Don Bosco Institute of Technology, Kurla
Academic Year 2024-25

EXPERIMENT NO. 1

SEMESTER: V **DATE OF PERFORMANCE: 10th July 2024**

SUBJECT: CN Lab **DATE OF SUBMISSION: 13th July 2024**

NAME OF THE STUDENT: David James Eluvathingal **ROLL NO.: 22**

AIM	Use basic networking commands in Linux
LEARNING OBJECTIVE	The student will understand the basic Linux networking commands.
LEARNING OUTCOME	● The student will experiment and explain the basic Linux networkingcommands. ● The student will be able to find out IP/MAC addresses, IP packetstatus, link status, network statistics, port scanning etc.
COURSE OUTCOME	**CSL 502.1:** Identify the important networking commands in Linux and understand their function.
PROGRAM OUTCOME	PO1,PO2,PO3,PO4,PO5,PO9,PO10,PSO1,PSO2,PSO3
BLOOM'S TAXONOMY LEVEL	Remember
THEORY	**1. ifconfig:** ifconfig is used to configure the system's kernel-resident network interfaces. Itis used at boot time to set up interfaces as necessary. After that, it is usually only needed when debugging or when system tuning is needed. If no arguments are given, ifconfig displays the status of the system's active interfaces. If a single interface argument is given, it displays the status of the given interface only. ● In the command prompt, type ifconfig. Running ifconfig with nooptions will display the configuration of all active interfaces. ● Attach output. **2. ping:** Ping is a basic Internet program that lets you verify that a particular Internet address exists and can accept requests. The verb ping means the act of using the ping utility or command. Ping is used diagnostically to ensure that a host computer you arr trying to reach is actually operating. If, for example, a user cannot ping a host, then the user will be unable to use the File Transfer Protocol (FTP) to send files to that host. Ping can also be used with a host thatis operating to see how long it takes to get a response back. Using ping, you can learn the number form of the IP address from the symbolic domain name Loosely, ping means "to get the attention of" or "to check for the presence of"another party online". Ping operates by sending a packet to a designated address and waiting for a response.

- Ask your friend to give his/her IP address.
- Now try a simple ping to their machine using e.g. **ping** your friend'sIP address.
- Attach output.
- Try the option ping -c 2 IP ADDRESS, then try ping -c 7 IPADDRESS. What differences do you notice?
- Attach output.

3. traceroute:
traceroute prints the route that packets take to a network host. It is used to findnetwork paths from machine to server.

- In the command prompt type traceroute www.dbit.in (Take website ofyour choice instead of www.dbit.com)
- Attach output of both website.

4. tracepath:
tracepath traces the complete path to a networking host discovering the MTUalong the path. It uses UDP port or some random port. It is similar to traceroute, only it does not require super user privileges and has no fancy options.

- In the command prompt type tracepath www.dbit.in (Take website ofyour choice instead of www.dbit.com)
- Attach output of both website.
- What difference do you see between traceroute and tracepathcommand?

5. nslookup:
nslookup is a network administration command-line tool available for many computer operating systems for querying the Domain Name System (DNS) toobtain domain name or IP address mapping or for any other specific DNS record.

- In the command prompt Type nslookup www.yahoo.com (Takewebsite of your choice instead of www.yahoo.com)
- Note that this command gives you the actual name of the server, as perthe hosting company's naming conventions; its IP address; and any aliases under which that server operates.
- Attach output.

6. Netstat:
Netstat allows you to display statistics about your Ethernet interface. If anyerrors are indicated in the display, you might have problems with your network connection that are slowing the network down. If the error packetsapproach 1% of the total number of packets, something is probably wrong with your NIC or physical interface.

- In the command prompt, type in netstat to list all current networkconnections, not just inbound but outbound as well.
- Try atleast 10 variations of netstat command doing specific operations.
- Attach output.

