**Overview**

The following lab has been designed to be completed on a system running MS Windows. If you think you would like to repeat this lab later on a computer running MacOS or Linux I have pointed out some differences but you may need to adjust the instructions slightly – or you can download the “COMP-CO910-Labs.ova” file from the course VM collection and open it in Oracle VM Virtualbox (your professor will discuss this process with you if needed).

I have highlighted the spots where you need to provide answers in red.

**Outcomes**

This lab is designed to familiarize you with basic file and directory management on a common OS. You will also do some personal research to answer additional lab questions related to the exercises.

|  |  |
| --- | --- |
| Elements of Performance | Related Course Learning Outcomes |
| EOP 3.5. Use command-line applications to obtain network information and to troubleshoot network related problems | CLO 3. Understand TCP/IP fundamentals and visualize data flow on a TCP/IP network |

**Time on Task**

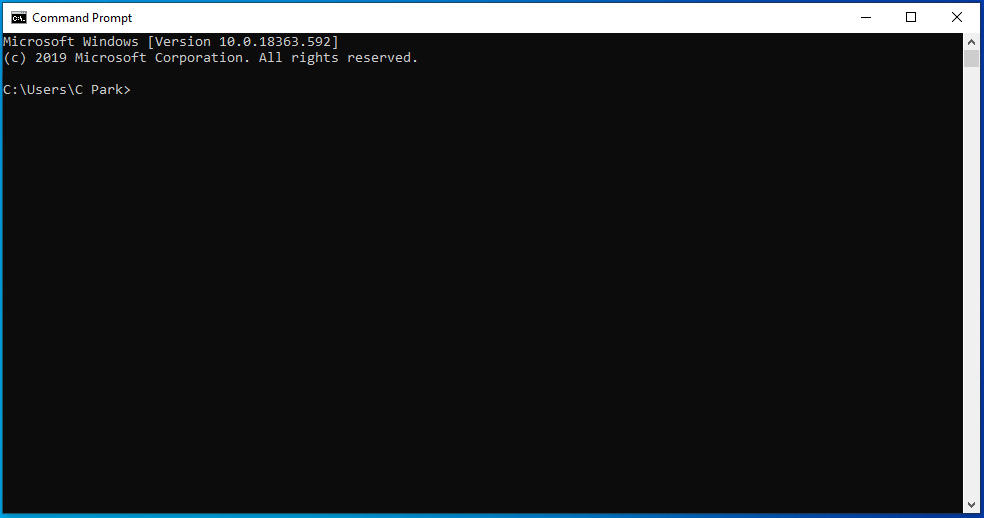
|  |  |
| --- | --- |
| Learning Activity | Approximate Time |
| Perform hands-on lab tasks and answer lab questions | 20-30 min |

Let’s get started!

**Lab Instructions and Questions**

1. Log into your OS and open a command window

* MS Windows users can do this quickly by pressing the “windows” key, typing **cmd** and clicking on the “Command Prompt” icon)
* If you’re doing this later on a system running Linux or MacOS and you’re comfortable translating some of these instructions, you can open a terminal window – if you get stuck along the way feel free to E-mail your professor for advice (they may be busy or unable pick up email at the exact moment you sent it, so they might answer you later or they may recommend another approach)



1. Before we do anything related to the network it sure would be nice to know if our default network device is working properly, right? It turns out that we have a great tool for that: ping!   
     
   We can use ping to check this by *pinging* (yes, that’s what we call it) something called “the loopback address”, which is a network address that *should* always be available in a working network configuration.  
     
   In the command window, type: **ping -n 5 127.0.0.1**MacOS/Linux users should type: **ping -c 5 127.0.0.1**  
     
   You should see output similar to this, although MacOS/Linux users will see slightly different (and more informative) output:  
     
   Pinging 127.0.0.1 with 32 bytes of data:

Reply from 127.0.0.1: bytes=32 time<1ms TTL=128

Reply from 127.0.0.1: bytes=32 time<1ms TTL=128

Reply from 127.0.0.1: bytes=32 time<1ms TTL=128

Reply from 127.0.0.1: bytes=32 time<1ms TTL=128

Reply from 127.0.0.1: bytes=32 time<1ms TTL=128

Ping statistics for 127.0.0.1:

Packets: Sent = 5, Received = 5, Lost = 0 (0% loss),

Approximate round trip times in milli-seconds:

Minimum = 0ms, Maximum = 0ms, Average = 0msIf you don’t see output like that first make sure you typed the command correctly, then ask the Professor if they have any thoughts about why you have something different.  
  
