

## 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:

```

1 @majd 5]$ ifconfig
2 enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
3         inet 10.0.2.15  netmask 255.255.255.0  broadcast 10.0.2.255
4         inet6 fe80::353d:4b9d:680e:7fd7  prefixlen 64  scopeid 0x20<link>
5         ether 08:00:27:75:2d:0b  txqueuelen 1000  (Ethernet)
6         RX packets 183167  bytes 266142745 (253.8 MiB)
7         RX errors 0  dropped 0  overruns 0  frame 0
8         TX packets 16509  bytes 1412514 (1.3 MiB)
9         TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
10
11 lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
12        inet 127.0.0.1  netmask 255.0.0.0
13        inet6 ::1  prefixlen 128  scopeid 0x10<host>
14        loop txqueuelen 1000  (Local Loopback)
15        RX packets 40  bytes 2000 (1.9 KiB)
16        RX errors 0  dropped 0  overruns 0  frame 0
17        TX packets 40  bytes 2000 (1.9 KiB)
18        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

```

It seems that our virtual 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:

```

1 C:\Users\Majd>ipconfig /all
2 ...
3 ...
4 ...
5 Wireless LAN adapter Wi-Fi:
6
7     Connection-specific DNS Suffix  . : middlebury.edu
8     Description . . . . . : Intel(R) Dual Band Wireless-AC 8265
9     Physical Address. . . . . : 30-24-32-AC-9C-BB
10    DHCP Enabled. . . . . : Yes
11    Autoconfiguration Enabled . . . . : Yes

```

```

12 Link-local IPv6 Address . . . . . : fe80::c493:3757:1139:4b2f%14(
    Preferred)
13 IPv4 Address. . . . . : 140.233.187.34(Preferred)
14 Subnet Mask . . . . . : 255.255.224.0
15 Lease Obtained. . . . . : Sunday, January 23, 2022 9:40:47 PM
16 Lease Expires . . . . . : Tuesday, January 25, 2022 1:41:40
    AM
17 Default Gateway . . . . . : 140.233.160.1
18 DHCP Server . . . . . : 140.233.1.4
19 DHCPv6 IAID . . . . . : 523248690
20 DHCPv6 Client DUID. . . . . : 00-01-00-01-22-D4-FD-2F
    -10-65-30-82-F5-DD
21 DNS Servers . . . . . : 140.233.253.2
22                        140.233.253.3
23                        140.233.2.204
24                        140.233.1.4
25 NetBIOS over Tcpip. . . . . : Enabled

```

so on the host device we have a wireless 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:

```

1 []@majd 5]$ route -n
2 Kernel IP routing table
3 Destination      Gateway            Genmask           Flags Metric Ref    Use
4 0.0.0.0           10.0.2.2          0.0.0.0           UG    1002   0      0
   enp0s3
5 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:

```

1 elp      Display mode      Restart statistics      Order of fields      quit
2
3           Packets
4           Loss% Drop
5 1. _gateway
6   0.0%      0
7 2. 140.233.160.3
8   0.0%      0
9 3. 140.233.9.254
10  0.0%      0
11 4. (waiting for reply)
12 5. (waiting for reply)
13 6. (waiting for reply)
14 7. (waiting for reply)
15 8. (waiting for reply)
16 9. (waiting for reply)
17 10. (waiting for reply)
18 11. (waiting for reply)
19 12. lga25s79-in-f14.1e100.net
20   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):

```

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

