ELEC-E7130 Internet Traffic Measurements and Analysis

Assignment 2. Basic measurements

Name: Xingji Chen

Student ID: 101659554

E-mail: xingji.chen@aalto.fi

Aalto University
School of Electrical
Engineering

Task 1: Measurements metrics

1. Explain the concept of network packet loss, its definition, calculation method, and its impact on network performance.

- Definition:

  Packet loss is the failure of one or more packets of data to reach its destination across the network. There are various possible reasons for packet loss, the most common being network congestion. When the amount of data transmitted over the network is too large, the network becomes congested, resulting in packets not reaching their destination in time. In addition, network equipment failures, long transmission distances, and network signal interference may also cause packet loss.
- Calculation method:

$$\text{Packet Loss Percentage} = \frac{\text{Number of Loss Packets}}{\text{Total Number of Packet Sent}} \times 100$$

- Impact of packet loss
  1) Impact on network transmission speed: packet loss will lead to slower network transmission speed, because the data packets need to be retransmitted, thus increasing the time of network transmission.
  2) Impact on network transmission quality: packet loss will lead to a decline in the quality of network transmission, the loss of packets will result in incomplete data transmission or errors.
  3) Impact on the experience of network applications: packet loss will lead to a decline in the experience of network applications.

2. Compare the concepts and characteristics of mean and median, and discuss their advantages and disadvantages.

Mean:

- Concepts:

  The mean is adding all the values and dividing by the number of values.
- Calculation formula:

$$\mu = \frac{1}{N}\sum_{i=1}^{N} x_i$$

- Advantages:
  1) Make full use of all data and have strong applicability.
  2) Suitable for situations with normal distribution or symmetric data.
- Disadvantages:
  1) Mean is susceptible to extreme values.
  2) Not suitable for skewed data or data that does not follow a normal distribution.

Median:

- Concepts:
  The median is the middle value in a data set when all values are arranged in ascending or descending order. If the number of values is an even number, the median is the average of the two middle values.
- Advantages:
  1) Robust to outliers, suitable for situations where data is skewed or contains extreme values.
  2) Represents typical values in a data set, especially when the data does not follow a normal distribution.
- Disadvantages:
  1) Not as accurate as mean because it does not use all data values.
  2) May be affected by how the data is sorted rather than the actual values themselves.
  3) May not work well when there are many values that occur with the same frequency in the data.

3. What is network stability? How can network stability be measured using various indicators or metrics?

- Network stability refers to the ability of the network to maintain continuous and reliable operation without interference or drops during the process of carrying data transmission, ensuring the integrity, accuracy and timeliness of data transmission. Network stability is an important indicator of network performance, which is directly related to network usage experience and service quality.
- Various indicators and metrics:
  1) Round Trip Time (RTT): RTT measures the time it takes for a packet to travel from source to destination and back again. High or varying RTT values may indicate network instability.
  2) Packet Loss Percentage: A low packet loss rate is an indicator of network stability.

3) Throughput Measurement: Throughput measures the amount of data that can be transmitted in a given amount of time. Consistent and adequate throughput is critical to network stability.
4) Error Rate: This metric measures the rate at which packets encounter errors during transmission. A high error rate may indicate network instability or hardware problems.
5) Network Congestion Metrics: Monitoring and measuring network congestion, such as network link utilization or queuing delays, can help assess network stability under varying loads.

Aalto University
School of Electrical
Engineering

Task 2: Measuring latency

1. Description of measurement setup

- Your UID is 3281478, thus your ccTLD is st (Sao Tome and Principe)

Server	Link
3 name servers	dns-st.bahnhof.net
	ns1.bahnhof.net
	southeast-2.dns-au.st
3 research server	cbg-uk.ark.caida.org
	bjl-gm.ark.caida.org
	msy-ius.ark.caida.org
2 iperf servers	ok1.iperf.comnet-student.eu
	blr1.iperf.comnet-student.eu

1) Method: 5 ICMP echo request

- Calculation of mean:

```
awk -F'[=/ ]+' '/rtt/ {print $7}' dns-st.bahnhof.net.txt |
awk '{sum+=$1; count++} END {print "Mean:", sum/count}'
```

This command extracts the seventh field from the line containing "rtt" in the file, calculates the average of these values, and prints the result in the form of "Mean:".

