MAUREEN MIRANDA
2022300060

CCN LAB-1

Aim: Use Networking utilities.

1)Ping

1)Ping <destination>

The ping command in Linux is a utility that helps to test connectivity between 2 devices on a network .ping command sends a request to a specified device and waits for a response . Response from device helps us to determine whether device is available or not.ping command uses ICMP packets to communicate with target device .

```
students@students-Veriton-Series:~/maureen$ ping www.google.com
PING www.google.com (216.239.38.120) 56(84) bytes of data.
64 bytes from any-in-2678.1e100.net (216.239.38.120): icmp_seq=1 ttl=118 time=1.
56 ms
64 bytes from any-in-2678.1e100.net (216.239.38.120): icmp_seq=2 ttl=118 time=1.
49 ms
64 bytes from any-in-2678.1e100.net (216.239.38.120): icmp_seq=3 ttl=118 time=1.
55 ms
64 bytes from any-in-2678.1e100.net (216.239.38.120): icmp_seq=4 ttl=118 time=1.
57 ms
^C
--- www.google.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 1.488/1.542/1.572/0.032 ms
```

2) -D
Print timestamp (unix time + microseconds as in gettimeofday)
before each line.

```
students@students-Veriton-Series:~/maureen$ ping -D www.youtube.com
PING youtube-ui.l.google.com (172.217.167.174) 56(84) bytes of data.
[1705637712.668049] 64 bytes from bom12s01-in-f14.1e100.net (172.217.167.174):
cmp_seq=1 ttl=118 time=1.72 ms
[1705637713.402581] 64 bytes from bom12s01-in-f14.1e100.net (172.217.167.174):
cmp_seq=2 ttl=118 time=1.47 ms
[1705637714.404419] 64 bytes from bom12s01-in-f14.1e100.net (172.217.167.174):
cmp_seq=3 ttl=118 time=1.51 ms
[1705637715.405471] 64 bytes from bom12s01-in-f14.1e100.net (172.217.167.174):
cmp_seq=4 ttl=118 time=1.48 ms
[1705637716.407371] 64 bytes from bom12s01-in-f14.1e100.net (172.217.167.174):
cmp_seq=5 ttl=118 time=1.51 ms
^C
--- youtube-ui.l.google.com ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4006ms
```

3)-V
Show version and exit.

```
students@students-Veriton-Series:~/maureen$ ping -V www.google.com
ping from iputils s20190709
```

4) -n
Numeric output only. No attempt will be made to lookup symbolic names for host addresses.

```
students@students-Veriton-Series:~/maureen$ ping -n www.google.com
PING www.google.com (216.239.38.120) 56(84) bytes of data.
64 bytes from 216.239.38.120: icmp_seq=1 ttl=118 time=1.52 ms
64 bytes from 216.239.38.120: icmp_seq=2 ttl=118 time=1.58 ms
64 bytes from 216.239.38.120: icmp_seq=3 ttl=118 time=1.64 ms
64 bytes from 216.239.38.120: icmp_seq=4 ttl=118 time=1.39 ms
64 bytes from 216.239.38.120: icmp_seq=5 ttl=118 time=1.59 ms
64 bytes from 216.239.38.120: icmp_seq=6 ttl=118 time=1.56 ms
64 bytes from 216.239.38.120: icmp_seq=7 ttl=118 time=1.48 ms
^C
--- www.google.com ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6011ms
rtt min/avg/max/mdev = 1.394/1.536/1.638/0.074 ms
```

5) -q
Quiet output. Nothing is displayed except the summary lines at startup time and when finished.

```
students@students-Veriton-Series:~/maureen$ ping -q www.google.com
PING www.google.com (216.239.38.120) 56(84) bytes of data.
^C
--- www.google.com ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 1.640/1.670/1.701/0.030 ms
```

2) ***nslookup***

nslookup — The command nslookup <host> will do a DNS query to find and report the IP address (or
addresses) for a domain name or the domain name corresponding to an IP address. To do this, it contacts

a "DNS server." Default DNS servers are part of a computer's network configuration. (For a static IP address
in Linux, they are configured in the file /etc/network/interfaces that you encountered in the last lab.) You
can specify a different DNS server to be used by nslokup by adding the server name or IP address to the
command:

1) nslookup <host> <server>

```
students@students-Veriton-Series:~/maureen$ nslookup google.com
Server:         127.0.0.53
Address:        127.0.0.53#53

Non-authoritative answer:
Name:   google.com
Address: 216.239.38.120
Name:   google.com
Address: 2404:6800:4009:806::200e
```

2) nslookup ip address

```
students@students-Veriton-Series:~/maureen$ nslookup 127.0.0.53
53.0.0.127.in-addr.arpa name = localhost.

Authoritative answers can be found from:
```

3) any:

Lookup for any record We can also view all the available DNS records using the -type=any option.

```
students@students-Veriton-Series:~/maureen$ man nslookup
students@students-Veriton-Series:~/maureen$ nslookup -type=any google.com
Server:         127.0.0.53
Address:        127.0.0.53#53

Non-authoritative answer:
Name:   google.com
Address: 216.239.38.120
google.com      rdata_65 = \# 13 000100000100060268320268 33

Authoritative answers can be found from:
```

