CSE 232: Programming Assignment 1
Using command-line utilities for network debugging

Himanshu Raj, 2022216

Q.1. a) Learn to use the ifconfig command, and figure out the IP address of your network interface. Put a screenshot.

```
rahi@rahi:~$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.28.35.42  netmask 255.255.240.0  broadcast 172.28.47.255
        inet6 fe80::215:5dff:fe2b:fb0d  prefixlen 64  scopeid 0x20<link>
        ether 00:15:5d:2b:fb:0d  txqueuelen 1000  (Ethernet)
        RX packets 875  bytes 103119 (103.1 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 25  bytes 1750 (1.7 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 8  bytes 1182 (1.1 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 8  bytes 1182 (1.1 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

IP address of my network: 172.28.35.42

b) Go to the webpage https://www.whatismyip.com and find out what IP is shown for your machine. Are they identical or different? Why?

IP address on the website: 103.25.231.125
They are evidently different because the IP shown by ifconfig is a private IP address assigned to my device by the router for communication in LAN, while IP address shown by the website is the public IP assigned to my device.

Q.2. a) Change the IP address of your network interface using the command line. Put a screenshot that shows the change. Revert to the original IP address.

Changing IP:

```
rahi@rahi:~$ sudo ifconfig eth0 172.28.28.28
rahi@rahi:~$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.28.28.28  netmask 255.255.0.0  broadcast 172.28.255.255
        inet6 fe80::215:5dff:fe2b:fb0d  prefixlen 64  scopeid 0x20<link>
        ether 00:15:5d:2b:fb:0d  txqueuelen 1000  (Ethernet)
        RX packets 1516  bytes 159129 (159.1 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 27  bytes 1890 (1.8 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 8  bytes 1182 (1.1 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 8  bytes 1182 (1.1 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Reverting back:

```
rahi@rahi:~$ sudo ifconfig eth0 172.28.35.42
rahi@rahi:~$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.28.35.42  netmask 255.255.0.0  broadcast 172.28.255.255
        inet6 fe80::215:5dff:fe2b:fb0d  prefixlen 64  scopeid 0x20<link>
        ether 00:15:5d:2b:fb:0d  txqueuelen 1000  (Ethernet)
        RX packets 1520  bytes 159509 (159.5 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 27  bytes 1890 (1.8 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 8  bytes 1182 (1.1 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 8  bytes 1182 (1.1 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Q.3. a) Use “netcat” to set up a TCP client/server connection between your VM and host machine. If you are not using a VM, you can set up the connection with localhost. Put a screenshot.

```
rahi@rahi:~$ nc -l -p 1234
hi, this is the client
hi, this is the server
```

```
rahi@rahi:~$ nc localhost 1234
hi, this is the client
hi, this is the server
```

The first terminal is the server, and the second one is the client.

b) Determine the state of this TCP connection(s) at the client node. Put a screenshot.

```
rahi@rahi:~$ nc -l -p 1234
hi, this is the client
hi, this is the server
```

```
rahi@rahi:~$ netstat -an | grep 1234
tcp        0      0 0.0.0.0:1234            0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:1234          127.0.0.1:42454         ESTABLISHED
tcp        0      0 127.0.0.1:42454         127.0.0.1:1234          ESTABLISHED
rahi@rahi:~$ netstat
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 localhost:1234          localhost:42454         ESTABLISHED
tcp        0      0 localhost:42454         localhost:1234          ESTABLISHED
Active UNIX domain sockets (w/o servers)
```

It says the connection is established.

Q.4. a) Get an authoritative result for “google.in” using nslookup. Put a screenshot. Explain how you did it.

```
rahi@rahi:~$ nslookup -q=soa google.in
Server:         10.255.255.254
Address:        10.255.255.254#53

Non-authoritative answer:
google.in
        origin = ns1.google.com
        mail addr = dns-admin.google.com
        serial = 668858537
        refresh = 900
        retry = 900
        expire = 1800
        minimum = 60

Authoritative answers can be found from:
ns1.google.com  internet address = 216.239.32.10
ns1.google.com  has AAAA address 2001:4860:4802:32::a
```

I used the `-q=soa` flag to get the 'Start of Authority' Record for google.in, which returns the authoritative servers for google.in, and then I directly requested this server in nslookup.

```
rahi@rahi:~$ nslookup google.in ns1.google.com
Server:         ns1.google.com
Address:        216.239.32.10#53

Name:   google.in
Address: 142.250.182.164
Name:   google.in
Address: 2404:6800:4002:815::2004
```

b) Find out the time to live for any website on the local DNS. Put a screenshot. Explain in words (with unit) after how much time this entry would expire from the local DNS server.

TTL for google.in on local DNS is 293 seconds. This entry would expire from the local DNS server in 4 mins 53 secs (293 secs).

```
rahi@rahi:~$ dig google.in

; <<>> DiG 9.18.1-1ubuntu1.3-Ubuntu <<>> google.in
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 59868
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4000
;; QUESTION SECTION:
;google.in.                     IN      A

;; ANSWER SECTION:
google.in.              293     IN      A       142.250.193.4

;; Query time: 0 msec
;; SERVER: 10.255.255.254#53(10.255.255.254) (UDP)
;; WHEN: Fri Aug 30 19:29:57 IST 2024
;; MSG SIZE  rcvd: 54
```

Q.6. Make your ping command fail for 127.0.0.1 (with 100% packet loss). Explain how you do it. Put a screenshot that it failed.

I deactivated the lo (loopback) interface, which the system uses for internal communication. The IP 127.0.0.1 is the IP for localhost or to connect with another program on the same machine. So, the ping command failed as the internal communication interface was disabled.

