COMP90007 Internet Technologies

Project 1: Network Analysis

Semester 2, 2024

Due Date: September 13, 2024, Friday 5pm

1 Introduction

Weighting: 15% of the total marks for the subject. The weighting of each question is shown beside the question.

Tasks: The project focuses on evaluating network performance by measuring hop count, bandwidth, delay, and jitter. The key output of this project is a report answering the questions in Sections 2-4, which has to follow a certain format (refer to Section 5 for specific requirements). It is recommended that you perform these tasks in a consistent networking environment to reduce the variance in your report.

Each Section and the relevant questions in them must be answered separately such that the relevant information needed to answer the question is present in that question alone. Please ensure that each answer is clearly marked with the relevant question number and **do not combine** answers for multiple questions. All the plots and charts (e.g., scatter plots, bar graphs, etc.) needed to answer questions should be placed in the main body of your report where you will explain the observations being derived. As evidence of your work, when you run the following commands, you will need to **take screenshots of the raw results** obtained and place them in the Appendix of your report. Reports failing to do so will be penalised.

Important Notes:

- All work presented must be your original **individual** effort.
- All commands executed must be run manually on the terminal/command prompt. No automation scripts or processes to run these commands would be accepted.
- Late submissions will attract a penalty of 10% per day (or part of a day).

2 Measuring the hop count (3 marks)

In this section, we will be observing the number of intermediate hosts in the route taken to communicate with a remote server and its relation to the physical geographical distance.

To count the number of hops taken to reach a destination host, the command `tracert` will be used (or its corresponding equivalent, depending on your operating system). This utility should be pre-installed on your operating system.

The utility can be invoked by launching a command line terminal and typing in the command. An example output of the `traceroute` command (on OS X) and `tracert` command (on Windows) is as follows:

```
$ traceroute -nw 1 cis.unimelb.edu.au
traceroute to cis.unimelb.edu.au (128.250.37.164), 64 hops max, 52
     byte packets
1  10.0.0.254  533.676 ms  1.063 ms  0.940 ms
2  58.96.2.205  27.872 ms  28.137 ms  28.293 ms
3  58.96.2.129  28.647 ms  28.577 ms  28.085 ms
```

```
4  218.100.78.33  28.299 ms  28.469 ms  28.332 ms
5  202.158.200.9  29.626 ms  28.871 ms  29.841 ms
6  202.158.210.26  31.320 ms  28.722 ms  29.135 ms
7  202.158.200.250  29.668 ms  29.096 ms  28.660 ms
8  * * *
9  * * *
10  * * *
11  128.250.37.130  957.521 ms  33.475 ms  29.891 ms
12  128.250.37.164  29.940 ms  29.260 ms  30.020 ms
```

In this section of the project, you are interested in the number of hops it takes to reach the destination server. In the example above, the number of hops to reach cis.unimelb.edu.au is 12.

Based on the number of measurements you will be taking, there are some useful command line parameters you may wish to take advantage of, to speed up the time it takes to gather results. The help documentation for the `traceroute` utility can be accessed by running `man traceroute` on OS X or `tracert /?` on Windows.

Specific task description

Please include all raw measurements in the Appendix.

2.1 Use Wireshark to capture the network trace generated when running the `traceroute` command, only sending 1 packet instead of the default 3 packets, on any one of the hosts mentioned in Table 1. Explain each step of traceroute's working to reach your chosen host by using the packets captured as well as the flow graph it generates. For the Wireshark capture, create a capture filter that catches only the traceroute request and the response received without any other packets (like dns, mdns, etc.) and explain if this filter will capture other requests (non-traceroute related) as well or not with examples.

2.2 Determine the hop count for the following hosts given in Table 1. We ask that students find one more public iperf server other than the ones listed here to gather their results.

Table 1: List of public `iperf` hosts

Host	Location
`iperf.he.net`	USA
`bouygues.testdebit.info`	France
`speedtest.telecom.mu`	Mauritius
`rychlost.poda.cz`	Czech Republic
`spd-fisrv.hostkey.com`	Finland
`ping-90ms.online.net`	France
`ping.online.net`	France
`speedtest.masnet.ec`	Ecuador
`speedtest.tds.net`	United States

Determine the approximate geographical distance for the above hosts and plot a scatter chart of the hop count versus the approximate geographical distance from the city you are currently in. Do you observe a correlation or not? Please explain your rationale with respect to networking concepts.

Key Considerations:

- You do not need a VPN to run your tests against these hosts. If you do end up using a VPN, your geographical distance would be the distance from your device (the one you are running your tests on) to the VPN servers geo-location and then to the end iperf hosts geo-location.
- You may use any scientific computing package or spreadsheet software to do your plotting, for example, Matlab, Microsoft Excel, etc.

