ELEC-E7130 Assignment 2. Basic measurements

Sebastian Sonntag Markus Peuhkuri Tran Thien Thi
Jiang Weixuan César Iván Olvera Espinosa Yu Fu

Prerequisites

1. To complete this assignment, you need to use the knowledge learned from Assignment 1.
2. You also need to use more Linux/Unix commands such as `dig`, `curl` or `crontab` as well as to make shell scripts.

   If you are not very familiar with Linux and Python, you can

   a. Watch the latency measurement and throughput measurement for this assignment. (You can find it also in the video section.) You could learn how to better start taking measurements in the video. You may wish to apply them in tasks.

   b. View supporting documents and linux_intro.pdf to look at those commands and codes in detail. It will help you in doing your assignment.

   c. Take a look at some code and code snippets 2 which may give you some help.
3. You need to have data processing skills. If you need guidance or further insights, you could refer to the introductory video data analysis.

Learning outcomes

At the end of this assignment, students should be able to

1. Measure network latency and throughput.
2. Learn some skills to make data processing easier.
3. Learn to analyze the performance of different servers.
4. Analyse the stability and variations of the network latency.
5. Analyse the methods to measure the throughput.

Introduction

This assignment has three tasks that complement ones for the first assignment. Please read all instructions before starting because it is helpful to identify common work.

- Task 1: Measurements metrics
- Task 2: Measuring latency
- Task 3: Measuring throughput

Task 1: Measurements metrics

You must answer the following points appropriately:

1. Explain the concept of network packet loss, its definition, calculation method, and its impact on network performance.
2. Compare the concepts and characteristics of mean and median, and discuss their advantages and disadvantages.
3. What is network stability? How can network stability be measured using various indicators or metrics?

Task 2: Measuring latency

For this task, you need to measure the latency on your own computer. If you are using macOS or Linux, you can easily utilize commands or execute shell scripts. However, if you are using Windows, we strongly recommend installing Windows Subsystem for Linux(WSL) or a Linux virtual machine on your computer to do the tasks. Also note that macOS command line utilities originate from BSD Unix and have different options than GNU based utilities in Ubuntu. You might need to adapt command and use workarounds.

1. Choose 3 name servers, 3 research servers and 2 iperf servers based on instructions at the servers section of this assignment. Start measurements and collect data for **at least 24 hours using a shell script and `crontab`.**

   > **Note:** Leave measurements running *at least for two weeks due to this data will be used in the Final Assignment.* Also put yourself a **reminder** to stop measurements after two weeks.
   >
   > It is recommended to use your own computer as the measurement source even if it is not running 24/7 as it will give more interesting results. Extend measurement time accordingly to have sufficient count of measurements. Even if you have metered subscription, the data volume won't be significant one.

   The table below describes the methods to apply for each server selected to measure the latency and defines the running time of the scripts based on

the columns '*Script executed*' and '*Selecting the minute*' to their respective servers.

Table 1: Measuring latency of each server with different method

Server	**Script executed**	**Selecting the minute**	**Method 1**	**Method 2**
3 name servers	Every hour	at *studentid modulo 60* [*1*]	5 ICMP echo request [*2*]	DNS query [*3*]
3 research servers	Every 10 mins	at *studentid modulo 10* [*1*]	5 ICMP echo request [*2*]	-
2 iperf servers	Every 10 mins	at *studentid modulo 10* [*1*]	5 ICMP echo request [*2*]	TCP connect latency [*4*]

NOTES:

1. A recommended way to get distributed (among students) value for minutes is to run on Linux `expr $(id -u) % 60`, where `$id` is not actually student id, but it is the Linux user ID (UID), but will serve an approximate one.
2. Using the command `ping` focusing on the *rtt*.
   - For Linux, check `-O` and `-D` options.
   - For macOS, check the `ts` tool.
3. Using the shell script `http-dig.sh` that can be found from supporting_material-code.zip, the focus is on measuring the latency by considering both the *query time* and *time_connect*. The total latency is obtained by summing the *query time* and *time_connect*. Running the script without any arguments provides a help message.
4. Using the command `curl` to send the **1K.bin file**, thereby it does not cause too much load, and to focus on the variables *time_connect* and *time_namelookup* (check `-w` options) to get the TCP connect latency by subtracting the *time_connect* - *time_namelookup*.