```

```

14 12. (waiting for reply)
15 13. (waiting for reply)
16 14. (waiting for reply)
17 15. (waiting for reply)
18 16. (waiting for reply)
19 17. (waiting for reply)
20 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:

1302	62.718213688	10.0.2.15	72.14.176.147	ICMP	78 Echo (ping) request	id=0x141b, seq=33019/64384, ttl=1 (no response found!)	
1303	62.718456664	10.0.2.15	72.14.176.147	ICMP	70 Time to live exceeded (Time to live exceeded in transit!)		
1304	62.765055425	10.0.2.15	72.14.176.147	ICMP	78 Echo (ping) request	id=0x141b, seq=33020/64384, ttl=2 (no response found!)	
1305	62.769442338	140.233.160.2	10.0.2.15	ICMP	70 Time to live exceeded (Time to live exceeded in transit!)		
1306	62.821993491	10.0.2.15	72.14.176.147	ICMP	78 Echo (ping) request	id=0x141b, seq=33021/64384, ttl=3 (no response found!)	
1307	62.824545442	140.233.160.2	10.0.2.15	ICMP	70 Time to live exceeded (Time to live exceeded in transit!)		
1308	62.879366593	10.0.2.15	72.14.176.147	ICMP	78 Echo (ping) request	id=0x141b, seq=33022/64384, ttl=4 (no response found!)	
1309	62.934612641	10.0.2.15	72.14.176.147	ICMP	78 Echo (ping) request	id=0x141b, seq=33023/64384, ttl=5 (no response found!)	
1310	62.991841842	10.0.2.15	72.14.176.147	ICMP	78 Echo (ping) request	id=0x141b, seq=33024/64384, ttl=6 (no response found!)	
1311	63.047848879	10.0.2.15	72.14.176.147	ICMP	78 Echo (ping) request	id=0x141b, seq=33025/64384, ttl=7 (no response found!)	
1312	63.184538858	10.0.2.15	72.14.176.147	ICMP	78 Echo (ping) request	id=0x141b, seq=33026/64384, ttl=8 (no response found!)	
1313	63.161368895	10.0.2.15	72.14.176.147	ICMP	78 Echo (ping) request	id=0x141b, seq=33027/64384, ttl=9 (no response found!)	
1314	63.217299768	10.0.2.15	72.14.176.147	ICMP	78 Echo (ping) request	id=0x141b, seq=33028/64384, ttl=10 (no response found!)	
1315	63.273677488	10.0.2.15	72.14.176.147	ICMP	78 Echo (ping) request	id=0x141b, seq=33029/64384, ttl=11 (no response found!)	
1316	63.338028038	10.0.2.15	72.14.176.147	ICMP	78 Echo (ping) request	id=0x141b, seq=33030/64384, ttl=12 (no response found!)	
1317	63.386120913	10.0.2.15	72.14.176.147	ICMP	78 Echo (ping) request	id=0x141b, seq=33031/64384, ttl=13 (no response found!)	
1318	63.441999951	10.0.2.15	72.14.176.147	ICMP	78 Echo (ping) request	id=0x141b, seq=33032/64384, ttl=14 (no response found!)	
1319	63.497815122	10.0.2.15	72.14.176.147	ICMP	78 Echo (ping) request	id=0x141b, seq=33033/64384, ttl=15 (no response found!)	
1320	63.553851522	10.0.2.15	72.14.176.147	ICMP	78 Echo (ping) request	id=0x141b, seq=33034/64384, ttl=16 (no response found!)	
1321	63.610739994	10.0.2.15	72.14.176.147	ICMP	78 Echo (ping) request	id=0x141b, seq=33035/64384, ttl=17 (no response found!)	
1322	63.667993295	10.0.2.15	72.14.176.147	ICMP	78 Echo (ping) request	id=0x141b, seq=33036/64384, ttl=18 (reply in 1322)	
1323	63.717998419	72.14.176.147	10.0.2.15	ICMP	78 Echo (ping) reply	id=0x141b, seq=33036/3201, ttl=54 (request in 1322)	
1324	63.723750657	10.0.2.15	72.14.176.147	ICMP	78 Echo (ping) request	id=0x141b, seq=33037/64384, ttl=1 (no response found!)	