7. ARP:
ARP command is used to view and then delete the ARP cache, and you use the ping command to generate ARP cache entries. Address Resolution Protocol (ARP) is a telecommunications protocol used for resolution of network layer addresses into link layer addresses, a critical function in multiple-access networks. ARP was defined by RFC 826 and is also the nameof the program for manipulating these addresses in most operating systems. In the command prompt, type arp -a. Remember, that previously the computerdiscovered the MAC address of your computer by using address resolution protocol (ARP). You have now resolved the globally unique MAC address ofyour device.

- In the command prompt, type in ARP.
- Attach output.

8.ip addr show:
ip addr show command is used to view ip addresses.

- In the command prompt, type ip addr show.
- Attach output.

9.dig:
Using dig command you can query DNS name server for your DNS lookuprelated tasks.

- In the command prompt type dig dbit.in (Take website of your choiceinstead of www.dbit.com)
- The dig command result will have header, question section, answersection, authority section and additional section.
- Attach output of both website.

10. route:
route command will display the routing table entries.

- In the command prompt type route. This will display the routing tableentries.
- Find an alternative command for the same task.
- What does route –n command do?
- Attach output for both the commands executed.

LAB EXERCISE

- Every student will execute the command as per the instruction andattach output as per the requirements.
- **Students will execute atleast three more networking commands(other than the ones listed in the document) commonly used, append its theory and attach the output.**
- **Append all snapshots here.**

SNAPSHOTS:

- **Ifconfig**

```
dwayne-nixon@dwayne-nixon-V1-04:~$ ifconfig
enp4s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.177  netmask 255.255.255.0  broadcast 192.168.0.255
        inet6 fe80::e6a8:dfff:fefd:8c85  prefixlen 64  scopeid 0x20<link>
        ether e4:a8:df:fd:8c:85  txqueuelen 1000  (Ethernet)
        RX packets 1076  bytes 686680 (686.6 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 1045  bytes 203720 (203.7 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

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 235  bytes 24492 (24.4 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 235  bytes 24492 (24.4 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wlp5s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.121  netmask 255.255.255.0  broadcast 192.168.0.255
        inet6 fe80::7139:148c:3475:2ef  prefixlen 64  scopeid 0x20<link>
        ether c0:3c:59:fa:5a:7d  txqueuelen 1000  (Ethernet)
        RX packets 198  bytes 34805 (34.8 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 165  bytes 21413 (21.4 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

- **Ping**

```
dwayne-nixon@dwayne-nixon-V1-04:~$ ping google.com
PING google.com (142.250.182.238) 56(84) bytes of data.
64 bytes from bom07s29-in-f14.1e100.net (142.250.182.238): icmp_seq=1 ttl=60 time=2.47 ms
64 bytes from bom07s29-in-f14.1e100.net (142.250.182.238): icmp_seq=2 ttl=60 time=2.83 ms
64 bytes from bom07s29-in-f14.1e100.net (142.250.182.238): icmp_seq=3 ttl=60 time=2.13 ms
64 bytes from bom07s29-in-f14.1e100.net (142.250.182.238): icmp_seq=4 ttl=60 time=2.86 ms
64 bytes from bom07s29-in-f14.1e100.net (142.250.182.238): icmp_seq=5 ttl=60 time=2.04 ms
64 bytes from bom07s29-in-f14.1e100.net (142.250.182.238): icmp_seq=6 ttl=60 time=2.87 ms
64 bytes from bom07s29-in-f14.1e100.net (142.250.182.238): icmp_seq=7 ttl=60 time=2.42 ms
64 bytes from bom07s29-in-f14.1e100.net (142.250.182.238): icmp_seq=8 ttl=60 time=2.14 ms
64 bytes from bom07s29-in-f14.1e100.net (142.250.182.238): icmp_seq=9 ttl=60 time=2.69 ms
64 bytes from bom07s29-in-f14.1e100.net (142.250.182.238): icmp_seq=10 ttl=60 time=2.43 ms
64 bytes from bom07s29-in-f14.1e100.net (142.250.182.238): icmp_seq=11 ttl=60 time=2.87 ms
64 bytes from bom07s29-in-f14.1e100.net (142.250.182.238): icmp_seq=12 ttl=60 time=2.43 ms
```