TIPS:

- It's possible to calculate the results by hand from the logs as there are only a few measurements, but in the future, we will have a bigger dataset that can be tedious or impossible to go through by hand; in addition, it helps to introduce machine learning later in the analysis.
- There must be three shell scripts created: one for name servers, one for research server measurements, and one for iperf servers. It is easier to run commands with `crontab` when command parameters are within script file and not in *crontab* file.
- Do not run command as at even hours but distribute over minutes based on your student id so not everyone at the course will run

command at the same time. This will most likely result a number of failed tests.
- You may log all output and make filtering and data collection later, or you can do filtering right away.
- After you have set up periodic measurements, **check after few hours** that results are what you expecting. Common errors are that you run commands too often, too seldom or you overwrite earlier results with a new ones and not append.

2. **Report a table** with following metrics for each of your target servers, including the results obtained using **all methods** (ones for method 1 and others for method 2).
   - Median delay with lost packets with delay of infinity, thus if more than 50 % of packets are lost, then consider as infinity.
   - Mean delay.
   - Loss ratio.
   - Delay spread as the difference with 75th and 25th percentiles. If more than 25 % of packets is lost, then consider as infinity.

Table 2: Example results from latency measurements in milliseconds.

Type	**Server**	**Method**	**Median delay**	**Mean delay**	**Loss ratio**	**Delay spread**
name server_1	a.ns-set.be	ping	29.839	30.357	0.0%	0.324
name server_2	b.ns-set.be	ping	29.743	29.940	0.0%	0.293
name server_3	y.ns-set.be	ping	12.157	12.773	0.0%	0.924
name server_1	a.ns-set.be	dig	32.000	49.384	0.0%	4.000
name server_2	b.ns-set.be	dig	28.000	29.846	0.0%	4.000
name server_3	y.ns-set.be	dig	12.000	13.076	0.0%	0.002
research server_1	cbg-uk.ark.caida.org	ping	40.748	40.963	0.0%	0.047
research server_2	us-man.nordu.net	ping	21.744	22.666	0.0%	2.314
research server_3	scl-cl.ark.caida.org	ping	232.550	232.910	0.0%	0.530

Type	Server	Method	Median delay	Mean delay	Loss ratio	Delay spread
iperf server_1	ok1.iperf.comnet-student.eu	ping	0.834	1.776	0.0%	2.254
iperf server_2	blr1.iperf.comnet-student.eu	ping	311.134	316.091	0.0%	13.338
iperf server_1	ok1.iperf.comnet-student.eu	curl	0.001	0.002	0.0%	0.001
iperf server_2	blr1.iperf.comnet-student.eu	curl	0.328	0.330	0.0%	0.011

3. Finally, **make conclusions** about stability of network delay. Were some of the hosts different from the others? Could you observe any daytime variations? Do the timezones where target servers (or you) have an impact?

Report, task 2

- Describe your measurement setup
- Table of measurement results.
- Conclusions on network stability.

Task 3: Measuring throughput

In this task, you will evaluate throughput using three different ways: file transfer, a specialized measurement tool, and a measurement service. The measurement service way requires manual execution, while the remaining ways can be automated using tools like `crontab`. Finally, you will compare the results obtained from these different approaches.

1. Start measurement and collect data for **at least 24 hours using a shell script and `crontab`**, and **describe your measurement setup** in the report.

   > **NOTE:** Leave measurements running *at least for two weeks due to this data will be used in the Final Assignment.* Also put yourself a **reminder** to stop measurements after two weeks.
   >
   > It is recommended to use your own computer as the measurement source even if it is not running 24/7 as it will give more interesting results. Extend measurement time accordingly to have sufficient count of measurements. Note, however, if you use metered subscription where you are charged by data volume, you might not want to use one.

   The table below defines the methods focused on measuring the throughput in different ways since most network users are interested only in a single

factor: *how many bits per second* can be downloaded or sent with their network connection. Therefore, run the following throughput tests at your home with your own computer.

> In optimum cases, one of the latency measurements should be run simultaneously as the throughput measurements.

Table 3: Measuring throughput in different ways

Way	**Script executed**	**Selecting the minute**	**Method 1**	**Tool**
1) *by file transfer*	Every hour	at *studentid modulo 60* [*2*]	HTTP download tool [*3*]	`curl`
2) *by a specialized measurement tool*	Every hour	at *studentid modulo 60* [*2*]	Network performance measurement tool [*4*]	`iperf3`
3) *by a measurement service*	Manually [*1*]	-	Measurement service	*e.g. Speed Test*

NOTES:

1. Run a few measurements **by hand with method 3** within the same time frame and write down the *date, time, and results from it*
2. A recommended way to get distributed (among students) value for minutes is to run on Linux `expr $(id -u) % 60`, where `$id` is not actually student id, but it is the Linux user ID (UID), but will serve an approximate one.
3. Download files from the target server using some HTTP download tool (**`curl` recommended**)
4. Network performance measurement tool **`iperf3` for 10 seconds in both upload and download directions** (check `-t` option).