1325	63.729244472	10.0.2.15	72.14.176.147	ICMP	70 Time to live exceeded (Time to live exceeded in transit!)		
1326	63.779899186	10.0.2.15	72.14.176.147	ICMP	78 Echo (ping) request	id=0x141b, seq=33038/64384, ttl=2 (no response found!)	

  

<p>Frame 1308: 78 bytes on wire (624 bits), 78 bytes captured (624 bits) on interface enp0s3, id 0</p> <p>Ethernet II, Src: PcsCompu, 75:2d:0b:08:00:27:75:2d:0b, Dst: RealtekU, 12:35:02:52:54:00:12:35:02</p> <p>Destination: RealtekU, 12:35:02:52:54:00:12:35:02</p> <p>Source: PcsCompu, 75:2d:0b:08:00:27:75:2d:0b</p> <p>Type: IPv4 (0x0800)</p> <p>Internet Protocol Version 4, Src: 10.0.2.15, Dst: 72.14.176.147</p> <p>0100 .... = Version: 4</p> <p>.... 0101 = Header Length: 20 bytes (5)</p> <p>Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)</p> <p>Total Length: 64</p> <p>Identification: 0x6800 (26624)</p> <p>Flags: 0x00</p> <p>...0 0000 0000 0000 = Fragment Offset: 0</p> <p>Time to Live: 4</p> <p>Protocol: ICMP (1)</p> <p>Header Checksum: 0x4a0d (validation disabled)</p> <p>[Header checksum status: Unverified]</p> <p>Source Address: 10.0.2.15</p> <p>Destination Address: 72.14.176.147</p> <p>Internet Control Message Protocol</p>	<p>0000 52 54 00 12 35 02 08 00 27 75 2d 0b 08 00 45 00 RT-5...u...E</p> <p>0010 00 40 68 00 00 00 04 01 4a 0d 0a 00 02 0f 48 0e @h....J....H</p> <p>0020 00 93 00 00 62 e5 14 1b 00 fe 00 00 00 00 00 00 .....b.....</p> <p>0030 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....0000000000000000</p> <p>0040 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....0000000000000000</p>
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

## 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.

1302	62.710213088	10.0.2.15	72.14.176.147	ICMP	78 Echo (ping) request	id=0x141b, seq=33019/64384, ttl=1 (no response found!)	
1303	62.710456064	10.0.2.15	72.14.176.147	ICMP	78 Echo (ping) request	id=0x141b, seq=33020/64400, ttl=2 (no response found!)	
1304	62.765056416	10.0.2.15	72.14.176.147	ICMP	78 Echo (ping) request	id=0x141b, seq=33021/64416, ttl=3 (no response found!)	
1305	62.769642838	140.233.160.2	10.0.2.15	ICMP	78 Echo (ping) request	id=0x141b, seq=33022/64432, ttl=4 (no response found!)	
1306	62.821993491	10.0.2.15	72.14.176.147	ICMP	78 Echo (ping) request	id=0x141b, seq=33023/64448, ttl=5 (no response found!)	
1307	62.826584044	10.0.2.15	72.14.176.147	ICMP	78 Echo (ping) request	id=0x141b, seq=33024/64464, ttl=6 (no response found!)	
1308	62.878366563	10.0.2.15	72.14.176.147	ICMP	78 Echo (ping) request	id=0x141b, seq=33025/64480, ttl=7 (no response found!)	
1309	62.934612641	10.0.2.15	72.14.176.147	ICMP	78 Echo (ping) request	id=0x141b, seq=33026/64496, ttl=8 (no response found!)	
1310	62.991814842	10.0.2.15	72.14.176.147	ICMP	78 Echo (ping) request	id=0x141b, seq=33027/64512, ttl=9 (no response found!)	
1311	63.047848879	10.0.2.15	72.14.176.147	ICMP	78 Echo (ping) request	id=0x141b, seq=33028/64528, ttl=10 (no response found!)	
1312	63.104538858	10.0.2.15	72.14.176.147	ICMP	78 Echo (ping) request	id=0x141b, seq=33029/64544, ttl=11 (no response found!)	
