The network commands chapter explains various tools which can be useful when networking with other computers both within the network and across the internet, obtaining more information about other computers

netstat

Displays contents of /proc/net files. It works with the Linux Network Subsystem, it will tell you what the status of ports are ie.
open, closed, waiting, masquerade connections. It will also display various other things. It has many different options.

tcpdump

This is a sniffer, a program that captures packets off a network interface and interprets them for you. It understands all basic internet protocols, and can be used to save entire packets for later inspection.

ping

The ping command (named after the sound of an active sonar system) sends echo requests to the host you specify on the command line, and lists the responses received their round trip time.

You simply use ping as:

```
ping ip_or_host_name
```

Note to stop ping (otherwise it goes forever) use **CTRL-C** (break).

Please note: Using ping/smbmount/ssh or other UNIX system programs with a computer name rather than IP address will only work if you have the computer listed in your /etc/hosts file. Here is an example:

```
192.168.1.100 new
```

This line says that their is a computer called “new” with IP address 192.168.1.100. Now that it exists in the /etc/hosts file I don't have to type the IP address anymore, just the name “new”.

hostname

Tells the user the host name of the computer they are logged into. Note: may be called *host.*

traceroute

traceroute will show the route of a packet. It attempts to list the series of hosts through which your packets travel on their way to a given destination. Also have a look at *xtraceroute* (one of several graphical equivalents of this program).

Command syntax:

```
traceroute machine_name_or_ip
```

tracepath

tracepath performs a very simlar function to *traceroute* the main difference is that *tracepath* doesn't take complicated options.

Command syntax:

```
tracepath machine_name_or_ip
```

nmap

“ network exploration tool and security scanner”. *nmap* is a very advanced network tool used to query machines (local or remote) as to whether they are up and what ports are open on these machines.

A simple usage example:

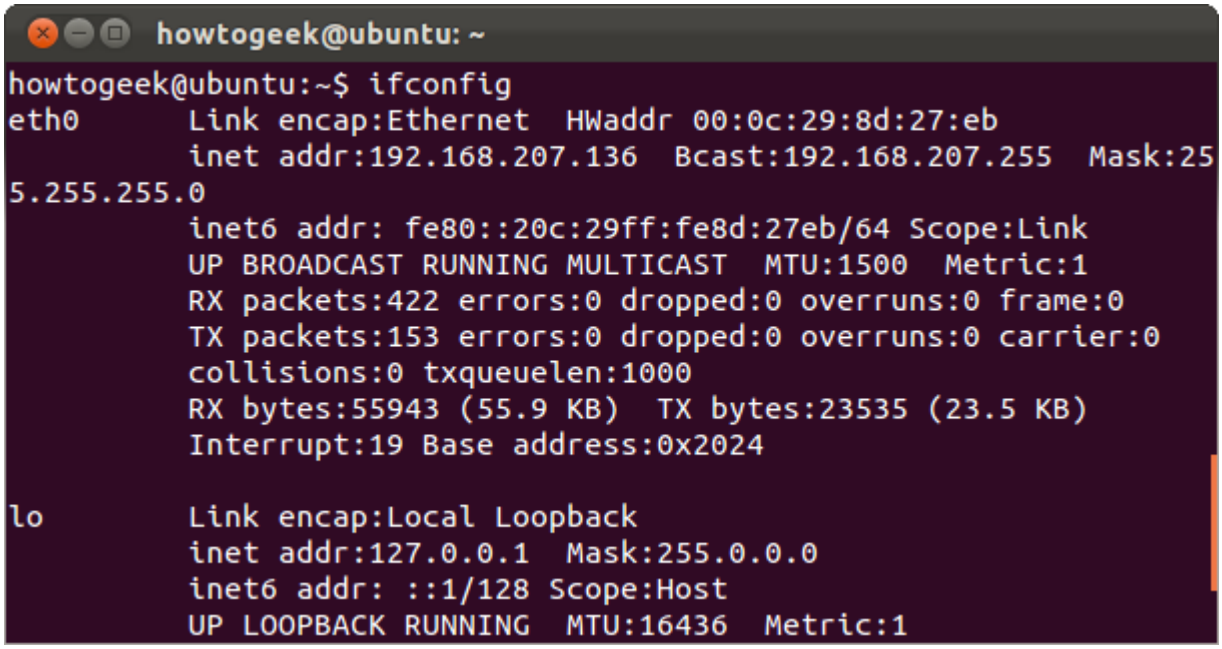
```
nmap machine_name
```

This would query your own machine as to what ports it keeps open. *nmap* is a very powerful tool, documentation is available on the nmap site as well as the information in the manual page.

ifconfig

The **ifconfig** command has a variety of options to configure, tune, and debug your system’s network interfaces. It’s also a quick way to view IP addresses and other network interface information. Type **ifconfig** to view the status of all currently active network interfaces, including their names. You can also specify an interface’s name to view only information about that interface.

