Internet Routes and Measure of Round Trip Times

Lab 4

50.005 Computer System Engineering

Due: 01 Apr 08:30 AM (Week 10)

Natalie Agus
INFORMATION SYSTEMS TECHNOLOGY AND DESIGN

Overview

In this lab exercise, you will learn how to use `ping` and `traceroute` to measure round trip times and find network routes.

Learning objectives

At the end of this lab exercise, you should be able to:

- Understand how the `ping` and `traceroute` utilities work.
- Use the `ping` utility to measure network round trip times.
- Use the `traceroute` utility to find network routes.
- Observe and understand the effects of varying packet sizes on delays experienced.

Preparation

You will need `ping` and `traceroute` to be installed on your OS. Most **Ubuntu** / **MacOS** installations should already include `ping` by default. You can install `traceroute` by running "`sudo apt-get install traceroute`" from the command line.

Submission

- **The total points for this lab is 35**
- Export this handout and fill in your answers in the blanks denoted in **blue**
- Export as pdf and **ZIP** it (not rar, or any other compression algorithm)
- **Upload** to `@csesubmitbot` telegram bot using the command `/submitlab4`
- **CHECK** your submission by using the command `/checksubmission`

Part 1: Measurement of round trip times using ping

The `ping` utility is one of the most widely-used network utilities. It enables you to measure the time that it takes for a packet to travel through the Internet to a remote host and back.

The `ping` utility works by sending a short message, known as an *echo-request*, to a remote host using the Internet Control Message Protocol (ICMP). When a host that supports ICMP receives an echo-request message, it replies by sending an echo-response message back to the originating host.

In the first part of this lab exercise, you will use the `ping` utility to send echo requests to a number of different hosts. In many of the exercises, you will be referring to hosts using their DNS names rather than their IP addresses. For more information about `ping`, you can look up its manual page by running "`man ping`" from the command line.

Round trip times

Use `ping` to send 10 packets to each of the following hosts. Each packet should have a size of 56 bytes, and there should be an interval of 5 seconds between each packet sent.

```
www.csail.mit.edu

www.berkeley.edu

www.usyd.edu.au

www.kyoto-u.ac.jp
```

Note: The size of each packet is 56 bytes by default, but you may observe that the actual size of the packet is larger than 56 bytes. You can look up the manual for `ping` to understand why such a discrepancy exists.

Question 1 [4pt] : For each host, record the percentage of packets sent that resulted in a successful response. Record also the minimum, average, and maximum round trip times for the packets that resulted in a response.

Your answer:

Website	Successful Percentage %	Min RTT (ms)	Average RTT (ms)	Max RTT (ms)
www.csail.mit.edu	100%	4.806	5.490	7.484
www.berkeley.edu	100%	198.564	199.539	200.368
www.usyd.edu.au	100%	217.778	218.441	219.508
www.kyoto-u.ac.jp	100%	43.841	44.84	45.454

Question 2 [4pt]: Describe and explain the differences in the minimum round trip time to each of these hosts.

Your answer:

Min round trip time is influenced by a number of factors such as distance, transmission medium, number of network hops and traffic levels. So for servers that are located far away from me, the propagation delay is going to be longer causing RTT to be longer.

MIT has the shortest average RTT, which means that there is likely a server very close to Singapore, for example in SUTD since we used to collaborate with MIT, so the number of network hops will be lesser resulting in shorter RTT.

Compared to Berkeley and University of Sydney, they likely have their servers hosting the website further away from me so more network hops were needed.

Question 3 [4pt]: Repeat the exercise using packet sizes of 56, 512 and 1024 bytes. Record the minimum, average, and maximum round trip times for each of the packet sizes. Why are the minimum round-trip times to the same hosts different when using 56, 512, and 1024–byte packets?

Your answer:

Packets that contain more bytes seem to take longer to transmit according to my data. This increase in round-trip time might be due to transmission delay, which is packet length (bit) divided by link bandwidth (bit/s). So the larger the packet length, the longer the transmission delay, given the same link bandwidth.