For finding out the physical geographical distance you may use any tool or application available online, for example, you may use a combination of: https://db-ip.com and https://www.freemaptools.com/how-far-is-it-between.htm or https://www.site24x7.com/find-website-location.html and https://www.distancecalculator.net/ or anything of your choice. However, do make sure to **document it and provide the appropriate reference** to that application/ tool/ software used.

Note: The servers listed in Table 1 are public servers and are not maintained by the University of Melbourne, hence they are likely to go down at any point in time. Based on past experience, it would be advisable to **conduct your tests on these as soon as possible rather than leave it till the end** as there is a high probability that these servers might not be available at times and this cannot be used as an excuse for a late submission as we give a number of weeks for you to do the project. If these servers stop responding for a long time for some reason then please visit the link: https://iperf.fr/iperf-servers.php and find your own servers (anything that is responding) or feel free to find any public iperf server from the internet and note this in your report. Some alternate strategies worth exploring also include changing port numbers and trying to get the iperf metrics.

3 Measuring delay and jitter (4 marks)

In this section, you will be measuring the delay and jitter of the **hosts used in Section 2**, located in different geographical locations.

We will be using the `ping` utility, to measure the round-trip delay of packets. The `ping` utility should be pre-installed on all major operating systems. The standard deviation of the round-trip delay time will be taken as the value for *jitter* for this project.

The **standard deviation** measures the variation in a set of data. It is defined as the square root of the variance and is expressed as follows:

$$\sigma = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \bar{x})^2}$$

where $\bar{x}$ is the mean of the set of data. Details of this simple statistical measure can be found in many websites if you do not remember this from high school years.

A sample output of the `ping` utility is shown below, but this output will vary depending on your operating system.

```
$ ping unimelb.edu.au
PING unimelb.edu.au (172.22.44.10): 56 data bytes
64 bytes from 172.22.44.10: icmp_seq=0 ttl=124 time=3.364 ms
64 bytes from 172.22.44.10: icmp_seq=1 ttl=124 time=3.416 ms
64 bytes from 172.22.44.10: icmp_seq=2 ttl=124 time=3.730 ms
^C
--- unimelb.edu.au ping statistics ---
3 packets transmitted, 3 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 3.364/3.503/3.730/0.162 ms
```

In the output above, various statistics, including the mean and standard deviation, were calculated for you by the utility but given your version of the Ping utility, you may or may not have exactly the same output. If you can capture approximate values with decimals as part of your Ping output then it would be good to perform your own calculations to find the mean values and their standard deviation, to confirm the result.

Specific task description

For this section, you may want to consider the user-facing implications of high delay and high jitter in networking applications, especially for applications sensitive to the affect of high delay and high jitter. Please include all raw measurements in the Appendix. For MacOS/ Linux users,

please use '-c 4' flag for ping to have 4 packets for each execution of ping. For example: ping www.google.com -c 4

3.1 Use Wireshark to capture the network trace generated when running the `ping` command on any one of the hosts mentioned in Table 1 (use 4 packets for a single ping execution). Explain each step of ping's working to reach your chosen host by using the packets captured as well as the flow graph it generates. For the Wireshark capture, create a capture filter that catches only the ping request and response without any other packets (like dns, mdns, etc) and explain if this filter will capture other requests (non-ping related) as well or not with examples.

3.2 Measure the round-trip delay for all the hosts used in Section 2 (the hosts in Table 1, plus one more public iperf server selected by you). Make **three** delay measurements (run this command 3 times - not 1 command gathering a few rows of ICMP responses) of each host and find the average round-trip delay and jitter by calculating the standard deviation manually or by using the standard deviation reported by your command output.

For all of the hosts, plot the scatter chart of average round-trip delay versus the approximate physical geographical distance to the server. Do the same with the jitter (i.e. jitter vs geo distance). Outline the results in a table and ensure the calculations are included in the Appendix of the report.

3.3 From the two plots above, do you observe any correlation between delay and distance? Any correlation between jitter and distance? Why or why not? Explain your results comparatively with reference to the network environment in which you were collecting your results (this includes metrics like your download/ upload speed, users sharing the network, load on network through other apps, etc.) and how does your networking environment influence your results obtained (examples required)?

4 Measuring the bandwidth-delay product (8 marks)

In this section, you will be measuring the bandwidth of different hosts in order to determine the bandwidth-delay product, using the results from the previous sections.