```
dwayne-nixon@dwayne-nixon-V1-04:~$ ping -c 2 google.com
PING google.com (142.250.182.238) 56(84) bytes of data.
64 bytes from bom07s29-in-f14.1e100.net (142.250.182.238): icmp_seq=1 ttl=60 time=2.50 ms
64 bytes from bom07s29-in-f14.1e100.net (142.250.182.238): icmp_seq=2 ttl=60 time=2.59 ms

--- google.com ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 2.499/2.544/2.589/0.045 ms
dwayne-nixon@dwayne-nixon-V1-04:~$ ping -c 7 google.com
PING google.com (142.250.182.238) 56(84) bytes of data.
64 bytes from bom07s29-in-f14.1e100.net (142.250.182.238): icmp_seq=1 ttl=60 time=2.48 ms
64 bytes from bom07s29-in-f14.1e100.net (142.250.182.238): icmp_seq=2 ttl=60 time=1.97 ms
64 bytes from bom07s29-in-f14.1e100.net (142.250.182.238): icmp_seq=3 ttl=60 time=2.53 ms
64 bytes from bom07s29-in-f14.1e100.net (142.250.182.238): icmp_seq=4 ttl=60 time=2.97 ms
64 bytes from bom07s29-in-f14.1e100.net (142.250.182.238): icmp_seq=5 ttl=60 time=2.02 ms
64 bytes from bom07s29-in-f14.1e100.net (142.250.182.238): icmp_seq=6 ttl=60 time=2.44 ms
64 bytes from bom07s29-in-f14.1e100.net (142.250.182.238): icmp_seq=7 ttl=60 time=3.43 ms

--- google.com ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6009ms
rtt min/avg/max/mdev = 1.971/2.547/3.425/0.475 ms
```

The ping command is used to test the reachability of a host on a network and measure the round-trip time for messages sent to the destination. The -c option specifies the number of ping requests to send.
The main differences between -c 2 and -c 7 are as follows:

1. **Number of Requests Sent:**
   ping -c 2 IP_ADDRESS: Sends 2 requests.
   ping -c 7 IP_ADDRESS: Sends 7 requests.
2. **Total Time Taken:**
   Since ping -c 7 IP_ADDRESS sends more requests, it will take longer to complete compared to ping -c 2 IP_ADDRESS.
3. **Statistics:**
   The summary of ping statistics at the end will show different numbers of packets transmitted and received, as well as different packet loss percentages if any packets are lost.
   The round-trip time (rtt) statistics will be calculated based on 2 pings in the first case and 7 pings in the second case. This includes minimum, average, maximum, and mean deviation of the round-trip times.

- **Traceroute**

```
dwayne-nixon@dwayne-nixon-V1-04:~$ traceroute www.google.com
traceroute to www.google.com (142.250.66.4), 64 hops max
  1   192.168.0.1  0.346ms  0.286ms  0.304ms
  2   172.22.3.18  1.781ms  0.926ms  0.626ms
  3   172.22.3.17  1.771ms  1.967ms  1.482ms
  4   103.71.17.42  1.053ms  2.199ms  2.041ms
  5   103.71.17.41  1.577ms  2.481ms  1.593ms
  6   72.14.204.16  4.678ms  5.166ms  5.318ms
  7   *  *  *
  8   72.14.239.246  4.448ms  4.522ms  3.136ms
  9   192.178.110.110  13.115ms  32.601ms  8.635ms
 10   142.250.209.71  3.920ms  3.834ms  3.938ms
 11   142.250.66.4  2.902ms  3.253ms  2.668ms
```