4) nslookup -type=soa redhat.com

Lookup for a soa record SOA record (start of authority), provides the authoritative information about the domain, the e-mail address of the domain admin, the domain serial number, etc…

```
students@students-Veriton-Series:~/maureen$ nslookup -type=soa google.com
Server:         127.0.0.53
Address:        127.0.0.53#53

Non-authoritative answer:
google.com
        origin = ns1.google.com
        mail addr = dns-admin.google.com
        serial = 599452336
        refresh = 900
        retry = 900
        expire = 1800
        minimum = 60

Authoritative answers can be found from:
```

5) nslookup -type=ns google.com

Lookup for an ns record. NS (Name Server) record maps a domain name to a list of DNS servers authoritative for that domain. It will output the name serves which are associated with the given domain.

```
students@students-Veriton-Series:~/maureen$ nslookup -type=ns google.com
Server:         127.0.0.53
Address:        127.0.0.53#53

Non-authoritative answer:
google.com      nameserver = ns2.google.com.
google.com      nameserver = ns3.google.com.
google.com      nameserver = ns1.google.com.
google.com      nameserver = ns4.google.com.

Authoritative answers can be found from:
```

6) nslookup -type=a google.com

Lookup for a record. We can also view all the available DNS records for a particular record using the -*type=a* option

```
students@students-Veriton-Series:~/maureen$ nslookup -type=a google.com
Server:         127.0.0.53
Address:        127.0.0.53#53

Non-authoritative answer:
Name:   google.com
Address: 216.239.38.120
```

7) nslookup -type=mx google.com

Lookup for an mx record.MX (Mail Exchange) maps a domain name to a list of mail exchange servers for that domain. The MX record says that all the mails sent to “google.com” should be routed to the Mail server in that domain.

```
students@students-Veriton-Series:~/maureen$ nslookup -type=mx youtub
e.com
Server:         127.0.0.53
Address:        127.0.0.53#53

Non-authoritative answer:
youtube.com     mail exchanger = 0 smtp.google.com.

Authoritative answers can be found from:
```

8) nslookup -type=txt google.com

Lookup for a txt record. TXT records are useful for multiple types of records like DKIM, SPF, etc. You can find all TXT records configured for any domain using the command below.

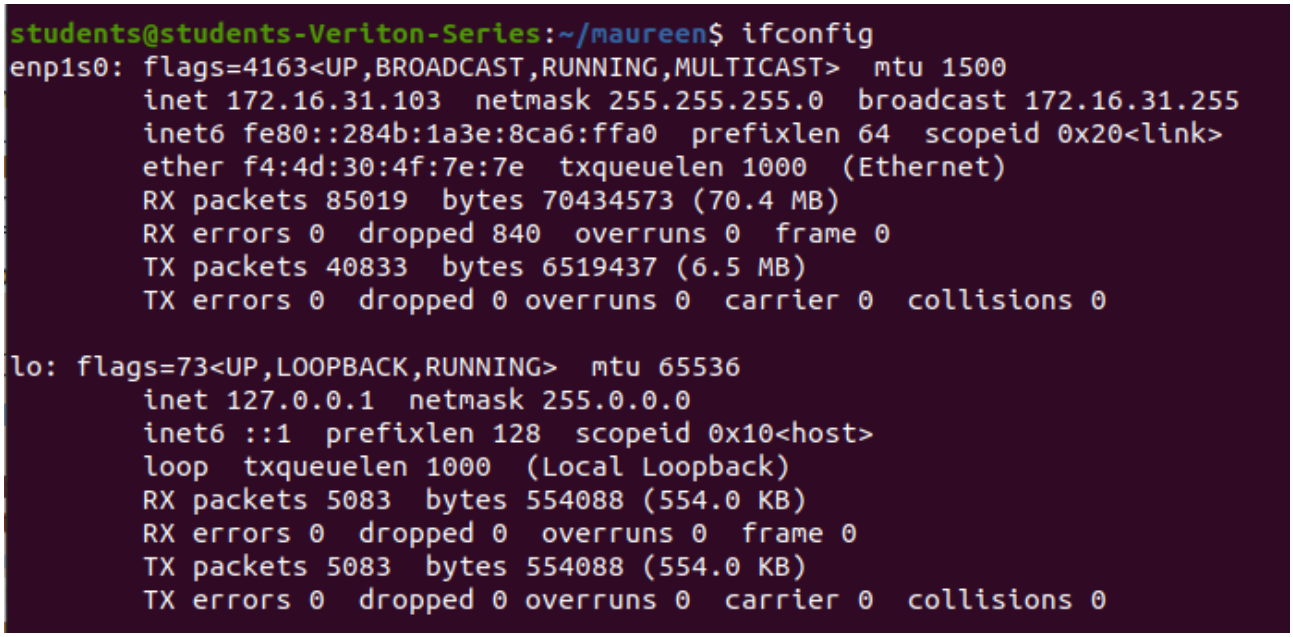
```
students@students-Veriton-Series:~/maureen$ nslookup -type=txt youtube.com
Server:         127.0.0.53
Address:        127.0.0.53#53

Non-authoritative answer:
youtube.com     text = "facebook-domain-verification=64jdes7le4h7e7lfpi22rijygx58j1"
youtube.com     text = "v=spf1 include:google.com mx -all"
youtube.com     text = "google-site-verification=QtQWEwHWM8tHiJ4s-jJWzEQrD_fF3luPnpzNDH-Nw-w"

Authoritative answers can be found from:
```