Website	Packet Size	Successful Percentage %	Min RTT (ms)	Average RTT (ms)	Max RTT (ms)
www.csail.mit.edu	56	100%	4.94	5.512	6.96
	512	100%	10.05	13.32	16.687
	1024	100%	12.504	15.24	25.765
www.berkeley.edu	56	100%	209.367	239.635	302.455
	512	100%	207.447	252.146	312.898
	1024	100%	209.281	266.332	318.267
www.usyd.edu.au	56	100%	220.484	273.659	330.759
	512	100%	224.317	297.175	431.365
	1024	90%	227.42	336.851	530.383
www.kyoto-u.ac.jp	56	100%	44.742	45.012	45.891
	512	100%	45.913	49.019	65.538
	1024	100%	46.665	50.747	56.968

Unanswered pings

Use `ping` to send 100 packets to the following host. Each packet should have a size of 56 bytes, and there should be an interval of 5 seconds between each packet sent.

```
www.wits.ac.za
```

Question 4 [8pt]: Record the **percentage** of the packets sent that resulted in a **successful response** for each host. What are some possible **reasons** why you may not have received a response? (Be sure to check the host in a web browser).

Your answer:

0%. All packets sent are unsuccessful. A few possible reasons include:

1. The host's system might have crashed and is unavailable.
2. The network connection between the host and my system might have failed
3. The host blocks all ICMP Echo requests
4. The network might be dropping packets due to congestion

The most possible system is that the Wits University uses a firewall to block all ICMP Echo requests to evade network mapping applications used by hackers. This will prevent attacks such as network discovery attacks like ping sweep. The reason why we still can browse its website is because they still allow HTTP requests to pass.

Part 2: Understanding Internet routes using traceroute

The `traceroute` utility is another useful network utility. It enables you to trace the route taken by a packet from your machine to a remote host.

Here is an example of the output produced when `traceroute` is used to trace the route taken by a packet to `www.mit.edu`.

```
traceroute to www.mit.edu (118.215.81.86), 30 hops max, 60 byte pack-
ets

 1  192.168.9.2 (192.168.9.2)  0.221 ms  0.193 ms  0.107 ms
 2  10.12.0.1 (10.12.0.1)  3.363 ms  2.555 ms  3.253 ms
 3  172.16.1.106 (172.16.1.106)  3.072 ms  3.416 ms  3.418 ms
 4  172.16.1.210 (172.16.1.210)  4.977 ms  4.712 ms  4.921 ms
 5  192.168.22.27 (192.168.22.27)  4.806 ms  6.521 ms  6.451 ms
 6  103.24.77.1 (103.24.77.1)  7.172 ms  3.590 ms  3.187 ms
 7  201.210-193-8.qala.com.sg (210.193.8.201)  4.312 ms  9.056 ms 7.870 ms
 8  137.203-211-158.unknown.qala.com.sg (203.211.158.137) 8.904 ms 6.690 ms  6.555 ms
 9  213.203-211-158.unknown.qala.com.sg (203.211.158.213) 7.710 ms 5.423 ms  5.193 ms
10  203.116.10.125 (203.116.10.125)  6.783 ms  6.705 ms  6.440 ms
```

Each line in the output begins with a host on the route from your computer to `www.mit.edu`, followed by the round-trip times for 3 packets sent to that host. For more information about `traceroute`, you can look up its manual page by running "`man traceroute`" from the command line.

Question 5 [5pt]: Explain how `traceroute` discovers a path to a remote host. (*Hint:* The `traceroute` manual will be helpful for answering this question.)

Your answer:

"Traceroute utilises the IP protocol 'time to live' field and attempts to elicit an ICMP TIME_EXCEEDED response from each gateway along the path to some host."

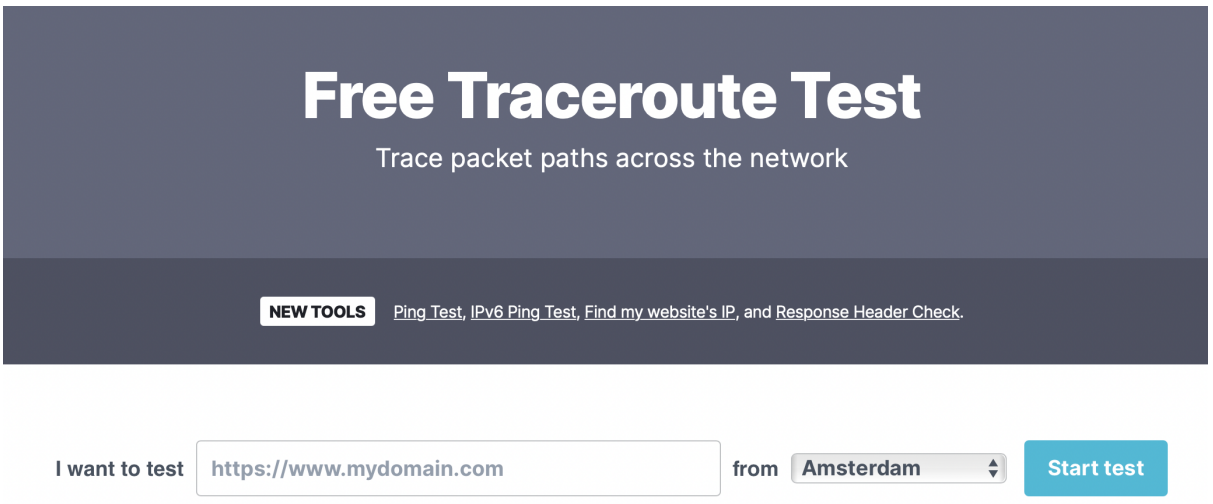
Firstly, traceroute will set the TTL for a packet to 1, sending it towards the requested destination host and listening for the reply. When it receives the time exceeded responses, the packet will be examined to determine where it came from, thus identifying the machine that is one hop away. So traceroute will generate a new packet with TTL 2 to determine the machine that is 2 hops away, and so on.

