Lab 6: Key Network Monitoring Tools

Understanding more about network monitoring tools ping , traceroute, and DNS

The objective of this postlab is to gain a more accurate understanding of what ping, traceroute, and DNS actually tell you about a target host.

You should be able to perform this exercise anywhere **except from anywhere within the college network**, since Algonquin's ITS team has chosen to block ICMP packets as a security measure. If necessary, try going to Tim's, McDonald's, Starbucks, or somebody's house to use their wifi and you'll hopefully get correct results.

The main principles this exercise is trying to illustrate are:

- Yes, yes, we know: ping and traceroute indicate whether or not a target host is reachable. That's good enough for 1st semester, but we need to go deeper.
- Traceroute **ONLY** tells you about the change in TTL in the network path **TO** the target host, and **NOTHING** about the return path. (Traceroute results are from ICMP **Requests**.)
- Ping **ONLY** tells you about the change in TTL in the network path back **FROM** the target host, and **NOTHING** about the outbound path. (Ping prints the TTL of the ICMP **Reply**.)
- In the good old days, it was common for the outbound and return paths to be identical. With today's internet, there's never any certainty that the two paths will be the same.

- A single mega monster-sized data centre may host a bazillion web sites. When you do a DNS query, don't be surprised if you get multiple answers (... that all hang together and will actually all work properly).

- Just take all the above as a sign that in today's networking world, you need to start thinking BIG (like 1000's or 100,000's of computers and network connections).

Time Required

- actual typing at the keyboard: 5-10 mins
- reading, thinking, getting confused, scratching your head, getting unconfused: 60-75 mins

What you need to submit and when:

Complete this Lab exercise and quiz on BrightSpace, **before** your lab session in Week 7.

Tip: some of the quiz questions are super easy and require a little work. Please be sure to attempt to them all; if necessary, skip over ones you find difficult.

This exercise and it's associated quiz is worth **100%**, of your mark for Lab 6.
(The pre-lab and post-lab are weighted at **0%** because there are none.)

References and Resources:

- A functioning laptop, and access to a command window; MS-Win, Linux, Mac OS.
- Access to an internet connection.

Task 1: Did you know? TTL values for all known Operating Systems.

In the history of humankind, across the entire globe and now into outer space, there have only been *three* different TTL values used when launching packets onto the wire. Let's collect some facts about the TTL values when packets are generated by a device and sent out.

OS	**TTL**
Linux, Mac OS, unix, Android, Apple IOS, + just about every other web-enabled gadget	64
MS-Win	128
Real **hardware** / networking equipment (Cisco, Nokia, HP, etc)	255

Step 1. Test out Linux: go ahead and ping your home router (e.g. try 192.168.0.1 or 192.168.1.1 or 192.168.2.1). It's almost certainly running a version of embedded Linux.
Step 2. Test out MS-Win: ping a real or virtual machine. Surely there's more than one laptop where you live? (Tip: anything you need to turn off, in order to get a response?)
Step 3. Test out Cisco hardware: ping a router or switch during a lab session (or view your saved results from previous labs), or use Packet Tracer (PC to a router or switch).

Task 2: Make Some Easy Cash

Currently, there's nowhere on (or off) the planet that's more than ~30 hops away. Honestly, really, truly. Here's the offer and challenge:

- I'll give $25 cash to the first student who emails me with info *correctly* identifying a site that's **more than 32 hops away**. The proof should be the complete traceroute output from your computer, with ***less than*** **20 unresolved hops** (marked "* * *" in the output). Those unresolved hops are often firewalls, load balancers, IPv4-IPv6 conversion, etc.
- Tips: (1) I've offered this challenge & reward in the past but so far no one has ever claimed it. The best prospects seem to be gaming sites in various places in Asia. Give it a shot and see what you can find. (2) The site "bad.horse" and others similar to it are engineered using tools such as CounterTrace and do not qualify. Please see last page for an analysis.

Now here's the *power* of that information.
A. Add a column to the chart, and do some easy math, assuming the furthest site that exists is exactly 30 hops away.

OS	**Starting TTL**	**Lowest Possible TTL** (assuming at most 30 hops)
Linux (and Mac OS, unix, etc)	64	??
MS-Win	128	??
Real (& expensive) **hardware**	255	??

B. Check it out:

- Could the TTL on a MS-Win packet ever drop low enough to get within the range of a Linux TTL packet? **No**, the chart above says the lowest (on planet earth) will always be well above the Linux range.
- What about the TTL of a real piece of networking hardware? Same story, it can't / won't get low enough to be confused for a MS-Win packet, and definitely never confused for a Linux packet.

Task 3: **What OS** is running FAANG, Algonquin College, and other sites?

FAANG: the biggest hi-tech companies and also the companies that essentially control our experience of the internet. Google it and you'll find what FAANG stands for.

Although we don't really care how far away these companies are, a secondary result of ping tells us what OS the server is running (... based on the knowledge from the previous task!!)

Step 1. Ping each of the sites listed below, **record the results**, and complete the chart. Tip: one of them won't work; they've got some sort of network device preventing pings.