3) ***ifconfig***

ifconfig — You used ifconfig in the previous lab. When used with no parameters, ifconfig reports some

information about the computer's network interfaces. This usually includes lo which stands for localhost; it
can be used for communication between programs running on the same computer. Linux often has an
interface named eth0, which is the first ethernet card. The information is different on Mac OS and Linux,
but includes the IP or "inet" address and ethernet or "hardware" address for an ethernet card. On Linux,
you get the number of packets received (RX) and sent (TX), as well as the number of bytes transmitted and
received.

ifconfig
To view information about all network interfaces on your Linux system



1)

-a Display all interfaces, including those that are down ifconfig -a

```
students@students-Veriton-Series:~/maureen$ ifconfig -a
enp1s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.16.31.103  netmask 255.255.255.0  broadcast 172.16.31.255
        inet6 fe80::284b:1a3e:8ca6:ffa0  prefixlen 64  scopeid 0x20<link>
        ether f4:4d:30:4f:7e:7e  txqueuelen 1000  (Ethernet)
        RX packets 85788  bytes 70542614 (70.5 MB)
        RX errors 0  dropped 860  overruns 0  frame 0
        TX packets 41385  bytes 6584685 (6.5 MB)
        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 5101  bytes 556270 (556.2 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 5101  bytes 556270 (556.2 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

2)

-s Display a short list, instead of details ifconfig -s

```
students@students-Veriton-Series:~/maureen$ ifconfig -s
Iface      MTU    RX-OK RX-ERR RX-DRP RX-OVR    TX-OK TX-ERR TX-DRP TX-OVR Flg
enp1s0    1500    85948      0    866 0         41423      0      0      0 BMRU
lo       65536     5105      0      0 0          5105      0      0      0 LRU
```

3)

-v Run the command in verbose mode ifconfig -v

```
students@students-Veriton-Series:~/maureen$ ifconfig -v
enp1s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.16.31.103  netmask 255.255.255.0  broadcast 172.16.31.255
        inet6 fe80::284b:1a3e:8ca6:ffa0  prefixlen 64  scopeid 0x20<link>
        ether f4:4d:30:4f:7e:7e  txqueuelen 1000  (Ethernet)
        RX packets 86277  bytes 70609409 (70.6 MB)
        RX errors 0  dropped 875  overruns 0  frame 0
        TX packets 41483  bytes 6602969 (6.6 MB)
        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 5113  bytes 557662 (557.6 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 5113  bytes 557662 (557.6 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

3) Netstat

netstat — The netstat command gives information about network connections. I often use netstat -t -n
which lists currently open TCP connections (that's the "-t" option) by IP address rather than domain name
(that's the "-n" option). Add the option "-l" (lower case ell) to list listening sockets, that is sockets that have
been opened by server programs to wait for connection requests from clients: netstat -t -n -l. (On Mac,
use netstat -p tcp to list tcp connections, and add "-a" to include listening sockets in the list.


```
students@students-Veriton-Series:~/maureen$ ifconfig -v
enp1s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.16.31.103  netmask 255.255.255.0  broadcast 172.16.31.255
        inet6 fe80::284b:1a3e:8ca6:ffa0  prefixlen 64  scopeid 0x20<link>
        ether f4:4d:30:4f:7e:7e  txqueuelen 1000  (Ethernet)
        RX packets 86277  bytes 70609409 (70.6 MB)
        RX errors 0  dropped 875  overruns 0  frame 0
        TX packets 41483  bytes 6602969 (6.6 MB)
        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 5113  bytes 557662 (557.6 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 5113  bytes 557662 (557.6 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```


netstat commands

```
students@students-Veriton-Series:~/maureen$ netstat
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 students-Veriton-:45870 ns1.spit.ac.in:domain   TIME_WAIT
tcp        0      0 students-Veriton-:50730 69.173.151.100:https    TIME_WAIT
tcp        0      0 students-Veriton-:56138 39.12.213.35.bc.g:https ESTABLISHED
tcp        0      0 students-Veriton-:43576 8.18.47.7:https         ESTABLISHED
tcp        0      0 students-Veriton-:51460 a23-212-254-32.de:https ESTABLISHED
tcp        0      0 students-Veriton-:54056 pnbomb-aa-in-f10.:https ESTABLISHED
tcp        0      0 students-Veriton-:48168 cloudproxy10022.su:http ESTABLISHED
tcp        0      0 students-Veriton-:54454 ec2-13-228-126-19:https ESTABLISHED
tcp        0      0 students-Veriton-:60578 8.159.244.35.bc.g:https ESTABLISHED
tcp        0      0 students-Veriton-:52952 93.243.107.34.bc.:https ESTABLISHED
tcp        0      0 students-Veriton-:44352 152.195.38.76:http      ESTABLISHED
tcp        0      0 students-Veriton-:39166 bom12s16-in-f10.1:https TIME_WAIT
tcp        0      0 students-Veriton-:51060 server-18-67-195-:https ESTABLISHED
tcp        0      1 students-Veriton-:54538 ip-185-184-8-90.r:https SYN_SENT
tcp        0      0 students-Veriton-:33220 bom07s36-in-f2.1e:https ESTABLISHED
tcp        0      0 students-Veriton-:45012 ec2-52-72-177-11.:https ESTABLISHED
tcp        0      0 students-Veriton-:57328 a23-212-254-65.de:https ESTABLISHED
```

1)

-a -all : Show both listening and non-listening sockets. With the –interfaces option, show interfaces that are not up.

netstat -a

