Varun Narravula
ID: 923-287-037

Prof. Qun Wang
CSC645 Computer Networks

Project 1

Task 1: Networking Commands

1. Using the `ifconfig` Command

Use `ifconfig` (or `ipconfig`) to check your IP address and MAC address.

I will be using the `ip` command from `iproute2`, since I am using NixOS, a Linux distribution.

```
$ ip addr show dev wlan0
3: wlan0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group
default qlen 1000
    link/ether c8:58:c0:c5:7c:f9 brd ff:ff:ff:ff:ff:ff
    altname wlp5s0
    inet 192.168.1.67/24 brd 192.168.1.255 scope global dynamic noprefixroute wlan0
       valid_lft 81066sec preferred_lft 81066sec
    inet6 2600:1700:3bd0:67d0:8cb6:276d:e704:c8f6/64 scope global temporary dynamic
       valid_lft 3468sec preferred_lft 3468sec
```

- IP address :: **`192.168.1.67`** (on private network)
- MAC address :: **`c8:58:c0:c5:7c:f9`**.

2. Using the `ping` Command

Use the `ping` command to ping your favorite website, such as google.com. Find out the IP address of that website server and the average delay. Use the ping command to ping with 6 packets to that IP address.

I'll be pinging https://snare.dev, my personal website.

```
$ ping -c 6 snare.dev
PING snare.dev (99.83.231.61) 56(84) bytes of data.
64 bytes from acd89244c803f7181.awsglobalaccelerator.com (99.83.231.61): icmp_seq=1
ttl=247 time=9.16 ms
64 bytes from acd89244c803f7181.awsglobalaccelerator.com (99.83.231.61): icmp_seq=2
ttl=247 time=9.19 ms
64 bytes from acd89244c803f7181.awsglobalaccelerator.com (99.83.231.61): icmp_seq=3
ttl=247 time=10.2 ms
64 bytes from acd89244c803f7181.awsglobalaccelerator.com (99.83.231.61): icmp_seq=4
ttl=247 time=9.98 ms
64 bytes from acd89244c803f7181.awsglobalaccelerator.com (99.83.231.61): icmp_seq=5
ttl=247 time=11.8 ms
64 bytes from acd89244c803f7181.awsglobalaccelerator.com (99.83.231.61): icmp_seq=6
ttl=247 time=10.7 ms

--- snare.dev ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5005ms
rtt min/avg/max/mdev = 9.156/10.170/11.844/0.921 ms
```

- IP Address: **`99.83.231.61`**
- Average Delay: **`10.17 ms`**

Use the ping command to ping yourself.

I can ping myself by pinging the `localhost`, `127.0.0.1`, or `::1` addresses.

```
$ ping -c 6 localhost
PING localhost (::1) 56 data bytes
64 bytes from localhost (::1): icmp_seq=1 ttl=64 time=0.047 ms
64 bytes from localhost (::1): icmp_seq=2 ttl=64 time=0.050 ms
64 bytes from localhost (::1): icmp_seq=3 ttl=64 time=0.055 ms
64 bytes from localhost (::1): icmp_seq=4 ttl=64 time=0.045 ms
64 bytes from localhost (::1): icmp_seq=5 ttl=64 time=0.095 ms
64 bytes from localhost (::1): icmp_seq=6 ttl=64 time=0.057 ms

--- localhost ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5114ms
rtt min/avg/max/mdev = 0.045/0.058/0.095/0.017 ms
```

3. Using the `traceroute` Command

Use the `traceroute` command to trace the routers for the IP address of your favorite website.

```
$ traceroute snare.dev
traceroute to snare.dev (75.2.60.5), 30 hops max, 60 byte packets
 1  homeportal (192.168.1.254)  3.521 ms  2.278 ms  4.123 ms
 2  162-226-172-1.lightspeed.sntcca.sbcglobal.net (162.226.172.1)  9.436 ms  9.418 ms
9.503 ms
 3  71.148.149.102 (71.148.149.102)  20.919 ms  20.901 ms  20.884 ms
 4  12.242.124.218 (12.242.124.218)  30.587 ms  22.637 ms  30.552 ms
 5  12.122.163.34 (12.122.163.34)  34.321 ms  35.201 ms  36.353 ms
 6  12.122.158.9 (12.122.158.9)  33.920 ms  31.846 ms  13.326 ms
 7  * * *
(truncated to 30 max hops, the rest are just * * *)
```

4. Using the `arp` Command

Use the `arp` command to show your ARP cache list.

