What's Wrong with the Network?

We all get furious when there is something wrong with the network. There is no fun in this world without being connected to the internet. In this chapter, you will learn the basics of Linux networking. You will also learn how to check network connectivity between two hosts, and gain a practical understanding of how DNS works and much more!

Testing network connectivity

An easy way to check whether you have internet access on your Linux machine is by trying to reach any remote host (server) on the internet. This can be done by using the [ping] command. In general, the syntax of the [ping] command is as follows:

```
ping [options] host
```

For example, to test whether you can reach [google.com], you can run the following command:

```
root@ubuntu-linux:~# ping google.com
PING google.com (172.217.1.14) 56(84) bytes of data.
64 bytes from iad23s25-in-f14.1e100.net (172.217.1.14): icmp_seq=1 ttl=55 time=38.7 ms
64 bytes from iad23s25-in-f14.1e100.net (172.217.1.14): icmp_seq=2 ttl=55 time=38.7 ms
64 bytes from iad23s25-in-f14.1e100.net (172.217.1.14): icmp_seq=3 ttl=55 time=40.4 ms
64 bytes from iad23s25-in-f14.1e100.net (172.217.1.14): icmp_seq=4 ttl=55 time=36.6 ms
64 bytes from iad23s25-in-f14.1e100.net (172.217.1.14): icmp_seq=5 ttl=55 time=40.8 ms
64 bytes from iad23s25-in-f14.1e100.net (172.217.1.14): icmp_seq=6 ttl=55 time=38.6 ms
64 bytes from iad23s25-in-f14.1e100.net (172.217.1.14): icmp_seq=7 ttl=55 time=38.9 ms
64 bytes from iad23s25-in-f14.1e100.net (172.217.1.14): icmp_seq=8 ttl=55 time=37.1 ms
^C
--- google.com ping statistics ---
8 packets transmitted, 8 received, 0% packet loss, time 66ms
rtt min/avg/max/mdev = 36.555/38.724/40.821/1.344 ms
```

The [ping] command sends a packet (unit of data) called an **ICMP echo request** to the specified host and waits for the host to send back a packet called an **ICMP echo reply** to confirm that it did receive the initial packet. If the host replies as we see in our example, then it proves that we were able to reach the host. This is like you sending a package to your friend's house and waiting for your friend to send you a text to confirm that they received it.

Notice that without any options, the [ping] command keeps sending packets continuously, and it won't stop until you hit *Ctrl* + *C*.

You can use the [-c] option to specify the number of packets you want to send to a host. For example, to only send three packets to [google.com], you can run the following command:

```
root@ubuntu-linux:~# ping -c 3 google.com
PING google.com (172.217.1.14) 56(84) bytes of data.

64 bytes from iad23s25-in-f14.1e100.net (172.217.1.14): icmp_seq=1 ttl=55 time=39.3 ms
64 bytes from iad23s25-in-f14.1e100.net (172.217.1.14): icmp_seq=2 ttl=55 time=49.7 ms
64 bytes from iad23s25-in-f14.1e100.net (172.217.1.14): icmp_seq=3 ttl=55 time=40.8 ms

--- google.com ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 59ms rtt min/avg/max/mdev =
39.323/43.267/49.708/4.595 ms
```

If you are not connected to the internet, you will get the following output from the [ping] command:

```
root@ubuntu-linux:~# ping google.com
ping: google.com: Name or service not known
```

Listing your network interfaces

You can list the available network interfaces on your system by viewing the contents of the [/sys/class/net] directory:

```
root@ubuntu-linux:~# ls /sys/class/net
eth0 lo wlan0
```

I have three network interfaces on my system:

1. [eth0]: The Ethernet interface
2. [lo]: The loopback interface
3. [wlan0]: The Wi-Fi interface

Notice that, depending on your computer's hardware, you may get different names for your network interfaces.