```
students@students-Veriton-Series:~/maureen$ netstat -a
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 localhost:ipp           0.0.0.0:*               LISTEN
tcp        0      0 localhost:33060         0.0.0.0:*               LISTEN
tcp        0      0 localhost:domain        0.0.0.0:*               LISTEN
tcp        0      0 localhost:mysql         0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:daap            0.0.0.0:*               LISTEN
tcp        0      0 students-Veriton-:34900 bom12s12-in-f1.1e:https TIME_WAIT
tcp        0      0 students-Veriton-:32894 hkg12s09-in-f10.1:https ESTABLISHED
tcp        0      0 students-Veriton-:56768 39.12.213.35.bc.g:https ESTABLISHED
tcp        0      0 students-Veriton-:51564 sh-in-f84.1e100.n:https ESTABLISHED
tcp        0      0 students-Veriton-:40388 151.101.153.229:https   ESTABLISHED
tcp        0      0 students-Veriton-:48158 bom12s16-in-f10.1:https TIME_WAIT
tcp        0      0 students-Veriton-:47152 218.64.98.34.bc.g:https ESTABLISHED
tcp        0      0 students-Veriton-:34388 any-in-2678.1e100:https TIME_WAIT
tcp        0      1 students-Veriton-:42672 228.180.214.35.bc:https SYN_SENT
tcp        0      0 students-Veriton-:54630 57.128.114.222:https    ESTABLISHED
^C
```

This command specifically lists all TCP ports, giving you information about the TCP connections your system is engaged in.

2)

netstat -at

```
students@students-Veriton-Series:~/maureen$ netstat -at
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 localhost:ipp           0.0.0.0:*               LISTEN
tcp        0      0 localhost:33060         0.0.0.0:*               LISTEN
tcp        0      0 localhost:domain        0.0.0.0:*               LISTEN
tcp        0      0 localhost:mysql         0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:daap            0.0.0.0:*               LISTEN
tcp        0      0 students-Veriton-:32894 hkg12s09-in-f10.1:https ESTABLISHED
tcp        0      0 students-Veriton-:51564 sh-in-f84.1e100.n:https ESTABLISHED
tcp        0      0 students-Veriton-:40388 151.101.153.229:https   ESTABLISHED
tcp        0      0 students-Veriton-:47152 218.64.98.34.bc.g:https ESTABLISHED
tcp        0      1 students-Veriton-:42672 228.180.214.35.bc:https SYN_SENT
tcp        0      0 students-Veriton-:54630 57.128.114.222:https    ESTABLISHED
^C
```

3)

this command focuses on UDP ports, revealing details about UDP connections.

netstat -au

```
students@students-Veriton-Series:~/maureen$ netstat -au
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
udp        0      0 localhost:domain        0.0.0.0:*
udp        0      0 students-Veriton:bootpc 172.16.10.37:bootps     ESTABLISHED
udp        0      0 0.0.0.0:631             0.0.0.0:*
udp        0      0 0.0.0.0:50392           0.0.0.0:*
udp        0      0 0.0.0.0:mdns            0.0.0.0:*
udp6       0      0 [::]:55827              [::]:*
udp6       0      0 [::]:mdns               [::]:*
```

4)

By using this option, you can see only the ports that are actively listening for incoming connections

netstat -l

```
udp6       0      0 [::]:mdns               [::]:*
students@students-Veriton-Series:~/maureen$ netstat -l
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 localhost:ipp           0.0.0.0:*               LISTEN
tcp        0      0 localhost:33060         0.0.0.0:*               LISTEN
tcp        0      0 localhost:domain        0.0.0.0:*               LISTEN
tcp        0      0 localhost:mysql         0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:daap            0.0.0.0:*               LISTEN
tcp6       0      0 ip6-localhost:ipp       [::]:*                  LISTEN
tcp6       0      0 [::]:daap               [::]:*                  LISTEN
udp        0      0 localhost:domain        0.0.0.0:*
udp        0      0 0.0.0.0:631             0.0.0.0:*
udp        0      0 0.0.0.0:50392           0.0.0.0:*
udp        0      0 0.0.0.0:mdns            0.0.0.0:*
udp6       0      0 [::]:55827              [::]:*
udp6       0      0 [::]:mdns               [::]:*
raw6       0      0 [::]:ipv6-icmp          [::]:*                  7
Active UNIX domain sockets (only servers)
Proto RefCnt Flags       Type       State         I-Node   Path
unix  2      [ ACC ]     STREAM     LISTENING     42080    @/home/students/.cache/ibus/dbus-JSffh
unix  2      [ ACC ]     STREAM     LISTENING     39599    @/tmp/.ICE-unix/1610
unix  2      [ ACC ]     STREAM     LISTENING     37356    /run/user/1000/systemd/private
```

5)

Narrowing it down further, this command specifically lists the TCP ports that are in a listening state.

netstat -lt