Route asymmetries

In this exercise, you will run traceroute in two opposite directions. First, you will run traceroute on a remote host to see the route taken to your network. You will also run traceroute from your computer to see the route taken to that host.

Step 1: Find out your computer's public IP address. (Hint: You can use a website like http://www.whatismypublicip.com/, or search for "what is my ip" using Google's search engine.)

Step 2: Visit https://www.uptrends.com/tools/traceroute in your web browser. Enter your computer's public IP address, select the "from Location" and click "Start Test" to start a traceroute to your computer. Follow the steps shown below for at least three locations namely: New York, Amsterdam, Tokyo.



Step 3: After `traceroute` finishes running, you should be able to view the route taken from `specified location` to your network. Record the IP address of the first hop, which will be used in the next step.

Hop	Time	Time	Time	Host name	IP address
1	<1	<1	<1	gateway.as64425.com	5.182.210.1
2	-	-	-		
3	1	1	1		212.119.24.97
4	2	1	1	ae-10.r25.amstnl02.nl.bb.gin.ntt.net	129.250.2.90
5	12	12	12	ae-6.r20.parsfr04.fr.bb.gin.ntt.net	129.250.4.138
6	14	15	17	ae-2.r21.parsfr04.fr.bb.gin.ntt.net	129.250.3.46
7	90	87	86	ae-13.r24.asbnva02.us.bb.gin.ntt.net	129.250.6.6
8	149	149	149	ae-2.r24.snjsca04.us.bb.gin.ntt.net	129.250.6.237
9	-	-	-		
10	149	149	149	ae-2.r00.mlpsca01.us.bb.gin.ntt.net	129.250.4.101
11	148	148	148		129.250.24.196
12	155	155	155		129.250.130.254
13	148	148	148		198.107.143.162
14	-	-	-		

Step 4: On your computer, run `traceroute` using the IP address recorded in the previous step as the remote destination.

`$ traceroute <ip address from step 3>`

Question 6 [5pt]: Record the output of `traceroute` when run in both directions above. **Paste it as screenshots at the end of this document.**

Question 7 [5pt]: Describe anything unusual you might observe about the output. Are the same routers traversed in both directions? If no, why might this be the case?

Your answer:

Not able to reach Amsterdam or New York.

From Tokyo to me: 12 hops
From me to Tokyo: 17 hops

The routers are not the same for both directions. This is because the internet is a large and complex aggregation of network hardware and there are many gateways, many routes to reach any destination.