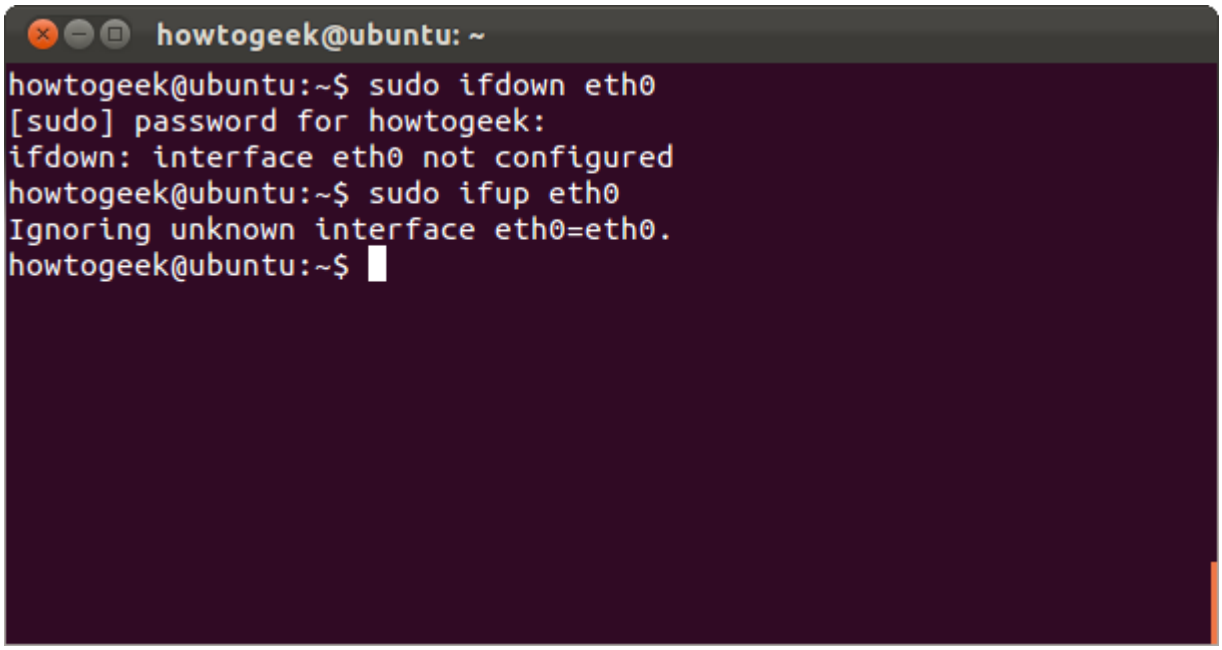
```
ifconfig
ifconfig eth0
```



ifdown & ifup

The **ifdown** and **ifup** commands are the same thing as running **ifconfig up** or **ifconfig down**. Given an interface's name, they take the interface down or bring it up. This requires root permissions, so you have to use sudo on Ubuntu.

```
sudo ifdown eth0
sudo ifup eth0
```