```
students@students-Veriton-Series:~/maureen$ netstat -lt
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 localhost:ipp           0.0.0.0:*               LISTEN
tcp        0      0 localhost:33060         0.0.0.0:*               LISTEN
tcp        0      0 localhost:domain        0.0.0.0:*               LISTEN
tcp        0      0 localhost:mysql         0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:daap            0.0.0.0:*               LISTEN
tcp6       0      0 ip6-localhost:ipp       [::]:*                  LISTEN
tcp6       0      0 [::]:daap               [::]:*                  LISTEN
```

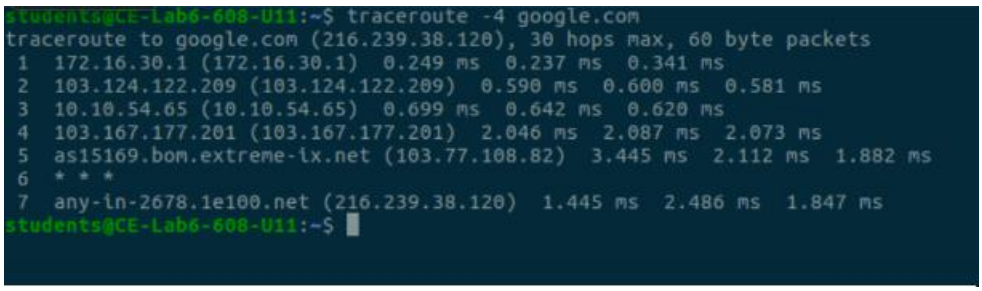
4) traceroute

traceroute command in Linux prints the route that a packet takes to reach the host. This command is useful when you want to know about the route and about all the hops that a packet takes. Below image depicts how traceroute command is used to reach the Google(172.217.26.206) host from the local machine and it also prints detail about all the hops that it visits in between.

```
students@students-Veriton-Series:~/maureen$ traceroute google.com
traceroute to google.com (216.239.38.120), 64 hops max
  1   172.16.31.1  0.297ms  *  0.327ms
  2   103.104.226.57  0.831ms  0.555ms  0.523ms
  3   43.252.192.229  0.875ms  0.797ms  0.656ms
  4   45.64.194.81  0.976ms  0.836ms  0.826ms
  5   27.109.1.149  0.806ms  0.756ms  0.747ms
  6   72.14.204.217  1.338ms  1.531ms  1.231ms
^C
```

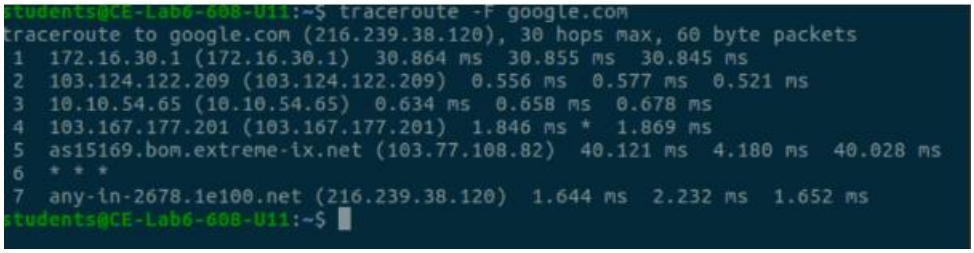
1) traceroute [options] host_Address [pathlength]

2) Use ip version 4 i.e. use IPv4



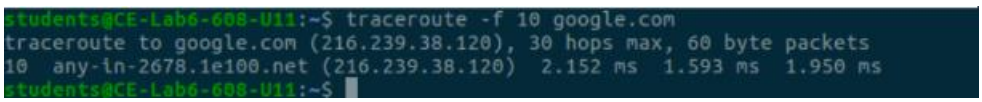
3) -F Option

Do not fragment packets



4) -f first_ttl Option

Start from the first_ttl hop (instead from 1).



Commands in Windows:

Ping

```
C:\Users\dcmau>ping google.com

Pinging google.com [142.250.192.14] with 32 bytes of data:
Reply from 142.250.192.14: bytes=32 time=29ms TTL=114
Reply from 142.250.192.14: bytes=32 time=74ms TTL=114
Reply from 142.250.192.14: bytes=32 time=27ms TTL=114
Reply from 142.250.192.14: bytes=32 time=122ms TTL=114

Ping statistics for 142.250.192.14:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 27ms, Maximum = 122ms, Average = 63ms
```

The -n option tells the ping command to send 3 ICMP Echo Requests instead of default of 4

```
C:\Users\dcmau>ping -n -3 google.com

Pinging google.com [142.250.192.14] with 32 bytes of data:
Reply from 142.250.192.14: bytes=32 time=12ms TTL=114
Reply from 142.250.192.14: bytes=32 time=11ms TTL=114
Reply from 142.250.192.14: bytes=32 time=55ms TTL=114

Ping statistics for 142.250.192.14:
    Packets: Sent = 3, Received = 3, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 11ms, Maximum = 55ms, Average = 26ms
```

Setting the size of the packet as 200

```
C:\Users\dcmau>ping -l 200 google.com

Pinging google.com [142.250.192.14] with 200 bytes of data:
Reply from 142.250.192.14: bytes=68 (sent 200) time=44ms TTL=114
Reply from 142.250.192.14: bytes=68 (sent 200) time=28ms TTL=114
Reply from 142.250.192.14: bytes=68 (sent 200) time=53ms TTL=114

Ping statistics for 142.250.192.14:
    Packets: Sent = 3, Received = 3, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 28ms, Maximum = 53ms, Average = 41ms
```

Timeout

```
C:\Users\dcmau>ping -w 2 google.com

Pinging google.com [142.250.192.14] with 32 bytes of data:
Request timed out.
Reply from 142.250.192.14: bytes=32 time=35ms TTL=114
Reply from 142.250.192.14: bytes=32 time=29ms TTL=114
Reply from 142.250.192.14: bytes=32 time=16ms TTL=114

Ping statistics for 142.250.192.14:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
    Minimum = 16ms, Maximum = 35ms, Average = 26ms
```

Hop count

```
C:\Users\dcmau>ping -s 3 google.com

Pinging google.com [142.250.192.14] with 32 bytes of data:
Request timed out.
Request timed out.