1313	63.161368895	10.0.2.15	72.14.176.147	ICMP	78 Echo (ping) request	id=0x141b, seq=33030/64560, ttl=12 (no response found!)	
1314	63.217299768	10.0.2.15	72.14.176.147	ICMP	78 Echo (ping) request	id=0x141b, seq=33031/64576, ttl=13 (no response found!)	
1315	63.273677488	10.0.2.15	72.14.176.147	ICMP	78 Echo (ping) request	id=0x141b, seq=33032/64592, ttl=14 (no response found!)	
1316	63.330029038	10.0.2.15	72.14.176.147	ICMP	78 Echo (ping) request	id=0x141b, seq=33033/64608, ttl=15 (no response found!)	
1317	63.386120913	10.0.2.15	72.14.176.147	ICMP	78 Echo (ping) request	id=0x141b, seq=33034/64624, ttl=16 (no response found!)	
1318	63.441999951	10.0.2.15	72.14.176.147	ICMP	78 Echo (ping) request	id=0x141b, seq=33035/64640, ttl=17 (no response found!)	
1319	63.497815122	10.0.2.15	72.14.176.147	ICMP	78 Echo (ping) request	id=0x141b, seq=33036/64656, ttl=18 (no response found!)	
1320	63.553851522	10.0.2.15	72.14.176.147	ICMP	78 Echo (ping) request	id=0x141b, seq=33037/64672, ttl=19 (no response found!)	
1321	63.610739994	10.0.2.15	72.14.176.147	ICMP	78 Echo (ping) request	id=0x141b, seq=33038/64688, ttl=20 (no response found!)	
1322	63.667693295	10.0.2.15	72.14.176.147	ICMP	78 Echo (ping) request	id=0x141b, seq=33039/64704, ttl=21 (no response found!)	
1323	63.717998419	72.14.176.147	10.0.2.15	ICMP	78 Echo (ping) reply	id=0x141b, seq=33036/3201, ttl=18 (reply in 1322)	
1324	63.723758657	10.0.2.15	72.14.176.147	ICMP	78 Echo (ping) request	id=0x141b, seq=33040/64720, ttl=22 (no response found!)	
1325	63.780000000	10.0.2.15	72.14.176.147	ICMP	78 Echo (ping) request	id=0x141b, seq=33041/64736, ttl=23 (no response found!)	
1326	63.779899186	10.0.2.15	72.14.176.147	ICMP	78 Echo (ping) request	id=0x141b, seq=33038/3713, ttl=2 (no response found!)	

  

Frame 1308: 78 bytes on wire (624 bits), 78 bytes captured (624 bits) on interface enp0s3, id 0		0000 52 54 00 12 35 02 08 00 27 75 2d 0b 08 00 45 00 RT: 5...u...E
Ethernet II, Src: PcsCompu 75:2d:0b:08:00:27:75:2d:0b, Dst: RealtekU 12:35:02:52:54:00:12:35:02		0010 00 40 00 00 00 04 01 4a 0d 0a 00 02 0f 48 0e @h...J...H
Destination: RealtekU 12:35:02:52:54:00:12:35:02		0020 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....b.....
Source: PcsCompu 75:2d:0b:08:00:27:75:2d:0b		0030 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....B.....
Type: IPv4 (0x0800)		0040 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....d.....
Internet Protocol Version 4, Src: 10.0.2.15, Dst: 72.14.176.147		
0100 .... = Version: 4		
... 0101 = Header Length: 20 bytes (5)		
Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)		
Total Length: 64		
Identification: 0x0000 (26024)		
Flags: 0x00		
... 0 0000 0000 0000 = Fragment Offset: 0		
Time to Live: 4		
Protocol: ICMP (1)		
Header Checksum: 0x4a0d (validation disabled)		
[Header checksum status: Unverified]		
Source Address: 10.0.2.15		
Destination Address: 72.14.176.147		
Internet Control Message Protocol		

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



1 .PNG