Tracerouting

Know your machine

- Use ifconfig to find the IP and MAC addresses of your virtual machine. (You'll note that the former will not start with 140.233, which is confusing because it's on the Middlebury network: this is because the virtual machine is hiding on a virtual network within your host.)

  Find out the IP and MAC addresses on the host operating system—on Mac, use ifconfig; on Windows, use ipconfig.

  Use the route command to find the IP address of your "default router"—ie, the machine that acts as gateway to the rest of the Internet.

On the virtual machine, when running ifconfig we get:

```
@majd 5]$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.2.15  netmask 255.255.255.0  broadcast 10.0.2.255
        inet6 fe80::353d:4b9d:680e:7fd7  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:75:2d:0b  txqueuelen 1000  (Ethernet)
        RX packets 183167  bytes 266142745 (253.8 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 16509  bytes 1412514 (1.3 MiB)
        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 40  bytes 2000 (1.9 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 40  bytes 2000 (1.9 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

It seems that our virtua; machine uses an ethernet interface called enp0s3 with Ip address 10.0.2.15 and Mac address 08:00:27:75:2d:0b.

On host device, we get:

```
C:\Users\Majd>ipconfig /all
...
...
...
Wireless LAN adapter Wi-Fi:

   Connection-specific DNS Suffix  . : middlebury.edu
   Description . . . . . . . . . . . : Intel(R) Dual Band Wireless-AC 8265
   Physical Address. . . . . . . . . : 30-24-32-AC-9C-BB
   DHCP Enabled. . . . . . . . . . . : Yes
   Autoconfiguration Enabled . . . . : Yes
```

```
   Link-local IPv6 Address . . . . . : fe80::c493:3757:1139:4b2f%14(
   Preferred)
   IPv4 Address. . . . . . . . . . . : 140.233.187.34(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.224.0
   Lease Obtained. . . . . . . . . . : Sunday, January 23, 2022 9:40:47 PM
   Lease Expires . . . . . . . . . . : Tuesday, January 25, 2022 1:41:40
   AM
   Default Gateway . . . . . . . . . : 140.233.160.1
   DHCP Server . . . . . . . . . . . : 140.233.1.4
   DHCPv6 IAID . . . . . . . . . . . : 523248690
   DHCPv6 Client DUID. . . . . . . . : 00-01-00-01-22-D4-FD-2F
   -10-65-30-82-F5-DD
   DNS Servers . . . . . . . . . . . : 140.233.253.2
                                       140.233.253.3
                                       140.233.2.204
                                       140.233.1.4
   NetBIOS over Tcpip. . . . . . . . : Enabled
```

so on the host device we have a wirless interface with ip 140.233.187.34 and mac address of 30-24-32-AC-9C-BB.

The two system have two different mac and ip addresses. This is because virtual box is creating an entirely different device withing the host device that connects to the outside world through the host device. What route does the inner device take to reach the outside world? We use route -n:

```
[]@majd 5]$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use
   Iface
0.0.0.0         10.0.2.2        0.0.0.0         UG    1002   0        0
   enp0s3
10.0.2.0        0.0.0.0         255.255.255.0   U     1002   0        0
   enp0s3
```

So to reach outside of the local network (destination of 0.0.0.0), we need to send packets to 10.0.2.2 which is the ip address of the machine that acts as a gateway to the rest of the internet.

1 "Automatic" traceroute

- install mtr, a tool that performs tracerouting. Use it to find the route to 72.14.176.147. Use wireshark to observe the packets. Find the TTL field in the binary data.

We get the following result when we run mtr:

It seems we are never getting a reply. We ran mtr with google.com, we got:

```
elp   Display mode   Restart statistics   Order of fields   quit

    Packets
             Pings
   Loss% Drop
 1. _gateway
   0.0%    0
 2. 140.233.160.3
   0.0%    0
 3. 140.233.9.254
   0.0%    0
 4. (waiting for reply)
 5. (waiting for reply)
 6. (waiting for reply)
 7. (waiting for reply)
 8. (waiting for reply)
 9. (waiting for reply)
10. (waiting for reply)
11. (waiting for reply)
12. lga25s79-in-f14.1e100.net
   0.0%    0
```

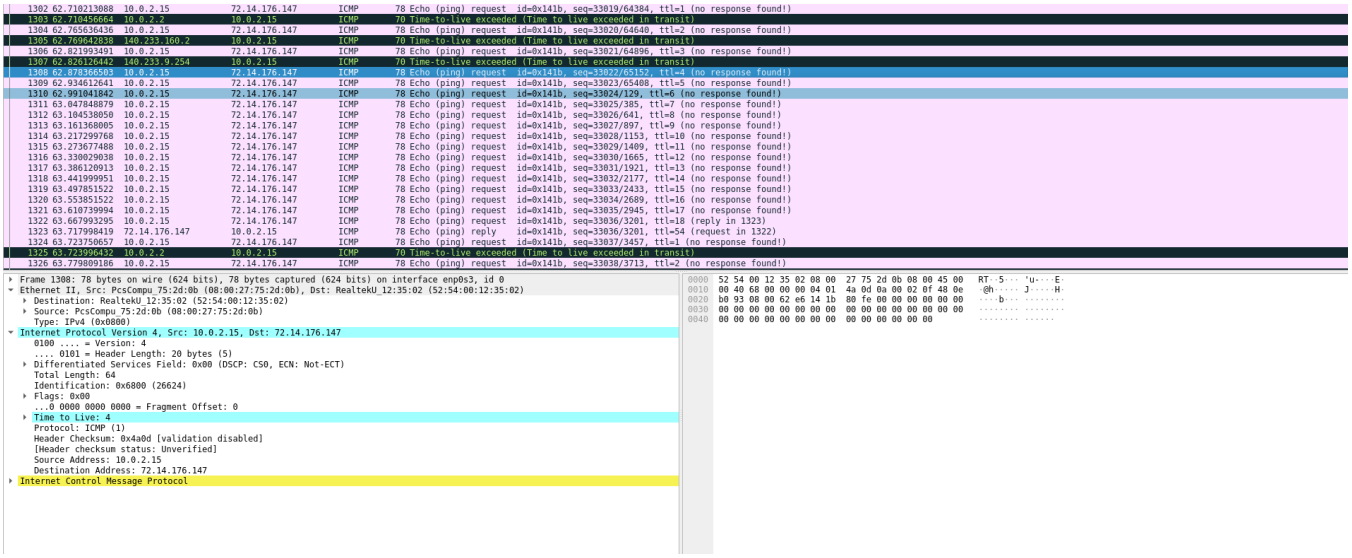
So are eventually getting somewhere. It seems that for the waiting for reply part that package is either getting lost or the server is never responding. We know that the servers are responding because we managed to see them when we connecting to a hotspot (other students work) so the package is getting lost (middlebury is blocking the package response in the name of security - Pete). But why we don't see that we got to the final destination when we run it on the ip address? After looking at mtr man page, we see that the time to live is set to 5 by default. So if the package need to make more than 5 hops, it will die and we won't see a response. We can increase the number of hops using -U 20 (increased it to 20 hops) (-LD so i can copy the stuff here because they keep refreshing):