```
dwayne-nixon@dwayne-nixon-V1-04:~$ traceroute www.dbit.in
traceroute to www.dbit.in (43.205.151.144), 64 hops max
  1   192.168.0.1  0.382ms  0.242ms  0.233ms
  2   172.22.3.18  0.809ms  0.885ms  0.969ms
  3   172.22.3.17  1.672ms  1.596ms  1.043ms
  4   103.71.17.42  1.732ms  0.881ms  1.086ms
  5   103.71.17.41  1.248ms  1.927ms  2.170ms
  6   103.77.108.148  27.259ms  3.317ms  2.365ms
  7   *  *  *
  8   *  *  *
```

- **Tracepath**

```
dwayne-nixon@dwayne-nixon-V1-04:~$ tracepath www.dbit.in
 1?: [LOCALHOST]                      pmtu 1500
 1:  _gateway                                              0.689ms
 1:  _gateway                                              0.583ms
 2:  _gateway                                              0.898ms pmtu 1480
 2:  172.22.3.18                                           1.395ms
 3:  172.22.3.17                                           1.834ms
 4:  ???                                                   1.952ms
 5:  ???                                                   2.286ms asymm  3
 6:  as16509.bom.extreme-ix.net                            2.807ms asymm  5
 7:  no reply
 8:  no reply
```

```
dwayne-nixon@dwayne-nixon-V1-04:~$ tracepath www.google.com
 1?: [LOCALHOST]                      pmtu 1500
 1:  _gateway                                              0.635ms
 1:  _gateway                                              0.365ms
 2:  _gateway                                              0.329ms pmtu 1480
 2:  172.22.3.18                                           1.618ms
 3:  172.22.3.17                                           2.488ms
 4:  ???                                                   1.854ms
 5:  ???                                                   2.329ms asymm  3
 6:  72.14.204.16                                          5.755ms asymm  5
 7:  no reply
 8:  no reply
```

Both traceroute and tracepath are network diagnostic tools used to track the path packets take from a source to a destination host across an IP network
Following are the differences between the two commands:

1. Availability:

- traceroute is available on most Unix-like operating systems by default.
- tracepath is part of the iputils package, which is often installed by default on many Linux distributions but not necessarily on all Unix-like systems.

2. Permissions:

- traceroute often requires superuser (root) privileges to execute because it typically uses raw sockets or the UDP protocol to send packets.
- tracepath usually does not require superuser privileges as it uses the SOCK_DGRAM socket type and the ICMP protocol.

3. Default Protocol:

- traceroute uses UDP packets by default, but it can also be configured to use ICMP Echo Request or TCP SYN packets.
- tracepath uses UDP packets with increasing TTL (Time to Live) values but falls back to ICMP Echo Request packets if necessary.

4. Output:

- The output of traceroute includes detailed information about each hop, including the round-trip time for packets to each hop.
- tracepath provides a simpler output, typically focusing on the path MTU (Maximum Transmission Unit) discovery along with the round-trip times for each hop.

5. Functionality:

- traceroute is more feature-rich and provides more options for advanced diagnostics, such as specifying different ports, packet sizes, and timeouts.
- tracepath is simpler and primarily focused on determining the path MTU, in addition to tracing the route.

- **Nslookup**

```
dwayne-nixon@dwayne-nixon-V1-04:~$ nslookup www.dbit.in
Server:         127.0.0.53
Address:        127.0.0.53#53

Non-authoritative answer:
Name:   www.dbit.in
Address: 43.205.151.144
```

- **Netstat**