Ping statistics for 142.250.192.14:
    Packets: Sent = 2, Received = 0, Lost = 2 (100% loss),
Control-C
```

nslookup

```
C:\Users\dcmau>nslookup google.com
Server:  UnKnown
Address:  202.88.131.89

Non-authoritative answer:
Name:    google.com
Addresses:  2404:6800:4009:827::200e
          142.250.192.14
```

ifconfig(ipconfig in windows)

```
C:\Users\dcmau>ipconfig

Windows IP Configuration


Wireless LAN adapter Local Area Connection* 1:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Wireless LAN adapter Local Area Connection* 2:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Wireless LAN adapter Wi-Fi:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::f141:8916:8144:a636%18
   IPv4 Address. . . . . . . . . . . : 192.168.0.31
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.0.1

Ethernet adapter Bluetooth Network Connection:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :
```

Traceroute(tracert in windows)

Specifies not to resolve address to host names

```
C:\Users\dcmau>tracert -d google.com

Tracing route to google.com [142.250.192.46]
over a maximum of 30 hops:

  1     1 ms     1 ms     1 ms  192.168.0.1
  2     9 ms     8 ms     9 ms  10.210.0.1
  3    19 ms    11 ms    11 ms  192.168.3.62
  4     9 ms    10 ms    11 ms  192.168.44.93
  5     *       19 ms    10 ms  192.168.27.142
  6     9 ms    42 ms     9 ms  192.168.221.34
  7    10 ms    10 ms    11 ms  125.99.55.253
  8    10 ms    11 ms     *     125.99.55.163
  9    13 ms    12 ms    12 ms  125.99.55.165
 10    11 ms    13 ms    13 ms  142.251.225.77
 11    11 ms    11 ms    11 ms  142.250.212.171
 12    11 ms    13 ms    10 ms  142.250.192.46

Trace complete.
```

Number of traces after is satisfied

```
C:\Users\dcmau>tracert -h 7 google.com

Tracing route to google.com [142.250.192.46]
over a maximum of 7 hops:

  1     2 ms     1 ms     1 ms  192.168.0.1
  2     9 ms     8 ms    11 ms  10.210.0.1
  3    40 ms    11 ms    13 ms  192.168.3.62
  4    11 ms     8 ms     9 ms  192.168.44.93
  5     9 ms    10 ms     9 ms  192.168.27.142
  6    16 ms    10 ms    10 ms  192.168.221.34
  7    15 ms    18 ms    15 ms  125.99.55.253

Trace complete.
```

Netstat

```
C:\Users\dcmau>netstat -i

Active Connections

  Proto  Local Address          Foreign Address        State           Time in State (ms)
TCP    192.168.0.31:49329     104.26.8.101:https     ESTABLISHED        2367737
TCP    192.168.0.31:49336     sa-in-f188:5228        ESTABLISHED        2367421
TCP    192.168.0.31:49372     20.198.119.143:https   ESTABLISHED        2310394
TCP    192.168.0.31:49497     20.198.119.143:https   ESTABLISHED        2366889
TCP    192.168.0.31:50138     20.62.48.180:https     CLOSE_WAIT         1843694
TCP    192.168.0.31:50393     a23-54-83-201:https    CLOSE_WAIT         886316
TCP    192.168.0.31:50394     a23-54-83-201:https    CLOSE_WAIT         785035
TCP    192.168.0.31:50395     a23-54-83-201:https    CLOSE_WAIT         886316
TCP    192.168.0.31:50396     a23-54-83-201:https    CLOSE_WAIT         886256
TCP    192.168.0.31:50397     40.99.111.34:https     ESTABLISHED        906599
TCP    192.168.0.31:50398     a23-54-82-201:https    CLOSE_WAIT         770436
TCP    192.168.0.31:50399     a23-54-82-201:https    CLOSE_WAIT         886169
^C
```

Exercise 1: Experiment with ping to find the round trip times to a variety of destinations. Write up any interesting observations, including in particular how the round trip time compares to the physical distance. Here are few places from who to get replies: www.uw.edu, www.cornell.edu, berkeley.edu, www.uchicago.edu, www.ox.ac.uk (England), www.u-tokyo.ac.jp (Japan).

www.uw.edu distance: 12848.08 (km)

```
students@students-Veriton-M200-H110:~$ ping www.uw.edu
PING www-uw.smslb.s.uw.edu (34.168.51.100) 56(84) bytes of data.
^C
--- www-uw.smslb.s.uw.edu ping statistics ---
8 packets transmitted, 0 received, 100% packet loss, time 7170ms
```

www.cornell.edu distance: 12537.89 (km)