   NOTE: Target servers for the method 1 and 2 are the same as the two iperf3 servers in Task 2.

2. **Make a table** where you results from these 3 methods and **calculate basic statistics**, such as mean, median, max, min and average deviation(mean absolute deviation). Note that for methods 2 and 3 you will receive upload (UL) and download (DL) readings. Report them separately.

Table 4: Example results from throughput measurements in megabits per second.

Key	HTTP (*server 1*)	HTPP (*server 2*)	Iperf UL (*server 1*)	Iperf DL (*server 1*)	Iperf UL (*server 2*)	Iperf DL (*server 2*)	ST UL	ST DL
2021-10-13 18:08:01	59.7905	1.1123	-	-	-	-	33.54	94.17
2021-10-13 19:08:01	54.8992	0.3974	11.8925	86.7741	0.2358	-	-	-
Mean	53.96195	0.4740	10.4326	84.5329	0.3156	0.5744	32.9	91.775
Median	57.3448	0.3261	11.8049	84.4473	0.2358	0.5744	32.9	91.775
Min	40.7213	0.1317	7.6005	82.3772	0.1646	0.4169	32.26	89.38
Max	60.4366	1.1123	11.8925	86.7741	0.5464	0.7319	33.54	94.17
Avg deviation	4.1048	0.1969	0.1298	3.0690	0.1056	0.2335	0.9488	3.5508

- *DL* = Download
- *UL* = Upload
- *ST* = SpeedTest
- *server 1* and *server 2* = the target servers selected.
- See definition of Avg deviation.

Report, task 3

- Describe your measurement setup. Include (example) code and samples of results.
- Table of results.
- **Make conclusions** about the methods of network throughput answering for at least the following topics. It may be **beneficial to graph results to identify some trends**.
  1. Are the results between methods in line with each other?
  2. Did some method have a lot of deviation? What do you think might cause this?
  3. Was there some method that gives higher values than others? What do you think might cause this?
  4. Did you observe any variation in throughput based on the time of day? For example, did you get higher throughput during the day or night?
  5. Were there any anomalies observed? For example, no connection or

very different capacity.

TIPS: Look how the log file structure looks like, and you could modify the existing `parse.py` file to suit the objective.

Servers

Nameservers

In the Aalto servers, the course includes some tools including `mycountry` which assigns your country based on your `ùserid` and performs a few checks, and provides some name servers related to the country.

1. The following commands must be run in one of Aalto server computers to **execute `mycountry`**

   ```
   source /work/courses/unix/T/ELEC/E7130/general/use.sh
   mycountry
   ```

   > **Note:** You may need to type the command `kinit` before to access to the directory. Also the source works with bourne shell compatible shells: bash and zsh (current Aalto default shell).

   The command will print something like this:

   ```
   br OK (Brazil): d.dns.br, e.dns.br, f.dns.br, a.dns.br, c.dns.br,
   b.dns.br Your UID is 1346517, thus your ccTLD is br (Brazil)
   ```

   For this example, the country assigned is Brazil with the next name servers:

   - `d.dns.br`
   - `e.dns.br`
   - `f.dns.br`
   - `a.dns.br`
   - `c.dns.br`
   - `b.dns.br`

2. In order to **select three servers**, you need to *test the connection* to the servers, if none of the nameservers do not respond to ICMP messages, try using `traceroute` to find the last hop before that server and use it as a target (it responds).

   > **Warning:** Do not test traffic more frequently than twice an hour.

   If you do not know which domain names exist, try a search `news site:br` with one popular search engine. Actually, it should not make a difference if the domain exists or not.

Note: You need to run the `mycountry` command yourself at Aalto IT `kosh.aalto.fi`, `lyta.aalto.fi`, `brute.aalto.fi` or `force.aalto.fi` server.

You can access them **easily via SSH connection remotely**. In the Aalto campus network direct DNS queries are disallowed from normal client networks. However, those are allowed from above servers. On the other hand, the same filtering applies to typical residential networks too. You need to *run actual DNS latency tests from those servers* from where direct DNS requests are allowed.

Note: Also make sure that you do not ask information from the local name server but the one far away!

Research servers

There are few distributed research testbeds, including Caida ARK that researchers can utilize for internet-wide measurements. Here we test latency for a few of these sites where we use the servers below.