The ip command

You can also use the [ip link show] command to view the available network interfaces on your system:

```
root@ubuntu-linux:~# ip link show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT group
default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: eth0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc pfifo_fast state DOWN mode
DEFAULT group default qlen 1000
    link/ether f0:de:f1:d3:e1:e1 brd ff:ff:ff:ff:ff:ff
3: wlan0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP mode DORMANT
group default qlen 1000
    link/ether 10:0b:a9:6c:89:a0 brd ff:ff:ff:ff:ff:ff
```

The nmcli command

Another method that I prefer is using the [nmcli] device status command:

```
root@ubuntu-linux:~# nmcli device status
DEVICE TYPE STATE CONNECTION
wlan0 wifi      connected   SASKTEL0206-5G
eth0  ethernet  unavailable --
lo    loopback  unmanaged   --
```

You can see the connection status of each network interface in the output. I am currently connected to the internet through my Wi-Fi interface.

Checking your IP address

Without a cell phone number, you can't call any of your friends; similarly, your computer needs an IP address to connect to the internet. There are many different ways you can use to check your machine's IP address. You can use the old-school (yet still popular) [ifconfig] command followed by the name of your network interface that is connected to the internet:

```
root@ubuntu-linux:~# ifconfig wlan0
wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
       inet 172.16.1.73 netmask 255.255.255.0 broadcast 172.16.1.255
       inet6 fe80::3101:321b:5ec3:cf9 prefixlen 64 scopeid 0x20<link>
       ether 10:0b:a9:6c:89:a0 txqueuelen 1000 (Ethernet)
       RX packets 265 bytes 27284 (26.6 KiB)
       RX errors 0 dropped 0 overruns 0 frame 0
       TX packets 165 bytes 28916 (28.2 KiB)
       TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

You can also use the [-a] option to list all network interfaces:

```
root@ubuntu-linux:~# ifconfig -a
eth0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
      ether f0:de:f1:d3:e1:e1 txqueuelen 1000 (Ethernet)
      RX packets 0 bytes 0 (0.0 B)
      RX errors 0 dropped 0 overruns 0 frame 0
      TX packets 0 bytes 0 (0.0 B)
      TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
      device interrupt 20 memory 0xf2500000-f2520000

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
     inet 127.0.0.1 netmask 255.0.0.0
     inet6 ::1 prefixlen 128 scopeid 0x10<host>
     loop txqueuelen 1000 (Local Loopback)
     RX packets 4 bytes 156 (156.0 B)
     RX errors 0 dropped 0 overruns 0 frame 0
     TX packets 4 bytes 156 (156.0 B)
     TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
     inet 172.16.1.73 netmask 255.255.255.0 broadcast 172.16.1.255
     inet6 fe80::3101:321b:5ec3:cf9 prefixlen 64 scopeid 0x20<link>
     ether 10:0b:a9:6c:89:a0 txqueuelen 1000 (Ethernet)
     RX packets 482 bytes 45500 (44.4 KiB)
     RX errors 0 dropped 0 overruns 0 frame 0
     TX packets 299 bytes 57788 (56.4 KiB)
     TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

You can see from the output that I am only connected to the internet through my Wi-Fi interface ([wlan0]), and my IP address is [172.16.1.73].

WHAT IS LOOPBACK?[
]Loopback (or [lo]) is a virtual interface that your computer uses to communicate with itself; it is mainly used for troubleshooting purposes. The IP address of the loopback interface is [127.0.0.1], and if you want to ping yourself! Go ahead and ping [127.0.0.1].[
]

You can also use the newer [ip] command to check your machine's IP address. For example, you can run the [ip address show] command to list and show the status of all your network interfaces:

```
root@ubuntu-linux:~# ip address show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc pfifo_fast state
        DOWN link/ether f0:de:f1:d3:e1:e1 brd ff:ff:ff:ff:ff:ff
3: wlan0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state
    UP link/ether 10:0b:a9:6c:89:a0 brd ff:ff:ff:ff:ff:ff
    inet 172.16.1.73/24 brd 172.16.1.255 scope global dynamic
      noprefixroute wlan0 valid_lft 85684sec preferred_lft 85684sec
    inet6 fe80::3101:321b:5ec3:cf9/64 scope link noprefixroute
      valid_lft forever preferred_lft forever
```

Checking your gateway address

Your computer grabs an IP address from a **router**; this router is also referred to as the **default gateway** as it connects you to the outside world (internet). Those routers are everywhere; they are at your house, coffee shops, schools, hospitals, and so on.