Company	**Received TTL**	**Likely Starting TTL**	**Probable Server OS**
Facebook.com			
Apple.com			
Amazon.ca (e-comm)			
Netflix.ca			
Google.ca			
Algonquincollege.com			
Cisco.com			
Nokia.com			

Step 2. Based on the above chart, assuming you wanted to work for a big internet company, what is the single most import OS you should become an expert in? i.e. the one that runs the most servers and networks?

Step 3. Challenge: One of the sites above didn't respond to pings. Can you start wireshark, browse to the website, stop the capture immediately, and then check the IP TTL values in the (encrypted TLS) packets for the web page?!! Maybe check with your classmates for ideas and tips?

Task 4: **Who** Is Running the website?

We're building a thread which runs through these tasks and we need a couple more steps in order to make sense of it. Let's examine the Algonquin college website more carefully. We'd like to know *who* is running it: the college itself, or is it outsourced to a hosting company? This is a job for DNS and reverse-DNS.

Step 1. From a command window, issue the CLI command to do a DNS query on Algonquin's domain name (DNS = I have a name and I want the corresponding IP)
MS-Win: **nslookup algonquincollege.com**
Linux / Mac: **host algonquincollege.com**

Step 2. Now we have the IP for Algonquin's website. Ok, we need a reverse-DNS query to tell us *who* is running it (*reverse DNS* = I have an IP, I want a name). **Record the result**:
MS-Win: **nslookup {IP address from step 1}**
Linux / Mac: **host {IP address from step 1}**

Step 3. Examine the answer carefully (check the part immediately prior to the .com ending), and you'll find who hosts the Algonquin site. Is it a big company or small company?

An objective of this exercise is to make sure you **understand the results** of the DNS + reverse-

DNS steps.

1. A hosting company usually hosts *many* websites.
2. So if you ask DNS to resolve the location (IP address) of where all those web sites point, they all point to the same ISP (i.e. to one or a small set of IP addresses).
3. But! There can only be one "owner" of an IP so with the given IP address, we can ask the (reverse-) DNS question of who that is.

Task 5: Contrasting Outbound (traceroute) and Return (ping) paths

We've collected some good info about AlgonquinCollege.com: (a) TTL from ping, (b) the IP address where it's hosted, and (c) what ISP is hosting the site. Now it's time to compare the paths **to** and **from** the college site.

Step 1. Re-Read the cover page summary about the difference between what is printed by the ping and the traceroute commands. Hold that solidly in mind in the following steps.
Step 2. Review your answer for the TTL value when pinging the college site. Do the math: How many hops are you **FROM** the college? **Record** your answer.
Step 3. Do a traceroute **TO** algonquincollege.com and **copy** the output. Be patient, it make take a few minutes for the traceroute to complete. Note that you can try with and without resolving the names of the intermediate hops, by specifying the "-d" option for tracert.
Step 4. Do traceroute and ping give the same result for the number of hops **TO** and **FROM** the site? If the difference is larger than 1 (why 1?), we need to understand the reason.
Step 5. **Record** your conclusion: (a) the two are the same (+/- 1), (b) they're different by a handful; (c) they're different by a dozen or two; or (d) they're different by a 100 or more.

Imagine you're a big company and lots of people want to: (a) hack your site to steal info and/or (b) disable your site to prevent you from doing business, and possibly (c) blackmail you to pay them money in order not to continue attacking your site.

Surely you would want to intercept all traffic coming **TO** your site by protecting it with:

- load-balancers to distribute the 100's (nope! Remember - think 1000's or 100,000's) of requests you receive amongst many different servers
- firewalls, to block nasty packets
- traffic shaping devices (i.e. putting limits on the bandwidth). For example, so that important traffic (to your e-commerce site) will get a high(er)-priority over other traffic (to ordinary sites that you host for other companies).
- etc, etc (i.e. stuff that we don't have time to cover now)

Surely, some or all of these devices result in additional hops along the **incoming** path. Would this be (a) a handful of hops (extra steps); (b) a dozen or two extra hops; or (c) a 100 or more extra hops?

For responses coming from the servers, we truly hope that the servers aren't generating traffic that's harmful to themselves. There shouldn't need to be as much (or any) protection when sending back the reply **TO** the client devices. With less protection required, there shouldn't be as many added hops, so the return path will likely be shorter.

Task 6: Think BIG networks

The theme of thinking bigger is important throughout this course. Consider most sites: people request a web page, spend a few (or many) minutes reading it, and then possibly repeat this pattern. A streaming company like Netflix has very different network needs: a customer visits the site, chooses a movie, and then spends multiple *hours* continuously streaming content (IP packets). That's the reason Netflix is still responsible for a crazy large percentage of all network traffic in the evening hours. How on earth do they organize their networks and servers to handle that traffic? Let's get some insight into that with the help of DNS.

Step 1. From a CLI window, do a DNS query of netflix**.ca** and **record** your output. (See Task 4, Step 1 for an example of the syntax.)

Step 2. Whoa! Do you see what I see? Can you recognize and distinguish IPv6 address(es) from IPv4 address(es)? Take a quick mental snapshot of the results.