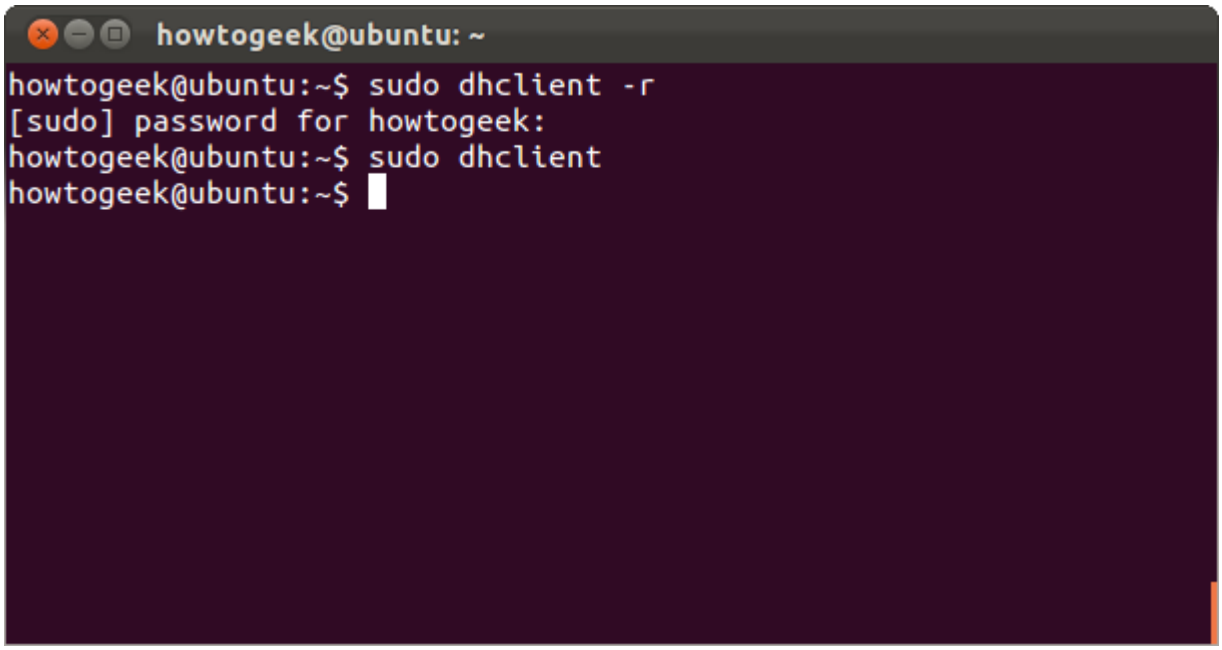
Try this on a Linux desktop system and you'll probably get an error message. Linux desktops usually use NetworkManager, which manages network interfaces for you. These commands will still work on servers without NetworkManager, though.

If you really need to configure NetworkManager from the command line, use the **nmcli**command.

dhclient

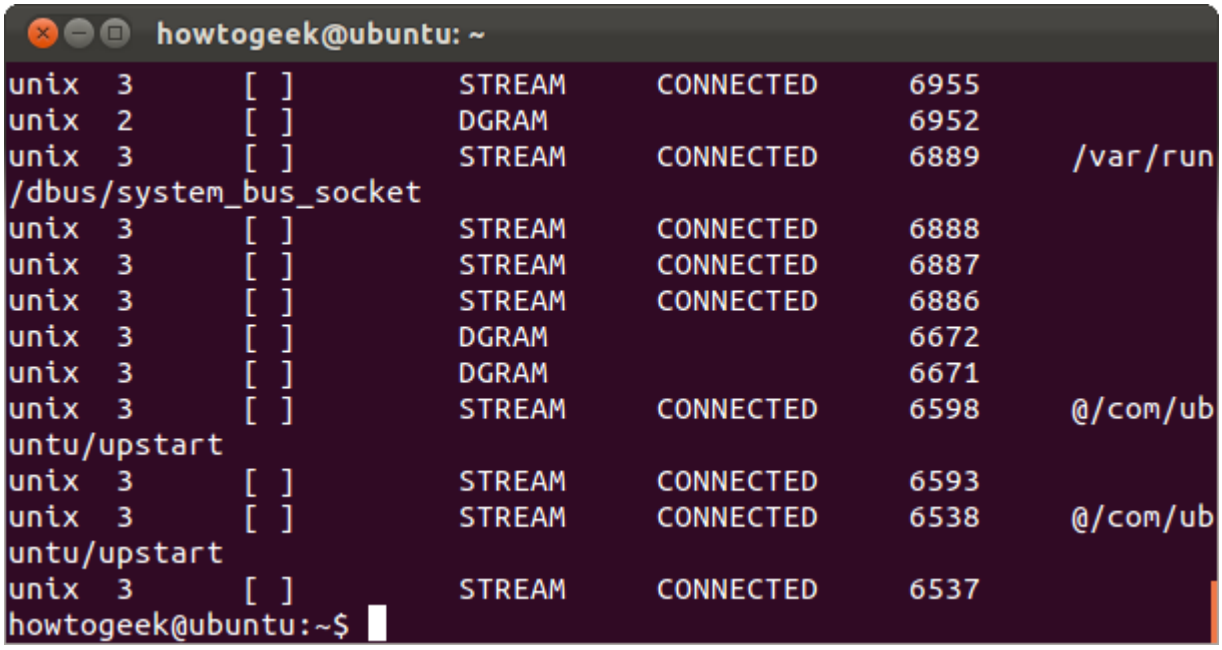
The **dhclient** command can release your computer's IP address and get a new one from your DHCP server. This requires root permissions, so use sudo on Ubuntu. Run dhclient with no options to get a new IP address or use the **-r** switch to release your current IP address.

```
sudo dhclient -r
sudo dhclient
```