You can check the IP address of your default gateway by running any of the following commands:

- [route -n]
- [netstat -rn]
- [ip route]

Let's start with the first command, [route -n]:

```
root@ubuntu-linux:~# route -n Kernel IP routing table
Destination Gateway       Genmask       Flags  Metric Ref Use Iface
0.0.0.0     172.16.1.254  0.0.0.0       UG     600     0   0 wlan0
172.16.1.0  0.0.0.0       255.255.255.0 U      600     0   0 wlan0
```

You can see from the output that my default gateway IP address is [172.16.1.254]. Now let's try the second command, [netstat -rn]:

```
root@ubuntu-linux:~# netstat -rn
Kernel IP routing table
Destination   Gateway      Genmask       Flags  MSS Window irtt Iface
0.0.0.0       172.16.1.254 0.0.0.0       UG     0   0      0    wlan0
172.16.1.0    0.0.0.0      255.255.255.0 U      0   0      0    wlan0
```

The output almost looks identical. Now the output differs a little bit with the third command, [ip route]:

```
root@ubuntu-linux:~# ip route
default via 172.16.1.254 dev wlan0 proto dhcp metric 600
172.16.1.0/24 dev wlan0 proto kernel scope link src 172.16.1.73 metric 600
```

The default gateway IP address is displayed on the first line: default via [172.16.1.254]. You should also be able to ping your default gateway:

```
root@ubuntu-linux:~# ping -c 2 172.16.1.254
PING 172.16.1.254 (172.16.1.254) 56(84) bytes of data.
64 bytes from 172.16.1.254: icmp_seq=1 ttl=64 time=1.38 ms
64 bytes from 172.16.1.254: icmp_seq=2 ttl=64 time=1.62 ms

--- 172.16.1.254 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 3ms rtt min/avg/max/mdev =
1.379/1.501/1.624/0.128 ms
```

Flying with traceroute

You are now ready to leave your house to go to work. You must go through different streets that eventually lead to your destination, right? Well, this is very similar to when you try to reach a host (website) on the internet; there is a route that you take that starts with your default gateway and ends with your destination.

You can use the [traceroute] command to trace the route to any destination. The general syntax of the [traceroute] command is as follows:

```
traceroute destination
```

For example, you can trace the route from your machine to [google.com] by running the following command:

```
root@ubuntu-linux:~# traceroute google.com
traceroute to google.com (172.217.1.14), 30 hops max, 60 byte packets
 1 172.16.1.254 (172.16.1.254) 15.180 ms 15.187 ms 15.169 ms
 2 207-47-195-169.ngai.static.sasknet.sk.ca (207.47.195.169) 24.059 ms
 3 142.165.0.110 (142.165.0.110) 50.060 ms 54.305 ms 54.903 ms
 4 72.14.203.189 (72.14.203.189) 53.720 ms 53.997 ms 53.948 ms
 5 108.170.250.241 (108.170.250.241) 54.185 ms 35.506 ms 108.170.250.225
 6 216.239.35.233 (216.239.35.233) 37.005 ms 35.729 ms 38.655 ms
 7 yyz10s14-in-f14.1e100.net (172.217.1.14) 41.739 ms 41.667 ms 41.581 ms
```

As you can see, my machine took seven trips (hops) to reach my final destination, [google.com]. Notice the first hop is my default gateway, and the last hop is the destination.

The [traceroute] command comes in handy when you are troubleshooting connectivity issues. For example, it may take you a very long time to reach a specific destination; in this case, [traceroute] can help you detect any points of failure on the path to your destination.

Breaking your DNS

Every website (destination) on the internet must have an IP address. However, we humans are not very good with numbers so we have invented the **Domain Name System** (**DNS**). The primary function of the DNS is that it associates a name (domain name) with an IP address; this way, we don't need to memorize IP addresses while browsing the internet ... thank God for the DNS!

