system, but I have not implemented that yet. Anyway, as it is now, you want to make sure you have `ifconfig` installed so first just run: `sudo apt install net-tools` Now that you hve `ifconfig` installed, you can now proceed to download these two files into your home directory:

```
https://raw.githubusercontent.com/John-Vision/Intel-7260-Wifi-Fix/master/fixwifi
```

```
https://raw.githubusercontent.com/John-Vision/Intel-7260-Wifi-Fix/master/fixwifi-force
```

To download them from within the terminal copy/paste/run the following lines

```
cd ~
```

```
wget https://raw.githubusercontent.com/John-Vision/Intel-7260-Wifi-Fix/master/fixwifi
```

```
wget https://raw.githubusercontent.com/John-Vision/Intel-7260-Wifi-Fix/master/fixwifi-force
```

Now that you have these two file in your home directory, you need to make them executable.

```
chmod +x ~/fixwifi

chmod +x ~/fixwifi-force
```

These two files are essentially the same, but with one difference: `fixwifi` first checks to see if you wifi is up and running; if it is then it just exits, but if not then it goes ahead and reset your wifi. 'fixwifi-force', on the other hand, does not bother to perform any check, and will reset your wifi whether it's already running or not.

Both of these files have some settings which you can change manually. Assuming that you have the Intel 7260 (which is what this is all about!) you shouldn't have to change anything, except *possibly* the line in each file (about line 19 in each) which says `interface="wlp3s0"` . Your interface may be different: typical values are things like, wlan0, wlp2s0, and the like. You can check your interface by executing `sudo lshw -C network | grep "logical name: w"` , as long as you run this while your wifi is working. So, if needed, just change the interface setting to whatever is appropriate for you, in each of these two files.

Once you have all this done, try `~/fixwifi-force` . If everything worked you should see your wifi get disconnected (if it was already connected) and then come back online. If this did not happen, then you need to check the output, and see if there are any errors. The most common (and easy to fix) error would be having the interface set wrong. (See paragraph above.) Another possibility is that you don't have an Intel 7260, in which case you would also have to change the part between the quotes in the setting for "wirelessPCI," and probably also the setting for "voodoo". (I have no idea of how to help you with the voodoo setting. This part is pretty much a mystery to me.)

Once you have `~/fixwifi-force` up and running, you are really in luck! Just make sure you have the same settings in `fixwifi` that worked for you in `fixwifi-force` . Now all you need to do is set up some cron jobs to run `fixwifi` periodically in the background, so you never have to think about it again!

If this is the first time you are using cron, the following makes sure it can run in the background:

```
sudo systemctl enable cron
```

Now it's time to go ahead and set up a crontab as root:

```
sudo crontab -e
```

It will ask what editor you want to use. Pick the one you want. (As the prompt will tell you, nano is the simplest.)

Now you need to add the following three lines, replacing the path with the actual path to your .fixwifi file. (Don't enter the path as a shortcut like "~/fixwifi" but actually go ahead and type out the full path.)

```
* * * * * /path/to/.fixwifi

* * * * * sleep 20; /path/to/.fixwifi

* * * * * sleep 40; /path/to/.fixwifi
```

When you have added these three lines, modified to reflect the actual path, save the file and you're done! (If you chose nano, press Ctrl-X to finish editing and then press "y" in response to "Save modified buffer?" and then just press "Enter" to accept the name of the file you want to send it to.)

That's it! Enjoy your new, stress free Intel 7260 Wifi!

Share Improve this answer Follow edited Aug 14, 2020 at 20:00 answered Jul 31, 2020 at 1:29

Giovanni_Visione
121 1 8

Great job in tracking the problem and providing a fix. Just want to mention - for those who would not look at the script source - that the bug is discussed here: bugzilla.kernel.org/show_bug.cgi?id=191601 – Pietro Battiston Feb 1, 2021 at 9:09

That's a nice workaround for the cron job minimum interval of 1 minute compared to the 20 seconds you needed. – progonkpa Nov 3, 2021 at 4:37

I just implemented your fix, let's see how it goes. Thanks for your efforts. This wifi bug is insufferable. It also always happens on the most inconvenient of times :) – progonkpa Nov 3, 2021 at 4:45

The BEST fix of all is to get a wifi card that actually works right, all the time. If, like me, you really want to use an Intel 7260, then the key is to find the RIGHT Intel 7260, as described in my answer. (This RIGHT one is the Model 7260HMW - not the 7260HMW BN, and probably not the 7260HMW NB or the 7260HMW AC.) If you can't change your wifi card for some reason, however, then my scripts will at least help alleviate your woes to some extent, but you will probably still experience troubles from time to time that will require restarting the machine, just not as often. Good luck! – Giovanni_Visione Nov 4, 2021 at 5:59

It definitely helps a great deal. I added a few lines to the script because the fixwifi processes would keep piling up for some reason. if [[$(ps aux | grep fixwifi | wc -l) -gt 5]]; then killall fixwifi exit fi – progonkpa Nov 9, 2021 at 11:46

0

The awesome script from Giovanni_Visione didn't quite work for me (even after configuring the variables), but after more searching and trying different things, I simplified it a bit to one that does:

```
#!/bin/bash

# You need to know the designations for your wifi interface.
# You can find these out by running: sudo lshw -C network

# interface name
interface="wlp1s0"

# ssid for your wifi
```

```
ssid="NETGEAR47"

# If this script works, then update cron by:
# "sudo crontab -e" and # add the following, without the initial hash (#) in each
line.
#* * * * * /home/<user>/fixwifi >> /home/<user>/fixwifi.log 2>&1

#################################################################################################


echo "OKAY ------ $(date) ------"


rfkiller=$(sudo rfkill list | grep "yes")
    if [[ $rfkiller != "" ]]  ; then
        echo -e "rfkill - something is blocked"
        sudo rfkill list | sed 's/^/      /'
        echo -e "FIXING - running: rfkill unblock all"
        sudo rfkill unblock all
        sleep 2
    fi

# Check if wifi is okay using: nmcli networking connectivity
# "unknown" and "none" indicate a problem.
connectivity=$(nmcli networking connectivity)
    if [ $connectivity = "unknown" ] || [ $connectivity = "none" ] ; then
        echo "nmcli networking connectivity: $connectivity"
```

Share Improve this answer Follow edited Jun 14 at 20:00 answered Jun 14 at 19:49

healthybodhi
1 1