Note: If you only have three Reply from… lines and you’re running on Windows you probably forgot the -n 5 in your command, but if the replies never stop and you’re on MacOS or Linux you probably forgot the -c 5 in your command.

1. Now that we know our network device is working, let’s use ping to find out if we can reach one of the world’s most important websites,   
   **cat-bounce.com**  
     
   If you are using Windows, type: **ping -n 5 cat-bounce.com**  
   If you are on MacOS or Linux, type: **ping -c 5 cat-bounce.com**  
     
   You should see output resembling:  
     
   Pinging cat-bounce.com [162.219.51.2] with 32 bytes of data:

Reply from 208.113.161.95: bytes=32 time=2.5ms TTL=51

Reply from 208.113.161.95: bytes=32 time=3.5ms TTL=51

Reply from 208.113.161.95: bytes=32 time=2.5ms TTL=51

Reply from 208.113.161.95: bytes=32 time=3.75ms TTL=51

Reply from 208.113.161.95: bytes=32 time=3ms TTL=51

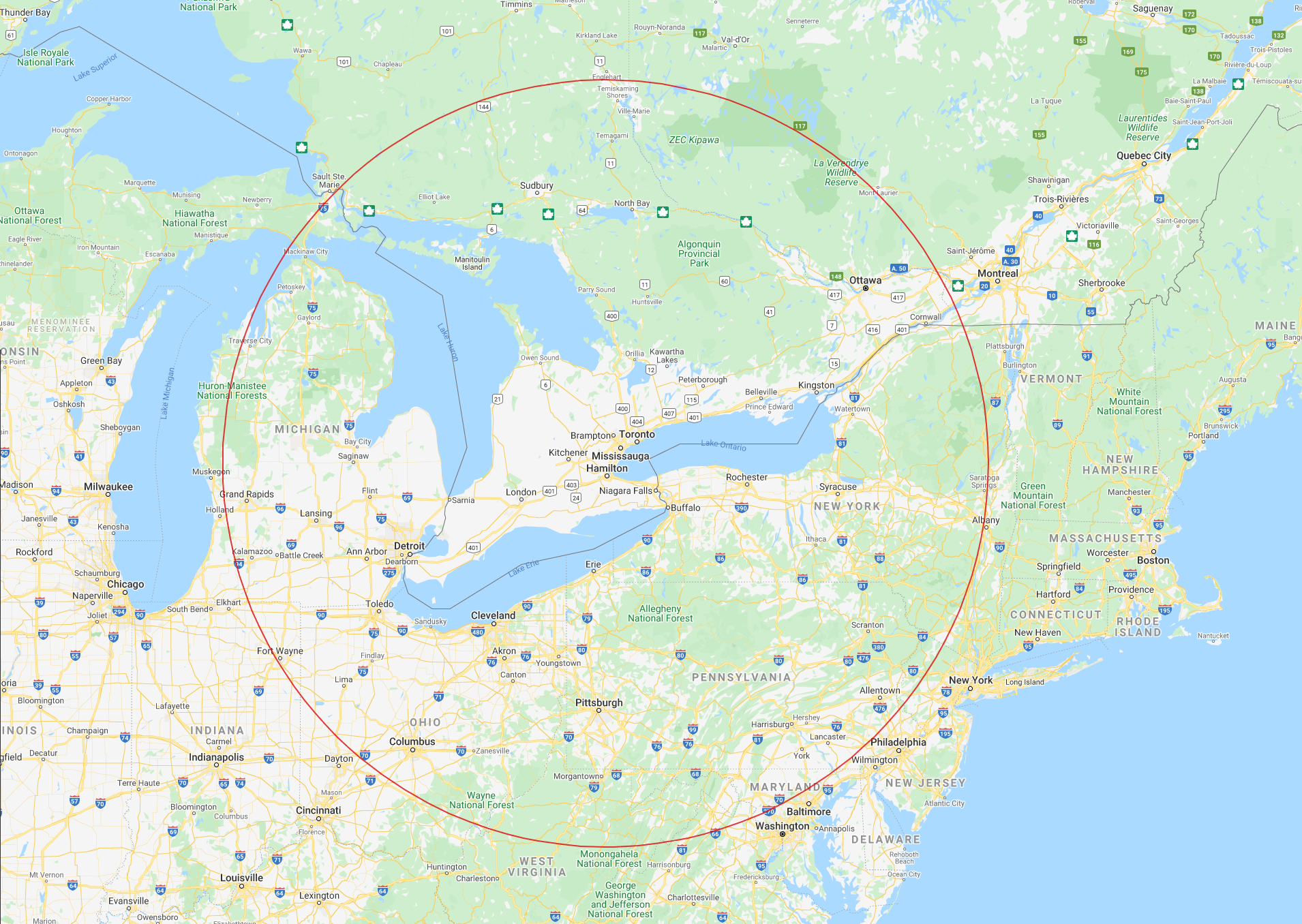
Ping statistics for 162.219.51.2:

Packets: Sent = 5, Received = 5, Lost = 0 (0% loss),

Approximate round trip times in milli-seconds:

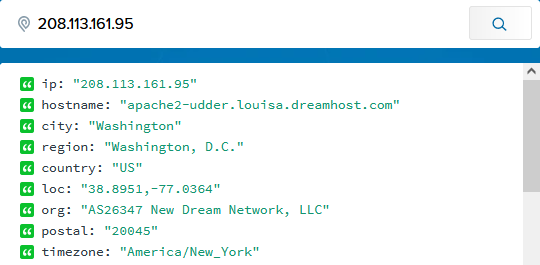
Minimum = 2.5ms, Maximum = 3.75ms, Average = 3ms  
  
This shows us that packets from our computer made it to the computer that hosts the cat-bounce.com website **and** that packets from that computer made it back to us. It also gives us a sense as to how fast the round-trip was (in the example output above we see an average of 3 milliseconds – that’s 3/1000th of a second (0.003 seconds), which is pretty fast. You may have different results – I wrote this at 2:00 AM, which was a very quiet time at the college and elsewhere in this timezone.

Let’s do some fun (if diversionary) math (yay)! We’ll make two assumptions: no delay in handing off the packet at any router or switch (which is not true, but it’s useful here) and that our data is travelling at the speed of light through our network media (which is almost true, so let’s go with it). Our packets are travelling at about 299,800 km/s, so in   
0.005 seconds they travel just under 900 km. This is the total distance in our return trip, so we know that the server is about 450 km away from us. We’re in Hamilton, Ontario, so let’s see what’s at a distance of 450 km:



It’s a little hard to make out the cities on a map this small, but it looks like the US cities of Albany in New York State, Baltimore in the state of Maryland, Fort Wayne in Indiana, Columbus in Ohio and Grand Rapids in the state of Michigan as well as Cornwall and Sault St. Marie here in Ontario are some of the larger cities at the 435 km distance.

On the day I wrote this I went to the site [**https://ipinfo.io**](https://ipinfo.io)and type in the IP address of cat-bounce.com (162.219.51.2) into the search box, and saw that the site is registered in Washington, DC (USA):



Washington, DC is a pretty expensive place to put a data centre, so I bet the computer with that site on it is in nearby Baltimore – which agrees with our unscientific but fun math diversion!

1. Ping is the go-to tool for systems administrators, software developers, or anyone else who wants to know if a network resource is online. Let’s see what happens when a site is **not** available…  
     
   Windows users should now type: **ping -n 5 142.222.6.180**MacOS and Linux users should instead type: **ping -c 5 142.222.6.180**  
     
   This address belongs to Mohawk College, but is not assigned to a live network device (as of the time this lab was written, of course). Therefore, Windows users should see output similar to:  
     
   Pinging 142.222.6.180 with 32 bytes of data:

Request timed out.

Request timed out.

Request timed out.

Request timed out.  
Request timed out.

Ping statistics for 142.222.6.180:

Packets: Sent = 5, Received = 0, Lost = 5 (100% loss),  
MacOS and Linux users may notice the output is a little more terse:

PING 142.22.6.180 (142.22.6.180) 56(84) bytes of data.

--- 142.22.6.180 ping statistics ---

5 packets transmitted, 0 received, 100% packet loss, time 4127ms

So now we have a tool in our toolbox that we can use to test if a network device is reachable by our system. Truly useful!

1. Next, let’s see if we can follow the path that our packet takes across our network and the Internet itself!   
     
   We’re going to use a new command here called *traceroute* (on Windows it’s named *tracert* because back in the “olden days” file names on earlier Microsoft operating systems could only be up to eight characters in length).  
     