Every time you enter a domain name on your browser, the DNS translates (resolves) the domain name to its corresponding IP address. The IP address of your DNS server is stored in the file [/etc/resolv.conf]:

```
root@ubuntu-linux:~# cat /etc/resolv.conf
# Generated by NetworkManager
nameserver 142.165.200.5
```

I am using the DNS server [142.165.200.5], which is provided by my **Internet Service Provider** (**ISP**). You can use the [nslookup] command to see DNS in action. The general syntax of the [nslookup] command is as follows:

```
nslookup domain_name
```

The [nslookup] command uses DNS to obtain the IP address of a domain name. For example, to get the IP address of [facebook.com], you can run the following command:

```
root@ubuntu-linux:~# nslookup facebook.com
Server: 142.165.200.5
Address: 142.165.200.5#53

Non-authoritative answer:
Name: facebook.com
Address: 157.240.3.35
Name: facebook.com
Address: 2a03:2880:f101:83:face:b00c:0:25de
```

Notice it displayed the IP address of my DNS server in the first line of the output. You can also see the IP address [157.240.3.35] of [facebook.com].

You can also ping [facebook.com]:

```
root@ubuntu-linux:~# ping -c 2 facebook.com
PING facebook.com (157.240.3.35) 56(84) bytes of data.
64 bytes from edge-star-mini-shv-01-sea1.facebook.com (157.240.3.35):
icmp_seq=1 ttl=55 time=34.6 ms
64 bytes from edge-star-mini-shv-01-sea1.facebook.com (157.240.3.35):
icmp_seq=2 ttl=55 time=33.3 ms

--- facebook.com ping statistics ---

2 packets transmitted, 2 received, 0% packet loss, time 2ms
rtt min/avg/max/mdev = 33.316/33.963/34.611/0.673 ms
```

Now let's break things! My mum once told me that I have to break things so I can understand how they work. Let's see what life is without DNS by emptying the file [/etc/resolv.conf]:

```
root@ubuntu-linux:~# echo > /etc/resolv.conf
root@ubuntu-linux:~# cat /etc/resolv.conf

root@ubuntu-linux:~#
```

Now let's do [nslookup] on [facebook.com]:

```
root@ubuntu-linux:~# nslookup facebook.com
```

You will see that it hangs as it is unable to resolve domain names anymore. Now let's try to ping [facebook.com]:

```
root@ubuntu-linux:~# ping facebook.com
ping: facebook.com: Temporary failure in name resolution
```

You get the error message [Temporary failure in name resolution], which is a fancy way of saying that your DNS is broken! However, you can still ping [facebook.com] by using its IP address:

```
root@ubuntu-linux:~# ping -c 2 157.240.3.35
PING 157.240.3.35 (157.240.3.35) 56(84) bytes of data.
64 bytes from 157.240.3.35: icmp_seq=1 ttl=55 time=134 ms
64 bytes from 157.240.3.35: icmp_seq=2 ttl=55 time=34.4 ms

--- 157.240.3.35 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 2ms
rtt min/avg/max/mdev = 34.429/84.150/133.872/49.722 ms
```

Let's fix our DNS, but this time we will not use the DNS server of our ISP; instead, we will use Google's public DNS server [8.8.8.8]:

```
root@ubuntu-linux:~# echo "nameserver 8.8.8.8" > /etc/resolv.conf
root@ubuntu-linux:~# cat /etc/resolv.conf
nameserver 8.8.8.8
```

Now let's do an [nslookup] on [facebook.com] again:

```
root@ubuntu-linux:~# nslookup facebook.com Server: 8.8.8.8
Address: 8.8.8.8#53

Non-authoritative answer:
Name: facebook.com
Address: 31.13.80.36
Name: facebook.com
Address: 2a03:2880:f10e:83:face:b00c:0:25de
```

Notice that my active DNS is now changed to [8.8.8.8]. I also got a different IP address for [facebook.com], and that's because Facebook is running on many different servers located in various regions of the world.

Changing your hostname

Every website has a domain name that uniquely identifies it over the internet; similarly, a computer has a hostname that uniquely identifies it over a network.