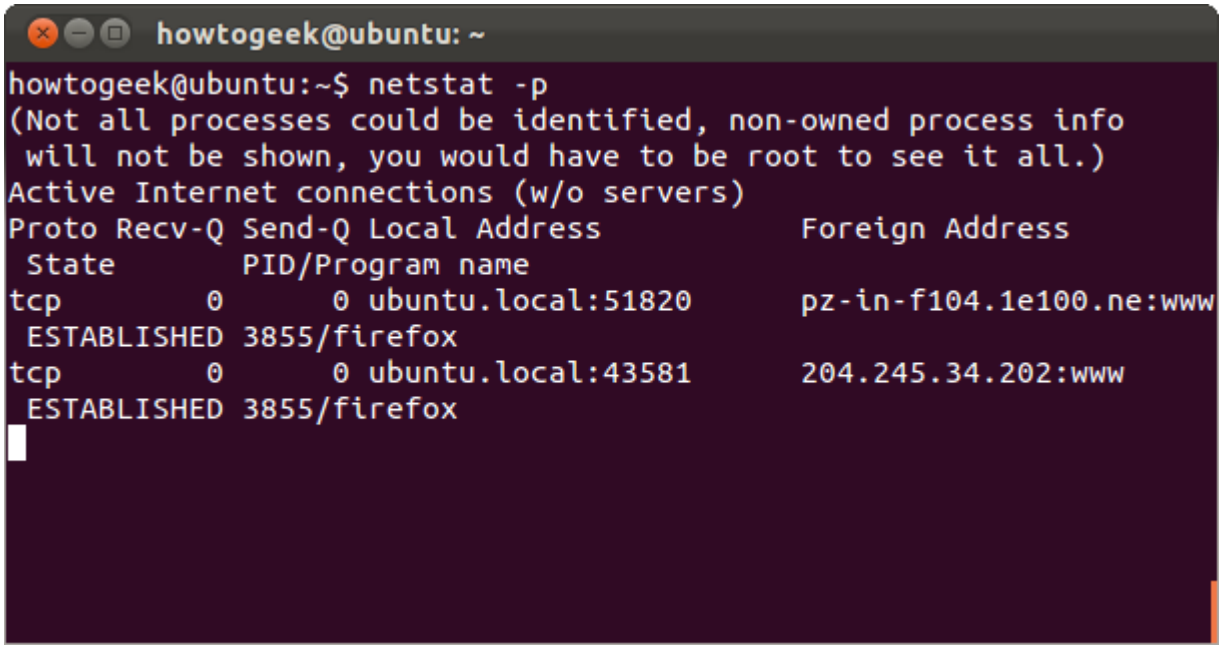


netstat

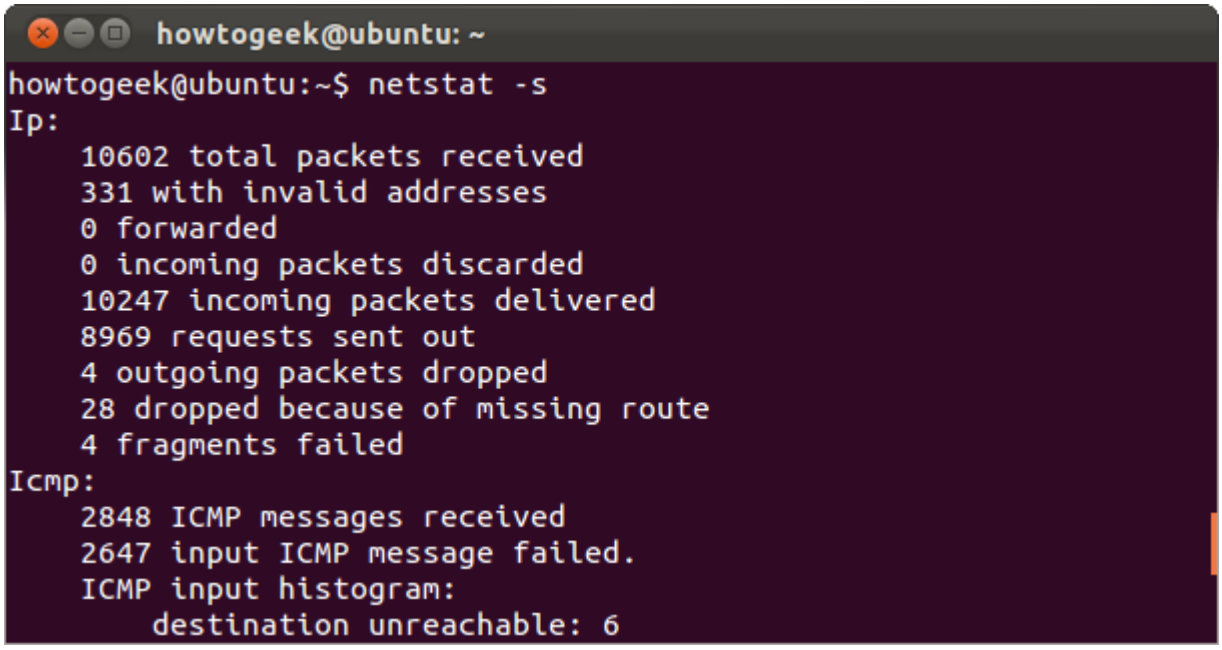
The **netstat** command can show a lot of different interface statistics, including open sockets and routing tables. Run the netstat command with no options and you'll see a list of open sockets.



There's a lot more you can do with this command. For example, use the **netstat -p** command to view the programs associated with open sockets.



View detailed statistics for all ports with **netstat -s**.



ARP Command

ARP (Address Resolution Protocol) is useful to view / add the contents of the kernel's ARP tables. To see default table use the command as.

```
# arp -e
Address                  HWtype  HWaddress           Flags Mask