```
students@students-Veriton-M200-H110:~$ ping www.cornell.edu
PING part-0040.t-0009.t-msedge.net (13.107.246.68) 56(84) bytes of data.
From 104.44.212.200 (104.44.212.200) icmp_seq=1 Time to live exceeded
From 104.44.212.200 (104.44.212.200) icmp_seq=2 Time to live exceeded
From 104.44.212.200 (104.44.212.200) icmp_seq=3 Time to live exceeded
From 104.44.212.200 (104.44.212.200) icmp_seq=4 Time to live exceeded
^C
--- part-0040.t-0009.t-msedge.net ping statistics ---
4 packets transmitted, 0 received, +4 errors, 100% packet loss, time 3004ms
```

berkeley.edu distance: 13692.94 (km)

```
students@students-Veriton-M200-H110:~$ ping berkeley.edu
PING berkeley.edu (141.193.213.21) 56(84) bytes of data.
64 bytes from 141.193.213.21 (141.193.213.21): icmp_seq=1 ttl=59 time=1.24 ms
64 bytes from 141.193.213.21 (141.193.213.21): icmp_seq=2 ttl=59 time=1.33 ms
64 bytes from 141.193.213.21 (141.193.213.21): icmp_seq=3 ttl=59 time=1.21 ms
64 bytes from 141.193.213.21 (141.193.213.21): icmp_seq=4 ttl=59 time=1.32 ms
^C
--- berkeley.edu ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3006ms
rtt min/avg/max/mdev = 1.213/1.276/1.327/0.050 ms
```

www.uchicago.edu distance: 12946.73 (km)

```
students@students-Veriton-M200-H110:~$ ping uchicago.edu
PING uchicago.edu (3.215.148.80) 56(84) bytes of data.
^C
--- uchicago.edu ping statistics ---
4 packets transmitted, 0 received, 100% packet loss, time 3066ms
```

www.ox.ac.uk (England) distance: 7270.04 (km)

```
students@students-Veriton-M200-H110:~$ ping www.ox.ac.uk
PING www.ox.ac.uk (151.101.2.216) 56(84) bytes of data.
64 bytes from 151.101.2.216 (151.101.2.216): icmp_seq=1 ttl=53 time=24.8 ms
64 bytes from 151.101.2.216 (151.101.2.216): icmp_seq=2 ttl=53 time=24.6 ms
64 bytes from 151.101.2.216 (151.101.2.216): icmp_seq=3 ttl=53 time=24.7 ms
^C
--- www.ox.ac.uk ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2002ms
rtt min/avg/max/mdev = 24.649/24.718/24.782/0.054 ms
```

www.u-tokyo.ac.jp (Japan) distance: 6729.24 (km)