```
dwayne-nixon@dwayne-nixon-V1-04:~$ netstat
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 dwayne-nixon-V1-0:50746 bom07s45-in-f10.1:https TIME_WAIT
tcp        0      0 dwayne-nixon-V1-0:34376 192.168.0.144:8009      ESTABLISHED
tcp        0      0 dwayne-nixon-V1-0:49650 sl-in-f188.1e100.n:5228 ESTABLISHED
tcp        0      0 dwayne-nixon-V1-0:41964 bom12s16-in-f3.1e:https ESTABLISHED
tcp        0      0 dwayne-nixon-V1-0:60670 192.168.0.102:8009      ESTABLISHED
udp        0      0 dwayne-nixon-V1-:bootpc _gateway:bootps         ESTABLISHED
udp        0      0 dwayne-nixon-V1-:bootpc _gateway:bootps         ESTABLISHED
Active UNIX domain sockets (w/o servers)
Proto RefCnt Flags       Type       State         I-Node   Path
unix  3      [ ]         STREAM     CONNECTED     39333
unix  3      [ ]         STREAM     CONNECTED     34582    /run/user/1000/bus
unix  2      [ ]         DGRAM      CONNECTED     37203
unix  3      [ ]         STREAM     CONNECTED     36215
unix  3      [ ]         STREAM     CONNECTED     38260    /run/user/1000/at-spi/bus
unix  3      [ ]         STREAM     CONNECTED     42149    /run/user/1000/bus
unix  3      [ ]         STREAM     CONNECTED     38211    /run/user/1000/bus
unix  3      [ ]         STREAM     CONNECTED     32237
unix  3      [ ]         STREAM     CONNECTED     35946
unix  3      [ ]         STREAM     CONNECTED     27382    /run/systemd/journal/stdout
unix  3      [ ]         STREAM     CONNECTED     32012
unix  3      [ ]         STREAM     CONNECTED     25474
unix  3      [ ]         STREAM     CONNECTED     38232    /run/cups/cups.sock
```

- **Arp**

```
dwayne-nixon@dwayne-nixon-V1-04:~$ arp
Address                  HWtype  HWaddress           Flags Mask            Iface
_gateway                 ether   a4:2b:b0:26:71:37   C                     enp4s0
192.168.0.128            ether   28:87:ba:3e:ce:f8   C                     enp4s0
192.168.0.144            ether   44:d8:78:44:27:b8   C                     enp4s0
192.168.0.102            ether   30:fd:38:03:f3:fc   C                     enp4s0
_gateway                 ether   a4:2b:b0:26:71:37   C                     wlp5s0
192.168.0.128            ether   28:87:ba:3e:ce:f8   C                     wlp5s0
192.168.0.144            ether   44:d8:78:44:27:b8   C                     wlp5s0
```

- **Ip addr show**

```
dwayne-nixon@dwayne-nixon-V1-04:~$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    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: enp4s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether e4:a8:df:fd:8c:85 brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.177/24 brd 192.168.0.255 scope global dynamic noprefixroute enp4s0
       valid_lft 6557sec preferred_lft 6557sec
    inet6 fe80::e6a8:dfff:fefd:8c85/64 scope link
       valid_lft forever preferred_lft forever
3: wlp5s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default qlen 1000
    link/ether c0:3c:59:fa:5a:7d brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.121/24 brd 192.168.0.255 scope global dynamic noprefixroute wlp5s0
       valid_lft 6556sec preferred_lft 6556sec
    inet6 fe80::7139:148c:3475:2ef/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

- **Dig**

```
dwayne-nixon@dwayne-nixon-V1-04:~$ dig www.dbit.in

; <<>> DiG 9.18.24-0ubuntu0.23.10.1-Ubuntu <<>> www.dbit.in
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 33414
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;www.dbit.in.                   IN      A

;; ANSWER SECTION:
www.dbit.in.            6816    IN      A       43.205.151.144

;; Query time: 0 msec
;; SERVER: 127.0.0.53#53(127.0.0.53) (UDP)
;; WHEN: Sat Jul 13 21:12:26 IST 2024
;; MSG SIZE  rcvd: 56
```

```
dwayne-nixon@dwayne-nixon-V1-04:~$ dig www.google.com

; <<>> DiG 9.18.24-0ubuntu0.23.10.1-Ubuntu <<>> www.google.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 40678
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;www.google.com.                        IN      A

;; ANSWER SECTION:
www.google.com.         24      IN      A       142.250.66.4

;; Query time: 0 msec
;; SERVER: 127.0.0.53#53(127.0.0.53) (UDP)
;; WHEN: Sat Jul 13 21:12:32 IST 2024
;; MSG SIZE  rcvd: 59
```