Iface
192.168.50.1             ether   00:50:56:c0:00:08   C
eth0
```

ETHTOOL Command

ethtool is a replacement of mii-tool. It is to view, setting speed and duplex of your Network Interface Card (NIC). You can set duplex permanently in /etc/sysconfig/network-scripts/ifcfg-eth0 with ETHTOOL_OPTS variable.

```
# ethtool eth0
Settings for eth0:
Current message level: 0x00000007 (7)
Link detected: yes
```

IWCONFIG Command

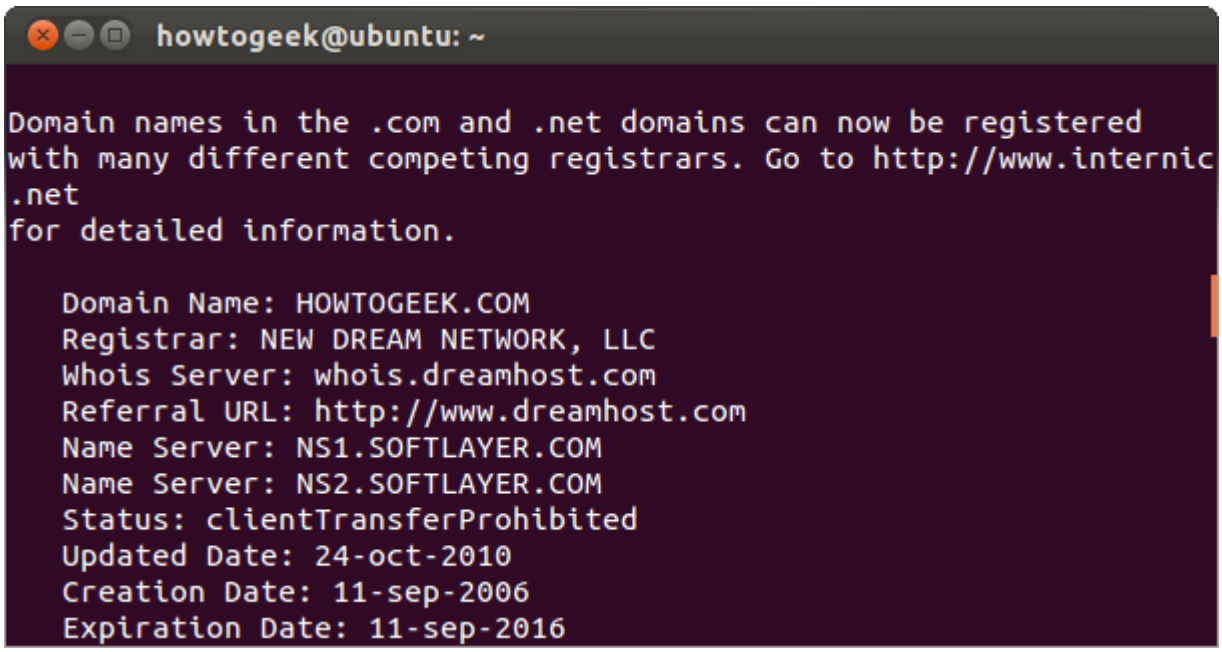
iwconfig command in Linux is use to configure a wireless network interface. You can see and set the basic Wi-Fi details like SSID channel and encryption. You can refer man page of iwconfig to know more.

```
# iwconfig [interface]
```