The utility that will be used to perform bandwidth measurements will be the `iperf` utility. This command line utility is available for download for all operating systems from https://iperf.fr. Alternatively you may choose to use the package manager for your relevant operating system.

There are two modes of operation in `iperf`. The server mode will host a server which will listen to incoming requests from a client. An `iperf` instance running in client mode will connect to the server, and packets will be exchanged and timed between the two hosts to calculate the bandwidth. In this project, we will be running `iperf` in client mode.

A sample output of `iperf` in client mode is shown below, noting the `-c` flag to designate operating in client mode.

```
$ iperf -c iperf.eenet.ee
------------------------------------------------------------
Client connecting to iperf.eenet.ee, TCP port 5001
TCP window size:  129 KByte (default)
------------------------------------------------------------
[  5] local 10.0.0.5 port 51878 connected with 193.40.100.7 port 5001
[ ID] Interval       Transfer     Bandwidth
[  5]  0.0-106.7 sec   128 KBytes  9.83 Kbits/sec
```

Note: Some servers in Table 1 may need to use the port 5002 or perhaps even port 80, so the command to override ports will be: "iperf3 -c speedtest.serverius.net -p 5002". Also, **some iperf servers respond to iperf2, rest to iperf3** so please try to use both iperf2 and iperf3 to verify if the server is responsive to either version. If you have response from both iperf2 and iperf3 then you can pick iperf2 amongst them.

Specific task description

Please include all raw measurements in the Appendix as usual.

4.1 What does the bandwidth-delay product tell us about the data transmission capability of networks? Collect **three** sets of measurements (run this command 3 times) measuring the bandwidth of the public `iperf` hosts in Section 2 and find the mean bandwidth for each host.

4.2 Pick one host from Table 1 that is the **furthest away** from the location where you are running your tests and run the `iperf` command on it across four time slots - Morning (between 9-10am), Afternoon (1-2pm), Evening (6-7pm) and Night (11pm-12 midnight). Do you observe changes in your captured metrics? Explain why or why not by discussing the networking environment and usage patterns of the intermediate hosts/ countries your request travels through at that time (using `traceroute`). Repeat the experiment and discussion for one more host from Table 1 which is **closest to** the location you are running your tests from.

4.3 Take the mean bandwidth and calculate the bandwidth-delay product in kilobits. You should use the mean round-trip delay time from your `ping` experiments as the delay time. Plot a bar chart for each host showing your results. You may wish to use a logarithmic scale, if appropriate.

Explain your results making a comparative analysis with reference to your networking environment in which you performed your measurements. How do your results reflect upon your actual internet link speed and how does your network environment influence your results obtained (provide examples)? Are there outliers in your data? If yes, point out the outliers and explain why they are marked as outliers in your data?

4.4 List 2 major challenges you have encountered in your experiments. For example, shared network, dynamic set of users, etc. Briefly describe these challenges and how you would address them *within* the constraints of the current experiments?

5 Project Administration

5.1 Getting help

If you have any questions, the discussion board (Ed Discussion) will be a useful resource in resolving any issues. If your concern is a personal matter, then you should email the subject coordinator.

Any announcements made about the assignment in the lectures and any answers posted by the subject coordinator or the academic staff on the discussion board will be considered as part of the project specification. In addition, please keep an eye on any Canvas announcements to any changes made to the project specification.

5.2 Report Format and Submission

The report will consist of all relevant discussion, graphs, data and answers from the experimentation conducted in this project. The raw data must be included as **screenshots** in the Appendix of the report. The diagrams for answering questions, like plots, charts, flow diagrams (if any) and so on must be placed in the main document, not the Appendix. Every diagram and/or raw measure used for a specific question must be referred to (using a consistent reference scheme, e.g. Fig. X) for us to verify the result. All plots and figures must be appropriately labelled.

Any information obtained that is not of your own work must be cited. This report will require appropriate citations and referencing style. The accepted referencing style for this report is: **IEEE**.

The report must be submitted as a **PDF file** on Canvas (submission link is on Project 1 page) and no handwritten content will be accepted. Please include your name, student ID and login user name on the top of the first page. The report is to be formatted on A4 sized paper in 10 pt text, 1.5 line spacing, single column. The report should not exceed a **maximum of 12 pages** (excluding appendix) else relevant penalties will apply. It is expected that students use the formatting scheme outlined in the **Format Guideline Document**.

All commands across these sections would need to be run manually by typing the command yourself and getting screenshots of the command as well as the output received. You should not be automating this. Output captured outside of the terminal/command prompt (like .txt, .csv, etc) will not be accepted and appropriate deductions will be made.