1. **Based on the next formulas (use integer division), select three servers** from tables below hosts: one from the first table, and two from the second table.
   - **Server 1:** id_0 = *studentID % 3*
   - **Server 2:** id_{1a} = *studentID % 6*
   - **Server 3:** id_{1b} = *studentID / 7 % 6*

Table 5: Select one: id_0

id_0	Server 1
0	cbg-uk.ark.caida.org
1	arn-se.ark.caida.org
2	bcn-es.ark.caida.org

Table 6: Select two: *id1a* and *id1b*

id_{1x}	Server 2	Server 3
0	bjl-gm.ark.caida.org	jfk-us.ark.caida.org
1	hkg-cn.ark.caida.org	eug-us.ark.caida.org
2	scl-cl.ark.caida.org	hnl-us.ark.caida.org
3	cjj-kr.ark.caida.org	msy-us.ark.caida.org
4	nrt-jp.ark.caida.org	sjc2-us.ark.caida.org
5	mnl-ph.ark.caida.org	san2-us.ark.caida.org

Warning: Traffic towards this destination can be more frequent (measurement intervals of *10 minutes* towards these destinations are acceptable).

For example, if the *studentID* is 123456 and modulo 3 is applied to it in the case of server, the result will be *0* and therefore the server 2 to use is `bjl-gm.ark.caida.org`.

Iperf servers

Iperf3 servers accept only one connection at a time. The hosts running Iperf3 servers used in this course are configured to run 11 different instances, each on a different port from 5200 to 5210. Utilize , for example, `$RANDOM` variable to select port (`-p` option) at random.

1. The **first iperf3 server** to use is local `ok1.iperf.comnet-student.eu`.
2. The **second iperf3 server** is selected according to $id_2 = studentID \ modulo \ 2$ according to the following table.

Table 7: the second iperf server

id_2	Far away iperf server
0	blr1.iperf.comnet-student.eu
1	sgp1.iperf.comnet-student.eu

NOTES:

- An URL is for example, http://blr1.iperf.comnet-student.eu/10M.bin:80. Based on iperf tests, select that file size that is most appropriate.

Table 8: Sample file sizes

file	size
1K.bin	1 KiB
5K.bin	5 KiB
10M.bin	10 MiB
50M.bin	50 MiB
100M.bin	100 MiB
500M.bin	500 MiB
500G.bin	500 GiB

In addition, the iperf servers will serve following files over HTTP at TCP port 80 which is useful for the **task 2 *Measuring latency* using the command `curl`**.

- As it is not known in advance how much capacity is available in the network, it would be prudent to define *the maximum time for transfer*. Depending on used tool, there are alternatives:

  - **curl** supports `-m secs` max-time option that will abort transfer if it takes more than `secs` seconds. Remember to include the amount of *bytes transfered* to your output format.
  - With any program, it is possible to use the **timeout** command that will kill (or send a signal to) the program if it runs longer than set timeout. A used signal can be specified with `-s INT` to be `SIGINT` for example. To set **wget** command timeout to 60 seconds following command can be used: `timeout 60 wget https://…`

As with all assignments with periodic automated measurements, verify few hour after setup that commands are run with right paramiters and with right intervals, not too early. A common mistake has been that iperf3 command has been run way too long which will result errors for other students.

Grading standard

To pass this course, you need to achieve at least 15 points in this assignment. And if you submit the assignment late, you can get a maximum of 15 points.

You can get up to 30 points for this assignment:

Task 1

- Explain the concepts requested related to measurements. (3p)

Task 2

- Successfully complete the measurement and describe detailed measurement steps. (2p)
- Perform data processing on the data of each server, and get the table. The table contains the four required items. (8p)
- Draw appropriate conclusions about the problems (4p)

Task 3

- Successfully complete the measurement and describe detailed measurement steps. (1p)
- Perform data processing on the data of each method, and get the table. The table contains the required items. (8p)
- Draw appropriate conclusions about the problems (4p)

The quality of the report (bonus 2p)

The instruction of assignment

For the assignment, your submission must contain (Please don't contain original data in your submission):

- A zip file that includes your codes and scripts.

- A PDF file as your report.

Regarding the report, your report must have:

- A cover page indicating your name, student ID and your e-mail address.
- The report should include a description of measurements, a summary of the results and conclusions based on the results.
- An explanation of each problem, explain how you solved it and why you did it.

Annex

Delay spread

Delay spread is absolute difference of two quantilities. If like 25% of measurements are <= 125 ms and 75% are <= 175 ms then the delay spread between 25th ad 75th perceintiles are 50 ms (175 ms - 125 ms).

Aalto network topology

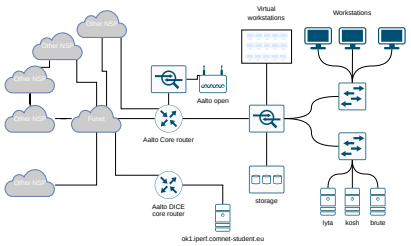

Figure 1: Principal-level view of Aalto network