Tokyo

```
❯ traceroute 31.204.145.131
traceroute to 31.204.145.131 (31.204.145.131), 64 hops max, 52 byte packets
 1  singtel-acplus (192.168.1.254)  8.989 ms  2.622 ms  2.391 ms
 2  bb42-60-51-254.singnet.com.sg (42.60.51.254)  43.573 ms  4.447 ms  6.217 ms
 3  202.166.123.130 (202.166.123.130)  3.874 ms  4.111 ms  4.134 ms
 4  202.166.123.129 (202.166.123.129)  3.717 ms  4.663 ms  5.786 ms
 5  ae8-0.qt-cr03.singnet.com.sg (202.166.121.101)  6.026 ms  4.639 ms  5.058 ms
 6  ae13-0.tp-cr03.singnet.com.sg (202.166.120.109)  5.123 ms  9.119 ms  4.310 ms
 7  ae4-0.tp-er03.singnet.com.sg (202.166.123.70)  4.667 ms  3.799 ms  5.205 ms
 8  203.208.191.113 (203.208.191.113)  4.136 ms
    203.208.145.233 (203.208.145.233)  4.115 ms
    203.208.191.197 (203.208.191.197)  8.436 ms
 9  203.208.173.166 (203.208.173.166)  170.856 ms
    203.208.182.249 (203.208.182.249)  5.223 ms  5.727 ms
10  203.208.172.233 (203.208.172.233)  176.850 ms  182.075 ms
    203.208.158.46 (203.208.158.46)  186.499 ms
11  ix-xe-0-1-2-0.tcore2.pdi-paloalto.as6453.net (66.198.144.41)  319.620 ms
    203.208.158.178 (203.208.158.178)  180.623 ms  177.814 ms
12  203.208.172.234 (203.208.172.234)  181.777 ms
    if-ae-2-2.tcore1.pdi-paloalto.as6453.net (66.198.127.1)  286.610 ms  331.660 ms
13  ix-xe-0-1-2-0.tcore2.pdi-paloalto.as6453.net (66.198.144.41)  281.048 ms
    203.208.172.233 (203.208.172.233)  180.377 ms  177.590 ms
14  ix-xe-0-1-2-0.tcore2.pdi-paloalto.as6453.net (66.198.144.41)  381.765 ms
    if-et-5-3.hcore1.kv8-chiba.as6453.net (209.58.86.69)  305.434 ms
    if-ae-0-2.tcore1.sv1-santaclara.as6453.net (63.243.251.1)  319.869 ms
15  * if-ae-2-2.tcore1.pdi-paloalto.as6453.net (66.198.127.1)  286.074 ms
    if-ae-7-2.tcore2.sv1-santaclara.as6453.net (209.58.86.73)  304.881 ms
16  if-ae-7-2.tcore2.sv1-santaclara.as6453.net (209.58.86.73)  290.907 ms
    if-ae-24-2.tcore2.tv2-tokyo.as6453.net (180.87.181.72)  307.711 ms  297.582 ms
17  hosted-by.i3d.net (31.204.145.131)  308.881 ms
    if-et-1-2.hcore2.kv8-chiba.as6453.net (120.29.211.3)  291.326 ms  333.709 ms
```

Hop	Time	Time	Time	Host name	IP address
1	5	5	2		31.204.145.131
2	12	4	14		109.200.218.0
3	44	44	44		109.200.218.221
4	48	58	49		123.255.90.77
5	79	94	80		203.208.171.230
6	177	177	177		203.208.182.253
7	258	275	279		203.208.175.34
8	258	275	275		202.166.120.222
9	263	279	279		202.166.121.102
10	267	283	283		202.166.123.130
11	*	*	*	Request timed out	
12	272	285	285		42.60.48.151

Amsterdam

```
❯ traceroute 5.182.210.1
traceroute to 5.182.210.1 (5.182.210.1), 64 hops max, 52 byte packets
 1  singtel-acplus (192.168.1.254)  8.620 ms  6.212 ms  3.465 ms
 2  bb42-60-51-254.singnet.com.sg (42.60.51.254)  5.068 ms  6.356 ms  5.267 ms
 3  202.166.123.130 (202.166.123.130)  9.023 ms  6.065 ms  9.924 ms
 4  202.166.123.129 (202.166.123.129)  12.665 ms  7.616 ms  11.425 ms
 5  ae8-0.qt-cr03.singnet.com.sg (202.166.121.101)  10.938 ms  8.255 ms  5.226 ms
 6  ae13-0.tp-cr03.singnet.com.sg (202.166.120.109)  8.034 ms  5.958 ms  4.824 ms
 7  ae4-0.tp-er03.singnet.com.sg (202.166.123.70)  6.100 ms  6.804 ms  5.104 ms
 8  203.208.191.113 (203.208.191.113)  4.955 ms
    203.208.145.233 (203.208.145.233)  6.661 ms
    203.208.191.113 (203.208.191.113)  5.524 ms
 9  203.208.166.202 (203.208.166.202)  159.925 ms  160.432 ms  160.283 ms
10  linx-224.retn.net (195.66.224.193)  237.538 ms  206.652 ms  203.370 ms
11  ae0-2.rt.ir9.ams.nl.retn.net (87.245.232.123)  198.681 ms  197.640 ms  198.664 ms
12  * * *
13  * * *
14  * * *
15  * * *
16  * * *
17  * * *
18  * * *
19  * * *
20  * * *
```