- **Route**

```
dwayne-nixon@dwayne-nixon-V1-04:~$ route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
default         _gateway        0.0.0.0         UG    100    0        0 enp4s0
default         _gateway        0.0.0.0         UG    600    0        0 wlp5s0
link-local      0.0.0.0         255.255.0.0     U     1000   0        0 wlp5s0
192.168.0.0     0.0.0.0         255.255.255.0   U     100    0        0 enp4s0
192.168.0.0     0.0.0.0         255.255.255.0   U     600    0        0 wlp5s0
```

```
dwayne-nixon@dwayne-nixon-V1-04:~$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         192.168.0.1     0.0.0.0         UG    100    0        0 enp4s0
0.0.0.0         192.168.0.1     0.0.0.0         UG    600    0        0 wlp5s0
169.254.0.0     0.0.0.0         255.255.0.0     U     1000   0        0 wlp5s0
192.168.0.0     0.0.0.0         255.255.255.0   U     100    0        0 enp4s0
192.168.0.0     0.0.0.0         255.255.255.0   U     600    0        0 wlp5s0
```

The route -n command is used to display the routing table of a Linux system without performing DNS lookups on the IP addresses. This provides a quicker and clearer view of the network routes.
Following is the list of the details seen on the command screen:

- **Destination:** The destination network or host.
- **Gateway:** The gateway through which packets should be forwarded.
- **Genmask:** The netmask for the destination network.
- **Flags:** Various flags that indicate route status:
    - **U:** Route is up.
    - **G:** Route is to a gateway.
    - **H:** Route is to a host.
    - **R:** Reinstate route for dynamic routing.
    - **D:** Dynamically installed by daemon or redirect.
    - **M:** Modified from routing daemon or redirect.

- **A:** Installed by AddrConf.
- **C:** Cache entry.
- **!:** Reject route.

- **Metric:** The cost associated with the route. Lower metrics are preferred.
- **Ref:** The number of references to this route.
- **Use:** The count of lookups for the route.
- **Iface:** The interface to which packets for this route will be sent.

- **Iwconfig**

```
dwayne-nixon@dwayne-nixon-V1-04:~$ iwconfig
lo        no wireless extensions.

enp4s0    no wireless extensions.

wlp5s0    IEEE 802.11  ESSID:"Nixon_5GHz"
          Mode:Managed  Frequency:5.2 GHz  Access Point: A4:2B:B0:26:71:36
          Bit Rate=866.7 Mb/s   Tx-Power=22 dBm
          Retry short limit:7   RTS thr:off   Fragment thr:off
          Power Management:on
          Link Quality=70/70  Signal level=-32 dBm
          Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
          Tx excessive retries:0  Invalid misc:255   Missed beacon:0
```

Iwconfig is a Linux command-line tool used to configure and manage wireless network interfaces. It displays and allows you to modify settings such as the network name (ESSID), mode (e.g., Managed, Ad-Hoc), frequency, access point, bit rate, and transmission power.
Following are the key fields within the given command:
- **Interface Name:** e.g., `wlan0`
- **ESSID:** Network name
- **Mode:** Operation mode (Managed, Ad-Hoc, etc.)
- **Frequency:** Channel in use
- **Access Point:** MAC address of the connected AP
- **Bit Rate:** Data transmission rate
- **Tx-Power:** Transmission power
- **Link Quality:** Connection quality
- **Signal Level:** Signal strength in dBm