- Calculation of Median:

```
awk -F'[=/ ]+' '/rtt/ {print $7}' dns-st.bahnhof.net.txt |
sort -n |
awk '{a[i++]=$1} END {if (i%2==0) print "Median:",
(a[int(i/2-1)] + a[int(i/2)])/2; else print "Median:",
a[int(i/2)]}'
```

This command extracts the seventh field of the line containing "rtt", sorts them in ascending order, and then calculates the median. The result is printed as "median:".

- Calculation of Average Deviation:

```
awk -F'[=/ ]+' '/rtt/ {print $7}' dns-st.bahnhof.net.txt |
sort -n |
awk 'BEGIN{percentile=0.75; count=0} {data[count++]=$1}
END{idx=int(count*percentile); diff=data[idx]-
data[int(count*(1-percentile))]; print "Diff:", diff}'
```

This command extracts the seventh field from the line containing "rtt", sorts it in ascending order, and then calculates the difference between the 75th percentile value and the 25th percentile value (interquartile range). The result is printed as "Mean Deviation:".

2) Method: DNS query

- Preprocess:

```
awk '/Query time|Return|CURL/' dns-st.bahnhof.net.txt >>
dns-st.bahnhof.net.reloaded.txt
```

The purpose of this AWK command is to filter lines containing the "Query time", "Return" or "CURL" keywords from the anycastdns2.nic.td.txt file and append these lines to anycastdns2.nic.td.reloaded.txt in file.

- Data processing:

```
import re
import numpy as np

query_time_list = []
connect_time_list = []

with open('data/dns_query/dns-st.bahnhof.net.reloaded.txt',
'r') as file:
    skip_next_line = False
    for line in file:
        stripped_line = line.strip()
        if skip_next_line:
            skip_next_line = False
            continue
        if re.search("Return|Query|CURL", stripped_line):
            parts = stripped_line.split()
            if len(parts) >= 4:
                time_value = float(parts[3])
                if re.search("Query", stripped_line):
```

```
                    query_time_list.append(time_value)
                elif re.search("CURL", stripped_line):
                    connect_time_list.append(time_value)
                elif re.search("Return", stripped_line):
                    skip_next_line = True

delay_list = [(query_time / 1000 + connect_time * 1000) for
query_time, connect_time in zip(query_time_list,
connect_time_list)]

delay_array = np.array(delay_list)

median = np.median(delay_array)

mean = np.mean(delay_array)

percentile_75 = np.percentile(delay_array, 75)
percentile_25 = np.percentile(delay_array, 25)
percentile_diff = percentile_75 - percentile_25

print("Median:", median)
print("Mean:", mean)
print("Percentile Difference:", percentile_diff)
```

The main function of this code is to extract data for specific keywords ("Query", "CURL", "Return") from a text file, calculate the delay and perform statistical analysis (median, mean, 75th and 75th) the difference between the 25th percentile).