```
rahi@rahi:~$ sudo ifconfig lo down
rahi@rahi:~$ ping 127.0.0.1
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
^C
--- 127.0.0.1 ping statistics ---
3 packets transmitted, 0 received, 100% packet loss, time 2090ms

rahi@rahi:~$ sudo ifconfig lo up
rahi@rahi:~$
```

Q.5. a) Run the command, traceroute google.in. How many intermediate hosts do you see? What are the IP addresses? Compute the average latency to each intermediate host. Put a screenshot.

Intermediate hosts: 8 (excluding the 5th host), 9 (including)
Their IP addresses: 172.28.32.1, 192.168.32.254, 192.168.1.99, 103.25.231.1, 10.119.234.162, 72.14.194.160, 192.178.80.159, 142.251.54.89
Average latencies (in order of IP addresses): 0.378ms, 28.561ms, 2.868ms, 3.861ms, 6.186ms, 10.277ms, 38.129ms, 30.500ms

```
rahi@rahi:~$ traceroute google.in
traceroute to google.in (142.250.193.4), 64 hops max
  1   172.28.32.1  0.560ms  0.322ms  0.252ms
  2   192.168.32.254  22.558ms  34.402ms  27.723ms
  3   192.168.1.99  3.764ms  2.482ms  2.359ms
  4   103.25.231.1  4.789ms  3.621ms  3.173ms
  5   *  *  *
  6   10.119.234.162  7.089ms  7.146ms  5.325ms
  7   72.14.194.160  6.633ms  6.930ms  18.269ms
  8   192.178.80.159  30.934ms  30.199ms  54.256ms
  9   142.251.54.89  27.368ms  26.398ms  38.735ms
 10   142.250.193.4  29.852ms  27.675ms  43.399ms
```

b) Send 50 ping messages to google.in, Determine the average latency. Put a screenshot.

Average Latency: 34.292ms

```
--- google.in ping statistics ---
50 packets transmitted, 50 received, 0% packet loss, time 49080ms
rtt min/avg/max/mdev = 28.955/34.292/71.864/8.509 ms
rahi@rahi:~$
```

c) Add up the ping latency of all the intermediate hosts obtained in (a) and compare with (b). Are they matching, explain?

No, they are not matching (120.760ms and 34.292ms). ping shows the overall round-trip time to the final destination. traceroute measures time for each hop and back, the total latency of intermediate hosts is the combined time to reach all intermediate hosts and return and each hop's latency is added up cumulatively, which is different than what ping does.

d) Take the maximum ping latency amongst the intermediate hosts (in (a)) and compare it with (b). Are they matching, explain?

No, they are not matching (38.129ms and 34.292ms). The maximum ping latency in traceroute is the latency when a packet is travelling from an intermediate source (can be original source) to an intermediate destination (can be original destination), while ping is round-trip time to the final destination without considering the round trip time of intermediate hops.

e) You may see multiple entries for a single hop while using the traceroute command. What do these entries mean?

These entries represent multiple attempts to measure the round-trip time to that particular hop.

f) Send 50 ping messages to stanford.edu, Determine the average latency. Put a screenshot.

Average Latency: 294.549ms

```
--- stanford.edu ping statistics ---
50 packets transmitted, 50 received, 0% packet loss, time 49070ms
rtt min/avg/max/mdev = 289.116/294.549/327.482/9.578 ms
rahi@rahi:~$
```

g) Run the command, traceroute stanford.edu. Compare the number of hops between google.in and stanford.edu (between the traceroute result of google.in and stanford.edu).

```
rahi@rahi:~$ traceroute stanford.edu
traceroute to stanford.edu (171.67.215.200), 64 hops max
  1   172.28.32.1  0.369ms  0.182ms  0.163ms
  2   192.168.32.254  53.843ms  15.932ms  8.303ms
  3   192.168.1.99  3.217ms  2.790ms  3.351ms
  4   103.25.231.1  5.413ms  3.136ms  3.165ms
  5   10.1.209.201  31.396ms  30.325ms  30.384ms
  6   10.1.200.137  45.972ms  37.396ms  34.087ms
  7   10.255.238.254  32.933ms  29.459ms  26.937ms
  8   180.149.48.18  48.666ms  27.400ms  32.795ms
  9   *  *  *
 10   *  *  *
 11   *  *  *
 12   *  *  *
 13   *  *  *
 14   *  *  *
 15   *  *  *
 16   *  *  *
 17   *  *  *
 18   *  *  *
 19   *  *  *
 20   *  *  *
 21   *  *  *
 22   *  *  *
 23   *  *  *
 24   171.66.255.200  289.714ms  288.898ms  306.313ms
 25   171.64.255.232  308.079ms  297.771ms  282.955ms
 26   *  *  *
 27   171.67.215.200  289.801ms  289.094ms  290.954ms
```

Intermediate hosts for stanford.edu: 10 (excluding the invisible hosts), 26 (including)
Intermediate hosts for google.in: 8 (excluding the invisible hosts), 9 (including)

h) Can you explain the reason for the latency difference between google.in and stanford.edu (see (b) & (f))?

The latency difference between them is because of the difference in distance between their servers and my device. The more the distance, the more hops the packet will have to make to reach the destination and the more latency for the website.