   Windows users should now type: **tracert cat-bounce.com**  
   MacOS/Linux users will type: **traceroute cat-bounce.com**

Let’s take a look at the output from a sample run on a Windows node:

Tracing route to cat-bounce.com [208.113.161.95]

over a maximum of 30 hops:

1 <1 ms <1 ms <1 ms 192.168.10.1

2 7 ms 7 ms 7 ms 10.67.64.1

3 8 ms 8 ms 9 ms 10.0.81.9

4 8 ms 10 ms 13 ms 10.0.18.78

5 9 ms 8 ms 9 ms be5076.rcr21.yhm01.atlas.cogentco.com [38.122.17.81]

6 10 ms 9 ms 10 ms be2621.ccr31.yyz02.atlas.cogentco.com [154.54.40.77]

7 18 ms 16 ms 17 ms be2993.ccr21.cle04.atlas.cogentco.com [154.54.31.225]

8 28 ms 28 ms 34 ms be2891.ccr41.dca01.atlas.cogentco.com [154.54.82.250]

9 29 ms 30 ms 29 ms be2658.ccr22.iad02.atlas.cogentco.com [154.54.47.138]

10 31 ms 28 ms 35 ms be3563.rcr51.b037327-0.iad02.atlas.cogentco.com [154.54.25.226]

11 30 ms 33 ms 29 ms 38.122.62.254

12 32 ms 30 ms 28 ms iad1-cr-2.sd.dreamhost.com [208.113.156.58]

13 \* 31 ms 37 ms ip-208-113-156-49.dreamhost.com [208.113.156.49]

14 29 ms 30 ms 34 ms apache2-udder.louisa.dreamhost.com [208.113.161.95]

Trace complete.

This tool sends three requests (they’re very similar to pings, but not quite the same) to every network node along the way from the source node to the destination node. If possible, the node responds to the request. Sometimes the response gets lost along the way, in which case we see an \* in one of the response columns (see response line 13 above). Your output will be different than this as the path your packets will take will be different – in fact, if we were to run this command from the same node again the output may be different because the best route may have changed.

Looking at the output from a Linux system where traceroute was run (and at the same time as the tracert command we just looked at the output for was run on a Windows node) shows the different paths that packets can take even during the single time the command is run:

traceroute to cat-bounce.com (208.113.161.95), 30 hops max, 60 byte packets

1 192.168.10.1 (192.168.10.1) 0.091 ms 0.077 ms 0.088 ms

2 10.67.64.1 (10.67.64.1) 7.939 ms 7.908 ms 7.928 ms

3 10.0.81.9 (10.0.81.9) 9.585 ms 9.555 ms 9.505 ms

4 10.0.18.78 (10.0.18.78) 9.512 ms 9.464 ms 9.434 ms

5 be5076.rcr21.yhm01.atlas.cogentco.com (38.122.17.81) 9.023 ms 9.429 ms 9.395 ms

6 be2621.ccr31.yyz02.atlas.cogentco.com (154.54.40.77) 10.140 ms be2622.ccr32.yyz02.a

tlas.cogentco.com (154.54.40.101) 14.985 ms be2621.ccr31.yyz02.atlas.cogentco.com (154.  
54.40.77) 14.963 ms

7 be2994.ccr22.cle04.atlas.cogentco.com (154.54.31.233) 21.582 ms 16.296 ms be2993.c  
cr21.cle04.atlas.cogentco.com (154.54.31.225) 16.184 ms

8 be2891.ccr41.dca01.atlas.cogentco.com (154.54.82.250) 27.806 ms be2892.ccr42.dca01.  
atlas.cogentco.com (154.54.82.254) 31.178 ms be2891.ccr41.dca01.atlas.cogentco.com (154   
.54.82.250) 31.126 ms

9 be3084.ccr41.iad02.atlas.cogentco.com (154.54.30.66) 31.727 ms be2676.ccr22.iad02.a  
tlas.cogentco.com (154.54.47.166) 31.932 ms be3084.ccr41.iad02.atlas.cogentco.com (154.  
54.30.66) 32.073 ms

10 be3562.rcr51.b037327-0.iad02.atlas.cogentco.com (154.54.24.234) 31.626 ms 31.862 m  
s 31.869 ms