Your computer's hostname is stored in the file [/etc/hostname]:

```
root@ubuntu-linux:~# cat /etc/hostname
ubuntu-linux
```

You can use hostnames to reach other computers in the same network (subnet). For example, I have another computer with the hostname [backdoor] that is currently running, and I can ping it:

```
root@ubuntu-linux:~# ping backdoor
PING backdoor (172.16.1.67) 56(84) bytes of data.
64 bytes from 172.16.1.67 (172.16.1.67): icmp_seq=1 ttl=64 time=3.27 ms
```

```
64 bytes from 172.16.1.67 (172.16.1.67): icmp_seq=2 ttl=64 time=29.3 ms
64 bytes from 172.16.1.67 (172.16.1.67): icmp_seq=3 ttl=64 time=51.4 ms
^C
--- backdoor ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 20ms
rtt min/avg/max/mdev = 3.272/27.992/51.378/19.662 ms
```

Notice that [backdoor] is on the same network (subnet) and has an IP address of [172.16.1.67]. I can also ping myself:

```
root@ubuntu-linux:~# ping ubuntu-linux
PING ubuntu-linux (172.16.1.73) 56(84) bytes of data.
64 bytes from 172.16.1.73 (172.16.1.73): icmp_seq=1 ttl=64 time=0.025 ms
64 bytes from 172.16.1.73 (172.16.1.73): icmp_seq=2 ttl=64 time=0.063 ms
^C
--- ubuntu-linux ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 14ms
rtt min/avg/max/mdev = 0.025/0.044/0.063/0.019 ms
```

That's a smart way of displaying your computer's IP address -- simply ping yourself!

You can use the [hostnamectl] command to view and set your computer's hostname:

```
root@ubuntu-linux:~# hostnamectl
    Static hostname: ubuntu-linux
          Icon name: computer-vm
            Chassis: vm
         Machine ID: 106fd80252e541faafa4e54a250d1216
            Boot ID: c5508514af114b4b80c55d4267c25dd4
     Virtualization: oracle
   Operating System: Ubuntu 18.04.3 LTS
             Kernel: Linux 4.15.0-66-generic
       Architecture: x86-64
```

To change your computer's hostname, you can use the [hostnamectl set-hostname] command followed by the new hostname:

```
hostnamectl set-hostname new_hostname
```

For example, you can change the hostname of your computer to [myserver] by running the following command:

```
root@ubuntu-linux:~# hostnamectl set-hostname myserver
root@ubuntu-linux:~# su -
root@myserver:~#
```

Keep in mind that you need to open a new shell session so that your shell prompt displays the new hostname. You can also see that the file [/etc/hostname] is updated as it contains the new hostname:

```
root@ubuntu-linux:~# cat /etc/hostname
myserver
```

Restarting your network interface

It's probably an abused method, but sometimes doing a restart is the quickest fix to many computer-related troubles! I myself am guilty of overusing the restart solution for most of my computer problems.

You can use the [ifconfig] command to bring down (disable) a network interface; you have to follow the network interface name with the [down] flag as follows:

```
ifconfig interface_name down
```

For example, I can bring down my Wi-Fi interface, [wlan0], by running the following command:

```
root@myserver:~# ifconfig wlan0 down
```

You can use the [up] flag to bring up (enable) a network interface:

```
ifconfig interface_name up
```

For example, I can bring back up my Wi-Fi interface by running the following command:

```
root@myserver:~# ifconfig wlan0 up
```

You may also want to restart all your network interfaces at the same time. This can be done by restarting the [NetworkManager] service as follows:

```
root@myserver:~# systemctl restart NetworkManager
```

Now it's time to test your understanding of Linux networking with a lovely knowledge-check exercise.

Knowledge check

For the following exercises, open up your Terminal and try to solve the following tasks:

1. Change your hostname to [darkarmy].
2. Display the IP address of your default gateway.
3. Trace the route from your machine to [www.ubuntu.com].
4. Display the IP address of your DNS.
5. Display the IP address of [www.distrowatch.com].
6. Bring down your Ethernet interface.
7. Bring your Ethernet interface back up.