On Linux, it is actually possible to use `procfs` to get the ARP cache list. The file `/proc/net/arp` contains it, so we can read it with `cat`.

```
$ cat /proc/net/arp
IP address       HW type     Flags       HW address            Mask     Device
192.168.1.122    0x1         0x2         28:6b:35:67:ce:51     *        wlan0
192.168.122.4    0x1         0x2         52:54:00:4e:22:f8     *        virbr0
192.168.1.254    0x1         0x2         f8:18:97:b2:fa:8d     *        wlan0
192.168.1.124    0x1         0x2         16:4c:7b:9c:6d:bd     *        wlan0
```

Task 2: Wireshark Tutorial

1. Using Wireshark to Capture Packets

Find a packet that is sent to you. Show that the destination IP and MAC address are the same as your device.

The packets that I captured and sent use IPV6 instead of IPV4.

```
(Physical Frame Data...)
Ethernet II, Src: 2Wire_b2:fa:8d (f8:18:97:b2:fa:8d), Dst: Intel_c5:7c:f9
(c8:58:c0:c5:7c:f9)
Destination: Intel_c5:7c:f9 (c8:58:c0:c5:7c:f9)
Source: 2Wire_b2:fa:8d (f8:18:97:b2:fa:8d)
Type: IPv6 (0x86dd)
[Stream index: 2]
Internet Protocol Version 6, Src: 2606:4700:4400::6812:26e9, Dst:
2600:1700:3bd0:67d0:8cb6:276d:e704:c8f6
0110 .... = Version: 6
.... 0000 0000 .... .... .... .... .... = Traffic Class: 0x00 (DSCP: CS0, ECN: Not-
ECT)
.... 0100 1110 1101 0101 1110 = Flow Label: 0x4ed5e
Payload Length: 1050
Next Header: TCP (6)
Hop Limit: 55
Source Address: 2606:4700:4400::6812:26e9
Destination Address: 2600:1700:3bd0:67d0:8cb6:276d:e704:c8f6
[Stream index: 8]
(TCP, HTTP, OCSP request...)
```

- IP Source Address :: **2600:1700:3bd0:67d0:8cb6:276d:e704:c8f6**
- Ethernet Frame MAC Source Address :: **f8:18:97:b2:fa:8d**

Find a packet that is sent by you. Show that the source IP and MAC address are the same as your device.

```
(Physical Frame Data...)
Ethernet II, Src: Intel_c5:7c:f9 (c8:58:c0:c5:7c:f9), Dst: 2Wire_b2:fa:8d
(f8:18:97:b2:fa:8d)
    Destination: 2Wire_b2:fa:8d (f8:18:97:b2:fa:8d)
    Source: Intel_c5:7c:f9 (c8:58:c0:c5:7c:f9)
    Type: IPv6 (0x86dd)
    [Stream index: 2]
Internet Protocol Version 6, Src: 2600:1700:3bd0:67d0:8cb6:276d:e704:c8f6, Dst:
2606:4700:4400::6812:26e9
    0110 .... = Version: 6
    .... 0000 0000 .... .... .... .... .... = Traffic Class: 0x00 (DSCP: CS0, ECN:
Not-ECT)
    .... 1011 0000 1111 0011 1111 = Flow Label: 0xb0f3f
    Payload Length: 463
    Next Header: TCP (6)
    Hop Limit: 64
    Source Address: 2600:1700:3bd0:67d0:8cb6:276d:e704:c8f6
    Destination Address: 2606:4700:4400::6812:26e9
    [Stream index: 8]
(TCP, HTTP, OCSP request...)
```

- IP Destination Address :: **2600:1700:3bd0:67d0:8cb6:276d:e704:c8f6**
- Ethernet Frame MAC Destination Address :: **f8:18:97:b2:fa:8d**