3) Method: TCP connect latency

```
awk '
    !/^[0-9]{4}-[0-9]{2}-[0-9]{2} [0-9]{2}:[0-9]{2}:[0-
9]{2}$/ && NF > 0 {
        diff = $2 - $1;
        diffs[NR] = diff;
    }
    END {
        asort(diffs);
        n = length(diffs);
        median = (n % 2 == 1) ? diffs[int(n / 2) + 1] :
(diffs[n / 2] + diffs[n / 2 + 1]) / 2;
```

```
        median_ms = median * 1000;
        print "Median:", median_ms;
    }
' ok1.iperf.comnet-student.eu.txt

awk '
    !/^[0-9]{4}-[0-9]{2}-[0-9]{2} [0-9]{2}:[0-9]{2}:[0-
9]{2}$/ && NF > 0 {
        diff = $2 - $1;
        sum += diff;
        count++;
    }
    END {
        average = (sum / count) * 1000;
        print "Mean:", average;
    }
' ok1.iperf.comnet-student.eu.txt

awk '
    !/^[0-9]{4}-[0-9]{2}-[0-9]{2} [0-9]{2}:[0-9]{2}:[0-
9]{2}$/ && NF > 0 {
        diff = $2 - $1;
        diffs[NR] = diff;
    }
    END {
        asort(diffs);
        n = length(diffs);
        percentiles_75 = diffs[int(n * 0.75) + 1] * 1000;
        percentiles_25 = diffs[int(n * 0.25) + 1] * 1000;
        percentile_diff = (percentiles_75 - percentiles_25);
        print "Percentiles_75:", percentiles_75;
        print "Percentiles_25:", percentiles_25;
        print "Diff:", percentile_diff;
    }
' ok1.iperf.comnet-student.eu.txt
```

This AWK code performs processing and analysis of the data in the text file ok1.iperf.comnet-student.eu.txt, calculates the median, average and percentile differences respectively, and prints out the corresponding results at the end. .

Aalto University
School of Electrical
Engineering

2. Table of measurement results

Type	Server	Method	Median delay	Mean delay	Loss ratio	Delay spread
name server_1	dns-st.bahnhof.net	ping	17.962	15.886	0.0 %	1.393
name server_2	ns1.bahnhof.net	ping	19.233	16.161	0.0 %	3.256
name server_3	southeast-2.dns-au.st	ping	329.953	327.144	0.0 %	4.637
name server_1	dns-st.bahnhof.net	dig	2444.271	2490.907	0.0 %	687.095
name server_2	ns1.bahnhof.net	dig	2497.382	2524.543	0.0 %	737.096
name server_3	southeast-2.dns-au.st	dig	2723.663	2791.945	0.0 %	698.794
search server_1	cbg-uk.ark.caida.org	ping	38.855	36.790	0.0 %	2.108
search server_2	bjl-gm.ark.caida.org	ping	115.664	114.236	0.0 %	3.359
search server_3	msy-ius.ark.caida.org	ping	169.521	167.340	0.0 %	4.847
iperf server_1	ok1.iperf.comnet-student.eu	ping	6.279	4.565	0.0 %	1.363
iperf server_2	blr1.iperf.comnet-student.eu	ping	323.675	321.569	0.0 %	17.113
iperf server_1	ok1.iperf.comnet-student.eu	curl	4.561	6.294	0.0 %	1.616
iperf server_2	blr1.iperf.comnet-student.eu	curl	332.269	333.960	0.0 %	12.434

3. Conclusions on network stability

1) Network Latency Stability:

Metrics such as median and average latency were calculated. These metrics provide insights into the overall stability of network latency. Lower median and average latency typically indicate a more stable network, while higher values indicate greater variability. Thus, in nameserver, nameserver_1, which is dns-st.bahnhof.net is more stable. In search server, searchserver_1, which is cbg-uk.ark.caida.org is more stable. In iperf server, iperfserver_1, which is ok1.iperf.comnet-student.eu is more stable.

2) Differences between hosts:

There can be differences between hosts. For example, another student and I each measured different results on our laptops.

3) Daytime variations:

During the daytime, when the network is less congested, the delay is relatively low. When the network is congested, such as at 6pm, the delay is slightly higher.

4) Impact of time zones:

Because my UID is 3281478, thus my ccTLD is st (Sao Tome and Principe), which is in a different time zone than Finland. Thus there will be an impact on the measurement of network delay.