whois

The **whois** command will show you a website's whois records, so you can view more information about who registered and owns a specific website.

```
whois example.com
```



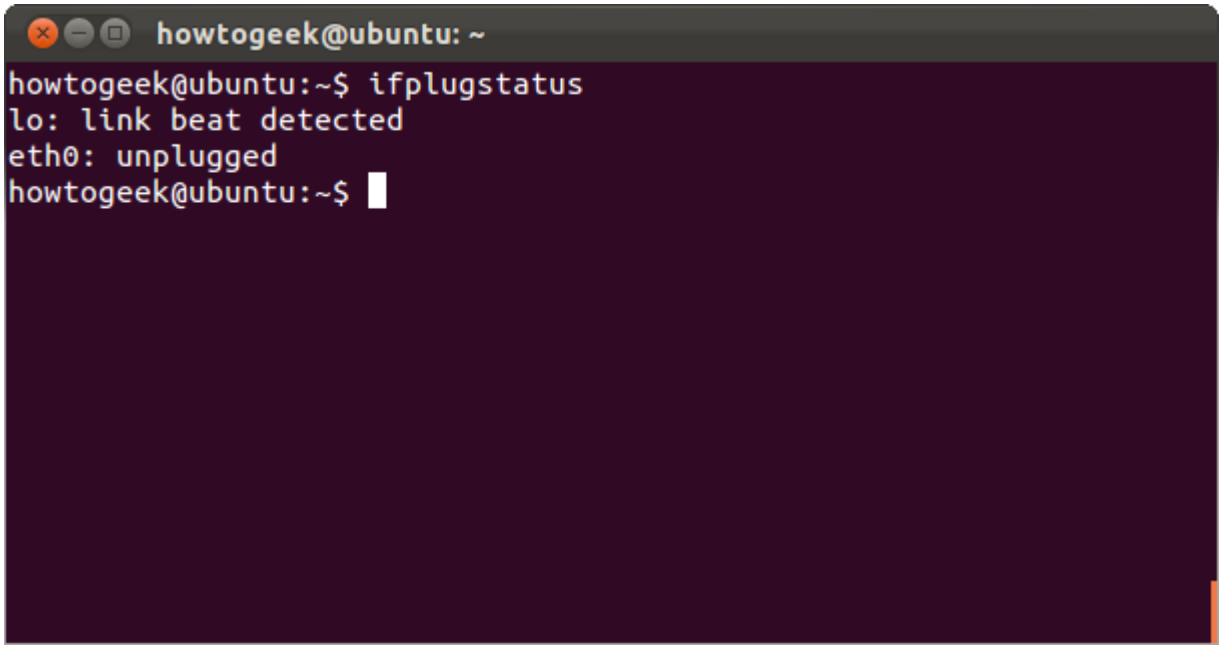
ifplugstatus

The **ifplugstatus** command will tell you whether a cable is plugged into a network interface or not. It isn't installed by default on Ubuntu. Use the following command to install it:

```
sudo apt-get install ifplugd
```

Run the command to see the status of all interfaces or specify a specific interface to view its status.

```
ifplugstatus
ifplugstatus eth0
```



“Link beat detected” means the cable is plugged in. You’ll see “unplugged” if it isn’t.

Iperf -

Iperf is a commonly used network testing tool that can create Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) data streams and measure the throughput of a network that is carrying them. Iperf is a tool for network performance measurement written in C