Step 3. Hopefully a minute or two has passed. Rerun the DNS query and **record** the output again.

Step 4. At least twice more, wait for at least a minute or two (grab a coffee, pop some microwave popcorn, whatever), then rerun the DNS query and **record** the output.

Step 5. Now compare the results of all the output, focusing for now on the IPv4 addresses. How many addresses are there? Are they always the exact same addresses? Are they always in the exact same order?

Step 6. (Optional) If you're interested, you can compare the view we have in Canada of the US version of Netflix: do a DNS query of netflix**.com** and see what you get.

Step 7. (Optional) If you're still curious, I used wireshark to capture the connection sequence to Netflix.com and I found out that the domain secure10.sync.com is part of their network setup. Repeat steps 1-2 for this new domain. You'll find that these IPs are ping-able, and that will tell you what kind of setup Netflix is using (as in Tasks 1-3).

Imagine that you need to server 1000's or 100,000's of customers so you decide to build an army of robot servants. Each robot acts as a clone of the others: each and every one is individually capable of handling any request. In this scenario, when a client comes with a request, does it actually matter which of the many robots responds? (Remember, they're all set up and behave identically!)

Replace the word "robot servant" with "server", and then you have Netflix's network & server setup. They've organized their DNS so that each client gets a randomized answer to their name query of netflix.com. The client machine chooses the first IP in the list they receive and goes to that server via whatever network path is appropriate to reach it.

So maybe, when we expand the network for 1000's and 100,000's, we kinda bend the rules about IP addresses being globally unique?! (This isn't exactly "anycast" but it has a similar flavour.)

Task 7: Revisit your Ping and Traceroute results from Post-Lab 2

We used a ring topology where non-local traffic *always* flowed clockwise. Did you understand the pattern and the reason for the TTL values and the # of hops from traceroute? Go back and re-visit (or re-do) that exercise. Knowing what ping and traceroute really measure, could you accurately predict the values if there was a question on a test or exam about it? If not, go over the last two tasks until you do.

(Analysis of "bad.horse" follows on the next pages)

Building sites like "bad.horse" using CounterTrace

For fun, go ahead and traceroute to "bad.horse". You'll want to allow ~50 hops for the complete path. The question becomes: What's going on? Continue reading for some clues and analysis.

1. You'll notice that the first "bad.horse" hop has IP address 162.252.205.130. Then every subsequent hop is +1 on the address, all the way up to .157. ... Something funny about that since it's difficult to have links with only a single address (i.e. a subnet with two connections and yet only a single address).

2. Next try pinging any one of those addresses [130-157]. You'll notice the TTL is in the 50's (53 in my case; Bell is my ISP). That means the return path is ~11 hops. Always. From every one of those addresses. That's not possible if every one of those addresses is truly a separate hop.

3. If we sniff around nearby addresses, we find some domain names that are revealing:
nslookup on: .129 = igw-badhorse.toroc1b.on.ca.sn11.net
.128 = network-badhorse.toroc1b.on.ca.sn11.net
.159 = broadcast-badhorse.toroc1b.on.ca.sn11.net
Based on the names, it looks like internally 162.252.205.128/27 is allocated for bad.horse.

4. It'd be nice to check out the web page at bad.horse but unfortunately the SSL certificate has expired, so you just get a warning: "Error code: SEC_ERROR_EXPIRED_CERTIFICATE". Examining the certificate we see that the expiry is 28 Aug 2021; you could get around this problem by manually setting your computer's clock to a prior time, but there's an easier way.

5. Using wget with the option to ignore the certificate (--no-check-certificate) we can download the HTML for the web page. It's a compact 478 bytes. Examining the file, we find a video: src="//www.youtube-nocookie.com/embed/F7GDaLijr1w?html5=1&autoplay=1&rel=0" or more conveniently: https://www.youtube.com/watch?v=F7GDaLijr1w It's the song with all the words!

6. But all that still doesn't tell us what's going on with the hop count. Find out who owns those addresses by going to ARIN: https://www.arin.net/resources/registry/whois/ In the search bar (top right side), type in the address 162.252.205.157. We find out that Sandwich.Net, LLC owns the block 162.252.204.0/22. If you continue looking, you can find the sysadmin's contact info.

7. You then google the sysadmin and find out he lives in good 'ol Canada. So you email him from your official Algonquin email account, using your credentials as a Professor of networking, and thanking him for such an interesting site. He replies and includes the info that he used the CounterTrace utility to spoof multiple hops from a single site: www.softpile.com/countertrace/

8. Mystery is now solved! The return TTL from pings indicates the true distance, and handy-dandy Linux can do almost anything when suitably programmed and configured. The internet is still *safe* and nothing on planet earth is more than ~30 hops away. Hey, this was kinda fun!

If you're still curious about Bad Horse, you can get more info and watch the full video:
https://en.wikipedia.org/wiki/Dr._Horrible%27s_Sing-Along_Blog
https://www.youtube.com/watch?v=Of9kHpCv1ts (full 3-part movie)