- **Ss**

```
dwayne-nixon@dwayne-nixon-V1-04:~$ ss
Netid  State      Recv-Q  Send-Q                                  Local Address:Port                  Peer Address:Port                 Process
u_str  ESTAB      0       0                                                   * 39333                            * 39332
u_str  ESTAB      0       0                                  /run/user/1000/bus 34582                            * 25460
u_dgr  ESTAB      0       0                                                   * 37203                            * 17173
u_str  ESTAB      0       0                                                   * 36215                            * 25255
u_str  ESTAB      0       0                        /run/user/1000/at-spi/bus 38260                            * 32369
u_str  ESTAB      0       0                                  /run/user/1000/bus 42149                            * 37258
u_str  ESTAB      0       0                                  /run/user/1000/bus 38211                            * 40159
u_str  ESTAB      0       0                                                   * 32237                            * 33561
u_str  ESTAB      0       0                                                   * 35946                            * 37898
u_str  ESTAB      0       0                       /run/systemd/journal/stdout 27382                            * 25026
u_str  ESTAB      0       0                                                   * 32012                            * 27361
u_str  ESTAB      0       0                                                   * 25474                            * 39305
u_str  ESTAB      0       0                                                   * 40168                            * 40169
u_str  ESTAB      0       0                                                   * 38343                            * 38344
u_str  ESTAB      0       0                                                   * 38299                            * 37417
u_str  ESTAB      0       0                                                   * 31530                            * 31531
u_str  ESTAB      0       0                          @/tmp/.ICE-unix/2178 40154                            * 25437
u_str  ESTAB      0       0                                                   * 41157                            * 33671
u_str  ESTAB      0       0                                                   * 38206                            * 41116
u_dgr  ESTAB      0       0                                                   * 25263                            * 17173
```

The ss command in Linux displays detailed information about network connections and sockets. It is a modern alternative to netstat, offering faster and more comprehensive data.

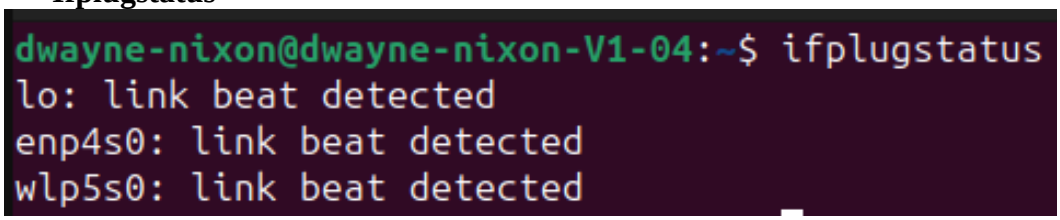
The Key Options of the Command Include:

- **ss -a:** Show all sockets.
- **ss -l:** List listening sockets.
- **ss -t:** Display TCP connections.
- **ss -u:** Display UDP connections.
- **ss -s:** Show socket statistics.

The Key Fields of the Command Include:

- **State:** Socket state (e.g., LISTEN).
- **Recv-Q/Send-Q:** Queue sizes.
- **Local Address**: Local IP and port.
- **Peer Address:** Remote IP and port.

- **Ifplugstatus**



The ifplugstatus command in Linux is used to display the link status of network interfaces, indicating whether the network cable is plugged in or not.
The Key Information Displayed Includes:

- **Interface Name:** The name of the network interface (e.g., eth0, wlan0).
- **Link Status:** Indicates whether the network cable is link beat detected (plugged in) or unplugged (not plugged in).

REFERENCES

- B.A. Forouzan, “Data Communications and Networking”, TMH,Fourth Edition.
- https://www.networkworld.com/article/2697039/unix- topnetworking-commands-and-what-they-tell-you.html
- https://www.youtube.com/watch?v=rurs7cdT5cc
- https://www.youtube.com/watch?v=V_3t2wBBB1U
- https://www.youtube.com/watch?v=75lCgcXP4dc