

# Networking and Network Interfaces

 [coursera.org/learn/linux-for-developers/supplement/A4S3i/networking-and-network-interfaces](https://coursera.org/learn/linux-for-developers/supplement/A4S3i/networking-and-network-interfaces)

The vast majority of network programming in Linux is done using the socket interface. Thus, standards-compliant programs should require little massaging to work properly with Linux.

Note, however, there are many enhancements and new features in the Linux networking implementation, such as new kinds of address and protocol families. For example, Linux offers the **netlink** interface, which permits opening up socket connections between kernel sub-systems and applications (or other kernel sub-systems). This has been effectively deployed to implement firewall and routing applications.

Historically, the wired Ethernet network devices have been known by a name such as **eth0**, **eth1**, etc., while wireless devices have had names like **wlan0**, **wlan1**, etc.

Basic information about active network interfaces on your system is gathered through both the **ip** utility, or the older **ifconfig** program:

33  
29  
30  
31  
32  
26  
27  
28  
23  
24  
25  
19  
20  
21  
22  
16  
17  
18  
13  
14  
15  
9  
10  
11  
12  
6  
7  
8  
3  
4  
5

```

1
2
    TX errors 0   dropped 0 overruns 0   carrier 0   collisions 0

    loop  txqueuelen 1000   (Local Loopback)

    RX packets 403   bytes 30112 (29.4 KiB)

    RX errors 0   dropped 0   overruns 0   frame 0

    TX packets 403   bytes 30112 (29.4 KiB)
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>

    TX errors 0   dropped 0 overruns 0   carrier 0   collisions 0

    device interrupt 20  memory 0xf7d00000-f7d20000


    ether 08:62:66:45:4d:16  txqueuelen 1000   (Ethernet)

    RX packets 5445514  bytes 7591143594 (7.0 GiB)

    RX errors 0   dropped 0   overruns 0   frame 0

    TX packets 2588767  bytes 743093891 (708.6 MiB)
eno1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500

    inet 192.168.1.200  netmask 255.255.255.0  broadcast 192.168.1.255

    inet6 fe80::44f9:65c:64f8:f842  prefixlen 64  scopeid 0x20<link>

    742798233  2586311  0          0          0          0

$ /sbin/ifconfig

    link/ether 08:62:66:45:4d:16 brd ff:ff:ff:ff:ff:ff

    RX: bytes  packets  errors  dropped overrun mcast

    7563473070 5424565  0      0      0      33762

    TX: bytes  packets  errors  dropped carrier collsns

    TX: bytes  packets  errors  dropped carrier collsns

    30112      403      0      0      0      0

2: eno1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP mode DEFAULT group default qlen 1000

    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00

    RX: bytes  packets  errors  dropped overrun mcast

    30112      403      0      0      0      0

$ ip -s link

1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT group default qlen 1000

```



Information displayed includes information about the hardware MAC address, the MTU (maximum transfer unit), and the IRQ the device is tied to. Also displayed are the number of packets and bytes transmitted, received, or resulting in errors.

This machine has a network card bound to **eno1**, and the loopback interface, **lo**, which handles network traffic bound to the machine. Note you can see the statistical information in abbreviated form by looking at **/proc/net/dev**, and in one quantity per line display in **/sys/class/net/etho/statistics**:

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26

```
$ ls -l /sys/class/net/en01/statistics
```

```
total 0
```

```
-r--r--r-- 1 root root 4096 Feb 10 11:55 collisions
-r--r--r-- 1 root root 4096 Feb 10 11:55 multicast
-r--r--r-- 1 root root 4096 Feb 10 11:55 rx_bytes
-r--r--r-- 1 root root 4096 Feb 10 11:55 rx_compressed
-r--r--r-- 1 root root 4096 Feb 10 11:55 rx_crc_errors
-r--r--r-- 1 root root 4096 Feb 10 11:55 rx_dropped
-r--r--r-- 1 root root 4096 Feb 10 11:55 rx_errors
-r--r--r-- 1 root root 4096 Feb 10 11:55 rx_fifo_errors
-r--r--r-- 1 root root 4096 Feb 10 11:55 rx_frame_errors
-r--r--r-- 1 root root 4096 Feb 10 11:55 rx_length_errors
-r--r--r-- 1 root root 4096 Feb 10 11:55 rx_missed_errors
-r--r--r-- 1 root root 4096 Feb 10 11:55 rx_nohandler
-r--r--r-- 1 root root 4096 Feb 10 11:55 rx_over_errors
```

```
-r--r--r-- 1 root root 4096 Feb 10 11:55 rx_packets
-r--r--r-- 1 root root 4096 Feb 10 11:55 tx_aborted_errors
-r--r--r-- 1 root root 4096 Feb 10 11:55 tx_bytes
-r--r--r-- 1 root root 4096 Feb 10 11:55 tx_carrier_errors
-r--r--r-- 1 root root 4096 Feb 10 11:55 tx_compressed
-r--r--r-- 1 root root 4096 Feb 10 11:55 tx_dropped
-r--r--r-- 1 root root 4096 Feb 10 11:55 tx_errors
-r--r--r-- 1 root root 4096 Feb 10 11:55 tx_fifo_errors
-r--r--r-- 1 root root 4096 Feb 10 11:55 tx_heartbeat_errors
-r--r--r-- 1 root root 4096 Feb 10 11:55 tx_packets
-r--r--r-- 1 root root 4096 Feb 10 11:55 tx_window_errors
```