```
@majd 5]$ mtr 72.14.176.147 -U 20 -o LD
             Pings
   Loss% Drop
 1. _gateway
   0.0%    0
 2. 140.233.160.2
   0.0%    0
 3. 140.233.9.254
   0.0%    0
 4. (waiting for reply)
 5. (waiting for reply)
 6. (waiting for reply)
 7. (waiting for reply)
 8. (waiting for reply)
 9. (waiting for reply)
10. (waiting for reply)
11. (waiting for reply)
```

```
12. (waiting for reply)
13. (waiting for reply)
14. (waiting for reply)
15. (waiting for reply)
16. (waiting for reply)
17. (waiting for reply)
18. menegroth.hiddenrock.com
    0.0%     0
```

success! we reach Pete's server at the 18th hop.

From mtr man page we understand that mtr investigates the network connection between the host mtr runs on and HOSTNAME by sending packets with purposely low TTLs. It continues to send packets with low TTL. This allows mtr to print the response percentage and response times of the internet route to HOSTNAME. mtr does this using ping. We can trace this in wireshark. When we run wireshark and select the ethernet interface, then run mtr, we get a lot of ping messages. We see that the TTL increases after every ping by one-just as mtr description says:

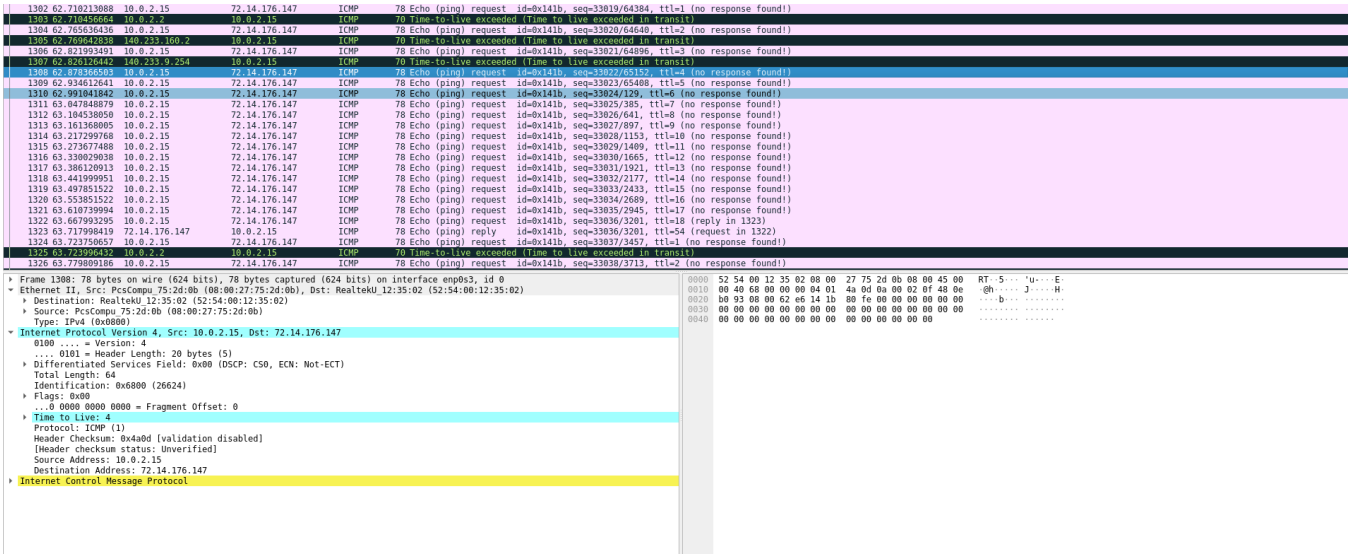


2 Manual traceroute

- Write a bash program that uses ping(8) to perform traceroute. It should take a single command-line argument—the destination IP address—and output the sequence of hops between the source and destination.

  The point is to get used to Wireshark, doing networky stuff, and composing various tools using shell scripting.

Tracerouting



To change the TTL for ping all we have to do is use -t number_of_hops.

Tracerouting