11 38.122.62.254 (38.122.62.254) 31.623 ms 31.597 ms 31.569 ms

12 iad1-cr-2.sd.dreamhost.com (208.113.156.58) 31.559 ms iad1-cr-1.sd.dreamhost.com (2  
08.113.156.208) 29.433 ms iad1-cr-2.sd.dreamhost.com (208.113.156.58) 33.400 ms

13 ip-208-113-156-49.dreamhost.com (208.113.156.49) 35.733 ms \* 33.527 ms

14 \* \* \*

The output here is similar (if a bit more precise), but in a few of the responses we can clearly see the different paths taken by some of the requests. For instance, in response lines such as #6 we can see that the second request followed a slightly different path. Interestingly, the final response is simply “\* \* \*” which means that none of the requests were replied to. This is not uncommon – in fact, you’ll see it a lot if you try this command on different destinations. It simply means that the node in that place in the chain did not reply (or its replies were lost), and it typically indicates that the node is either configured not to send replies to traceroute (or ping, usually, if responses to traceroute are disabled) or those responses have been blocked by a firewall or router along the way (usually for security purposes).

Now it’s time for you to investigate things for yourself!  
  
In the areas below paste screenshots (or grab the text, if you know how) of the responses to the commands you’ll be instructed to run. Your output (and perhaps even the details of the command you need to run) may be different than other students (yes, even someone next to you)!  
  
Show the output of your attempt to ping the search engine you like to use (google.com, google.ca, google.co.in, bing.com (?!), duckduckgo.com, etc.):  
  
C:\Users\000970878>ping google.com

Pinging google.com [142.251.41.78] with 32 bytes of data:

Reply from 142.251.41.78: bytes=32 time=5ms TTL=113

Reply from 142.251.41.78: bytes=32 time=5ms TTL=113

Reply from 142.251.41.78: bytes=32 time=5ms TTL=113

Reply from 142.251.41.78: bytes=32 time=5ms TTL=113

Ping statistics for 142.251.41.78:

Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),

Approximate round trip times in milli-seconds:

Minimum = 5ms, Maximum = 5ms, Average = 5ms  
  
  
Show the output of your attempt to ping www.mohawkcollege.ca:  
  
C:\Users\000970878>ping www.mohawkcollege.ca

Pinging www.mohawkcollege.ca [35.182.220.128] with 32 bytes of data:

Reply from 35.182.220.128: bytes=32 time=11ms TTL=52

Reply from 35.182.220.128: bytes=32 time=12ms TTL=52

Reply from 35.182.220.128: bytes=32 time=12ms TTL=52

Reply from 35.182.220.128: bytes=32 time=12ms TTL=52

Ping statistics for 35.182.220.128:

Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),

Approximate round trip times in milli-seconds:

Minimum = 11ms, Maximum = 12ms, Average = 11ms  
  
  
Show the output of your attempt to tracert (or traceroute) www.mohawkcollege.ca site:  
  
C:\Users\000970878>tracert www.mohawkcollege.ca

Tracing route to www.mohawkcollege.ca [35.182.220.128]

over a maximum of 30 hops:

1 <1 ms <1 ms <1 ms 142.222.95.253

2 <1 ms <1 ms <1 ms 10.255.0.10

3 1 ms <1 ms 1 ms 10.255.0.3

4 <1 ms <1 ms <1 ms 10.255.3.28

5 2 ms 2 ms 2 ms 10.255.3.35

6 1 ms 1 ms 1 ms unallocated-static.rogers.com [72.138.50.17]

7 1 ms 1 ms 1 ms 209.148.225.225

8 4 ms 5 ms 7 ms cpe00222d163efa-cm00222d163ef6.cpe.net.cable.rogers.com [72.139.133.230]

9 5 ms 4 ms 4 ms 9044-cgw01.mtnk.asr9k.rmgt.net.rogers.com [209.148.230.53]

10 4 ms 4 ms 4 ms 209.148.235.222

11 \* \* \* Request timed out.

12 \* \* \* Request timed out.

13 \* \* \* Request timed out.

14 11 ms 11 ms 12 ms 52.94.83.95

15 12 ms 12 ms 12 ms 52.94.83.126

16 12 ms 12 ms 13 ms 52.94.81.249

17 13 ms 12 ms 12 ms 52.94.81.50

18 \* \* \* Request timed out.

19 \* \* \* Request timed out.

20 \* \* \* Request timed out.

21 \* \* \* Request timed out.

22 \* \* \* Request timed out.