```
students@students-Veriton-M200-H110:~$ ping www.u-tokyo.ac.jp
PING www.u-tokyo.ac.jp (210.152.243.234) 56(84) bytes of data.
^C
--- www.u-tokyo.ac.jp ping statistics ---
3 packets transmitted, 0 received, 100% packet loss, time 2044ms
```

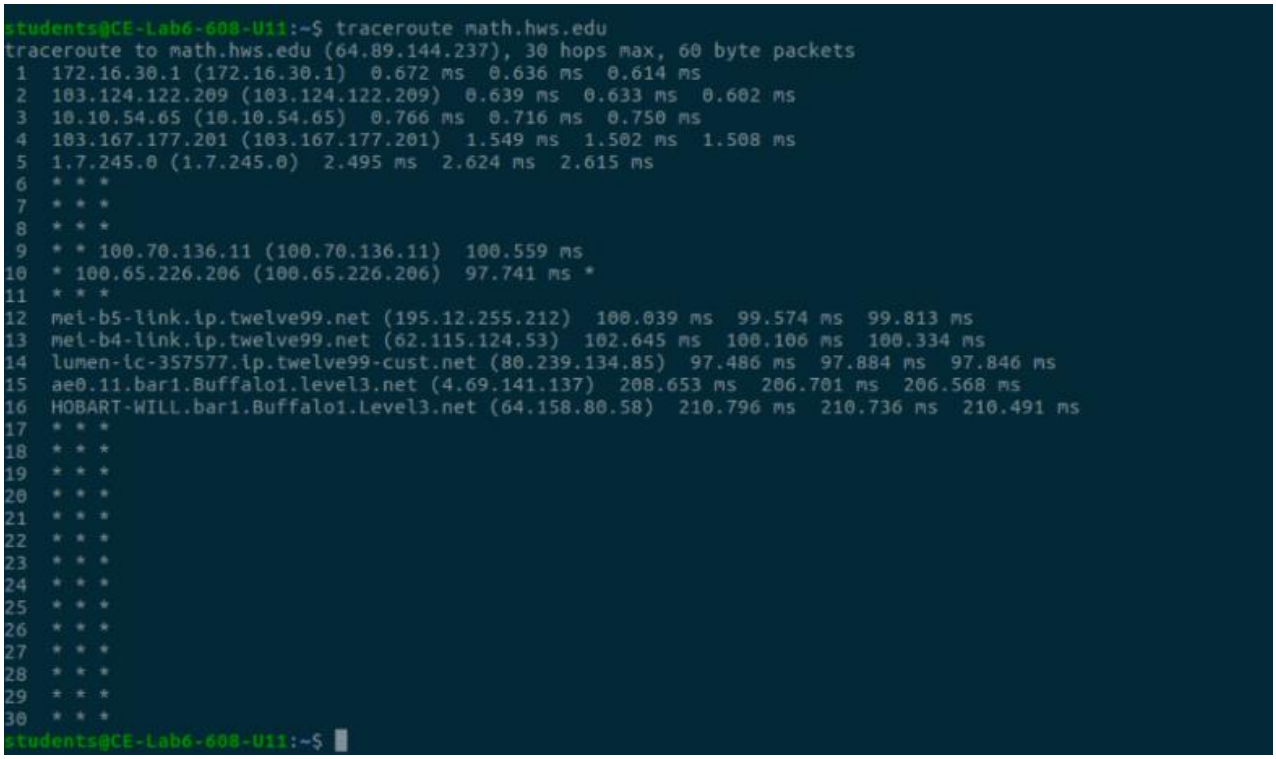
Observations:

On sending information over the internet the time it takes will be dependent on the distance it is required to travel. If it's a long way, it takes more time. Also, when the information passes through different stops on the way, like nodes in the network, it takes extra time at each stop.

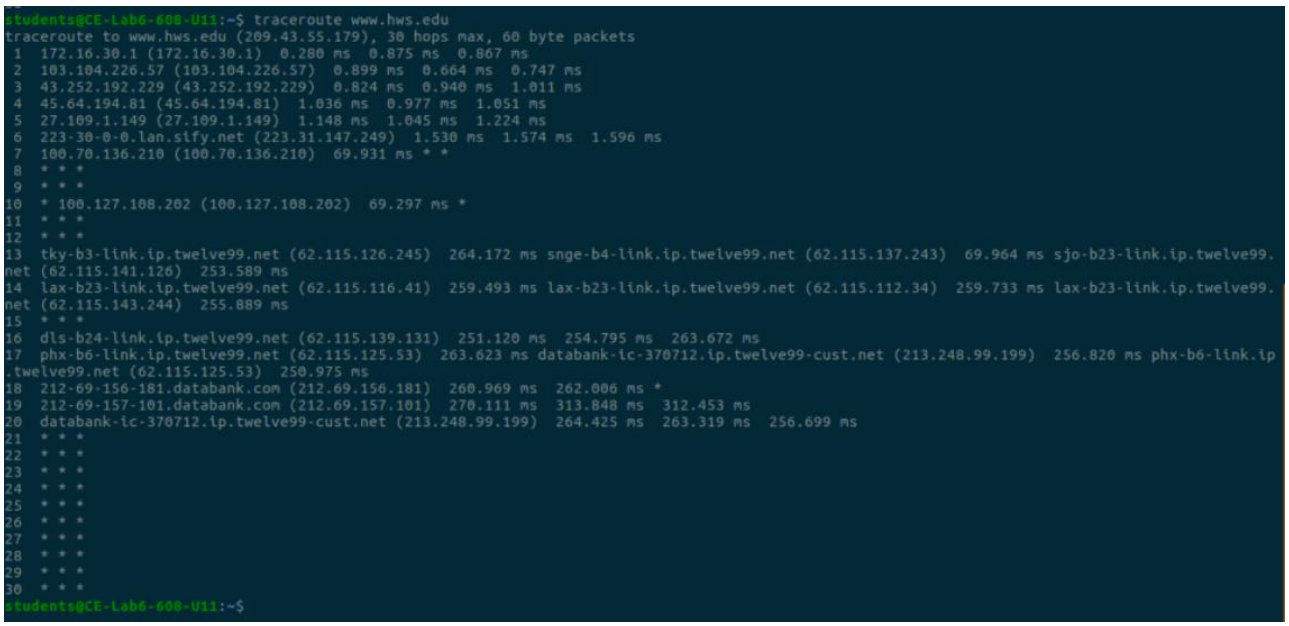
This travel time, known as round trip time (RTT), can increase when the network is busy or when we send more pieces of information. In our experiments, we noticed some issues. When trying to reach certain university websites, some pieces of information got lost during the trip. This could be because of network problems, like restrictions or the server not being available.
the time it takes for information to travel back and forth can vary a lot. This suggests that the information might be taking different routes each time it goes out and comes back

Exercise 2: (Very short.) Use traceroute to trace the route from your lab computer to math.hws.edu and to www.hws.edu. Explain the difference in the results.

traceroute math.hws.edu



traceroute www.hws.edu



Observations:
We tried to find the path that our information takes to reach a destination. Each stop it makes along the way is like a router or hop. What we noticed is that the path can be different each time, depending on the network. Some routers responded, but some didn't. The '*' you see means that the router didn't respond.In the end, we couldn't reach the final destination or end user.

Conclusion:
In conclusion, the exploration of networking commands and their applications provides valuable insights into the network connectivity, diagnostics, and information retrieval.
The ping command, a fundamental tool, allows users to test connectivity by sending ICMP packets and analyzing responses. The nslookup command is a tool for DNS-related inquiries, offering the ability to query domain names and IP addresses, along with retrieving various types of DNS records. ifconfig includes details such as IP addresses, hardware addresses, and statistics related to data transmission. The netstat command helped in monitoring network connections and identifying listening sockets and is used to test the system's network activity. The traceroute command played a crucial role in visualizing the route that packets take to reach a destination, showcasing the network hops and potential variations in paths.
In the first experiment, I used ping to see how fast data travels between computers. I found out that the time it takes depends on how far apart the computers are and how busy the network is.
The second experiment used traceroute to draw a map of how data travels from one place to another on the internet. I discovered that the path data takes can be different for various destinations.
Overall, the experiment enhanced my understanding of essential command-line networking tools on Linux and Windows.