Hop	Time	Time	Time	Host name	IP address
1	<1	<1	<1		5.182.210.1
2	1	1	1		87.245.246.38
3	185	186	186		87.245.232.11
4	*	*	*	Request timed out	
5	195	195	196		203.208.171.23
6	192	192	192		203.208.182.25
7	192	194	192		203.208.192.30
8	192	192	192		202.166.120.22
9	192	192	192		202.166.121.10
10	193	199	193		202.166.123.13
11	*	*	*	Request timed out	
12	194	194	194		42.60.48.151

New York

```
❯ traceroute 72.9.99.137
traceroute to 72.9.99.137 (72.9.99.137), 64 hops max, 52 byte packets
 1  singtel-acplus (192.168.1.254)  3.740 ms  4.238 ms  2.007 ms
 2  bb42-60-51-254.singnet.com.sg (42.60.51.254)  4.659 ms  21.550 ms  4.350 ms
 3  202.166.123.130 (202.166.123.130)  4.586 ms  7.814 ms  4.194 ms
 4  202.166.123.129 (202.166.123.129)  5.028 ms  4.417 ms  3.850 ms
 5  ae8-0.qt-cr03.singnet.com.sg (202.166.121.101)  10.113 ms  6.263 ms  4.717 ms
 6  ae13-0.tp-cr03.singnet.com.sg (202.166.120.109)  5.057 ms  4.528 ms  5.259 ms
 7  ae4-0.tp-er03.singnet.com.sg (202.166.123.70)  4.806 ms  5.342 ms  5.949 ms
 8  203.208.191.113 (203.208.191.113)  5.273 ms  4.874 ms  4.257 ms
 9  203.208.178.206 (203.208.178.206)  186.829 ms
    203.208.182.249 (203.208.182.249)  4.851 ms  9.327 ms
10  203.208.183.133 (203.208.183.133)  6.192 ms
    ae5.mpr1.pao1.us.zip.zayo.com (64.125.35.189)  185.843 ms
    203.208.183.133 (203.208.183.133)  5.712 ms
11  ae9.cs1.sjc2.us.eth.zayo.com (64.125.27.188)  188.100 ms
    203.208.172.234 (203.208.172.234)  211.372 ms  184.277 ms
12  ae5.mpr1.pao1.us.zip.zayo.com (64.125.35.189)  184.678 ms
    ae27.cr1.sjc2.us.zip.zayo.com (64.125.30.231)  197.302 ms  187.313 ms
13  ae5.mpr1.pao1.us.zip.zayo.com (64.125.35.189)  176.655 ms
    ae16.mpr3.sjc7.us.zip.zayo.com (64.125.31.13)  189.444 ms
    ae5.mpr1.pao1.us.zip.zayo.com (64.125.35.189)  176.640 ms
14  zayo-level3.sjc7.us.zip.zayo.com (64.125.13.242)  172.115 ms
    ae27.cr1.sjc2.us.zip.zayo.com (64.125.30.231)  189.071 ms
    zayo-level3.sjc7.us.zip.zayo.com (64.125.13.242)  187.554 ms
15  ae27.cr1.sjc2.us.zip.zayo.com (64.125.30.231)  185.749 ms *  193.979 ms
16  ae16.mpr3.sjc7.us.zip.zayo.com (64.125.31.13)  185.194 ms  205.979 ms
    core-techno.ear3.newark1.level3.net (4.16.89.78)  259.357 ms
17  * * *
18  * core-techno.ear3.newark1.level3.net (4.16.89.78)  270.184 ms *
19  * core-techno.ear3.newark1.level3.net (4.16.89.78)  316.286 ms *
20  * * *
21  * * *
22  * * *
23  * * *
24  * * *
25  * * *
```

Hop	Time	Time	Time	Host name	IP address
1	1	<1	<1		72.9.99.137
2	1	<1	<1		72.9.111.131
3	*	*	*	Request timed out	
4	1	2923	<1		204.145.67.65
5	*	*	*	Request timed out	
6	2923	2999	1		205.251.126.90
7	*	*	*	Request timed out	
8	*	*	*	Request timed out	
9	*	*	*	Request timed out	
10	*	*	*	Request timed out	
11	*	*	*	Request timed out	
12	*	*	*	Request timed out	
13	*	*	*	Request timed out	
14	*	*	*	Request timed out	
15	256	254	254		202.166.123.130
16	*	*	*	Request timed out	
17	253	252	252		42.60.48.151