23 \* \* \* Request timed out.

24 12 ms 12 ms 12 ms ec2-35-182-220-128.ca-central-1.compute.amazonaws.com [35.182.220.128]

Trace complete.  
  
  
Show the output of your attempt to ping mycanvas.mohawkcollege.ca:  
  
C:\Users\000970878>ping mycanvas.mohawkcollege.ca

Pinging canvas-yul-prod-c263-57015929.ca-central-1.elb.amazonaws.com [15.222.200.155] with 32 bytes of data:

Reply from 15.222.200.155: bytes=32 time=10ms TTL=243

Reply from 15.222.200.155: bytes=32 time=11ms TTL=243

Reply from 15.222.200.155: bytes=32 time=11ms TTL=243

Reply from 15.222.200.155: bytes=32 time=11ms TTL=243

Ping statistics for 15.222.200.155:

Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),

Approximate round trip times in milli-seconds:

Minimum = 10ms, Maximum = 11ms, Average = 10ms  
  
  
You might notice the addresses here are ***very*** different – they’re not part of the same general range. Show your output from a tracert (or traceroute) to the same address:  
  
C:\Users\000970878>tracert mycanvas.mohawkcollege.ca

Tracing route to canvas-yul-prod-c263-57015929.ca-central-1.elb.amazonaws.com [15.222.200.155]

over a maximum of 30 hops:

1 <1 ms <1 ms <1 ms 142.222.95.253

2 1 ms <1 ms <1 ms 10.255.0.9

3 1 ms <1 ms <1 ms 10.255.0.4

4 1 ms <1 ms <1 ms 10.255.3.28

5 2 ms 1 ms 1 ms 10.255.3.35

6 3 ms 3 ms 2 ms 199.243.248.209

7 4 ms 4 ms 4 ms 10.98.14.61

8 \* \* \* Request timed out.

9 4 ms 5 ms 6 ms 142.124.127.72

10 \* \* 7 ms 142.124.127.159

11 4 ms 4 ms 4 ms 64.230.97.145

12 \* \* \* Request timed out.

13 \* \* \* Request timed out.

14 \* \* \* Request timed out.

15 11 ms 12 ms 11 ms 52.94.83.145

16 74 ms 11 ms 11 ms 52.94.83.200

17 11 ms 11 ms 11 ms 52.94.82.251

18 13 ms 13 ms 13 ms 52.94.81.40

19 \* \* \* Request timed out.

20 \* \* \* Request timed out.

21 \* \* \* Request timed out.

22 \* \* \* Request timed out.

23 \* \* \* Request timed out.

24 \* \* \* Request timed out.

25 11 ms 10 ms 10 ms ec2-15-222-200-155.ca-central-1.compute.amazonaws.com [15.222.200.155]

Trace complete.  
  
  
Now, this question is a little tricky… Why do you think these mohawkcollege.ca addresses may be in different ranges and could have vastly different routes?  
  
Because my route is determined by my first ping, so it can vary depending on traffic and who is offering the best route at the time. The route is dynamic, not static.

1. Let’s investigate addresses a bit more!  
     
   First, let’s get the IP address for www.cibc.com by using ping (ie. in a command window, try the command “ping www.cibc.com”).

23.53.4.107

Next, let’s get the IP address for the city of Hamilton website by using a different tool called nslookup. In a command window, try the command “nslookup www.hamilton.ca”. You’re going to notice that there are several different ones listed!

Addresses: 2606:4700:10::ac43:b39

2606:4700:10::6816:2eba

2606:4700:10::6816:2fba

172.67.11.57

104.22.46.186

104.22.47.186.  
  
I think there are multiple servers listed for redundancy and downtime prtoection  
  
when I ping I get 104.22.46.186  
  
So it uses the last IP listed on the nslookup. In this case the last 2

Finally, let’s turn that last task around and see if we can’t figure out what company is using a specific IP address!  
  
In a command window, try the command “nslookup 72.14.180.202”.

C:\Users\000970878>nslookup 72.14.180.202

Server: acfe-064-dc012.acad.mc.local

Address: 142.222.64.12

Name: www-loadbal1.linode.com

Address: 72.14.180.202  
  
Linode.com is my best guess

Congratulations! You’ve successfully made it to the end of this lab! Yay!

Now, save this file and upload it to the submission link on the course page. If you’re not sure how to do that, please ask! That’s what the Professor is for!