Aalto University
School of Electrical
Engineering

Task 3: Measuring throughput

1. Description of measurement setup

1) HTTP download tool

```
import re
import statistics

def convert_to_mbps(speed_str):
    if speed_str.endswith('M'):
        return float(speed_str.rstrip('M'))
    elif speed_str.endswith('k'):
        return float(speed_str.rstrip('k')) / 1000
    else:
        return float(speed_str) / 1000000

download_speeds = []

with open('data/http_download/ok1.iperf.comnet-student.eu.txt',
'r') as file:
    for line in file:
        stripped_line = line.strip()
        if re.search("100", stripped_line):
            items = stripped_line.split()
            speed_str = items[6]
            download_speed = convert_to_mbps(speed_str)
            download_speeds.append(download_speed)


mean_speed = statistics.mean(download_speeds)
median_speed = statistics.median(download_speeds)
min_speed = min(download_speeds)
max_speed = max(download_speeds)
speed_deviation = statistics.stdev(download_speeds)


print(f"Mean: {mean_speed:.8f} Mbps")
print(f"Median: {median_speed:.8f} Mbps")
print(f"Min: {min_speed:.8f} Mbps")
```

```
print(f"Max: {max_speed:.8f} Mbps")
print(f"Avg deviation: {speed_deviation:.8f} Mbps")
```

The main purpose of this code is to analyze download speed data read from a file. The download speed data was converted to the more understandable units of megabits per second and statistical information about this speed data was calculated. The mean, median, minimum, maximum, and average deviation are then output.

2) Network performance measurement tool

```
import re
import statistics

speeds = []

with
open('data/network_performance_measurement_tool/ok1.iperf.comnet
-student.eu.txt', 'r') as file:
    for line in file:
        stripped_line = line.strip()
        if re.search("receiver", stripped_line):
            items = stripped_line.split()
            speed = float(items[4])
            speeds.append(speed)

print(speeds)

mean_speed = statistics.mean(speeds)
median_speed = statistics.median(speeds)
min_speed = min(speeds)
max_speed = max(speeds)
avg_deviation_speed = statistics.mean([abs(x - mean_speed) for x
in speeds])

print("Mean:", mean_speed)
print("Median:", median_speed)
print("Min:", minimum_speed)
print("Max:", maximum_speed)
print("Average Deviation:", avg_deviation_speed)
```

This code is used to process network performance measurements read from a file and to compute statistical metrics about them, which are then printed to the console.

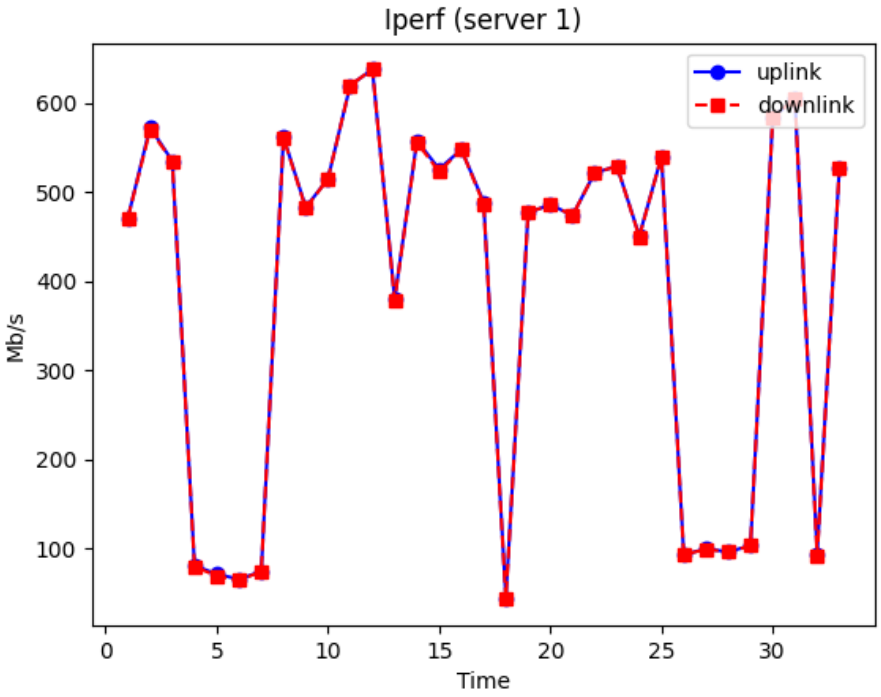
Aalto University
School of Electrical
Engineering

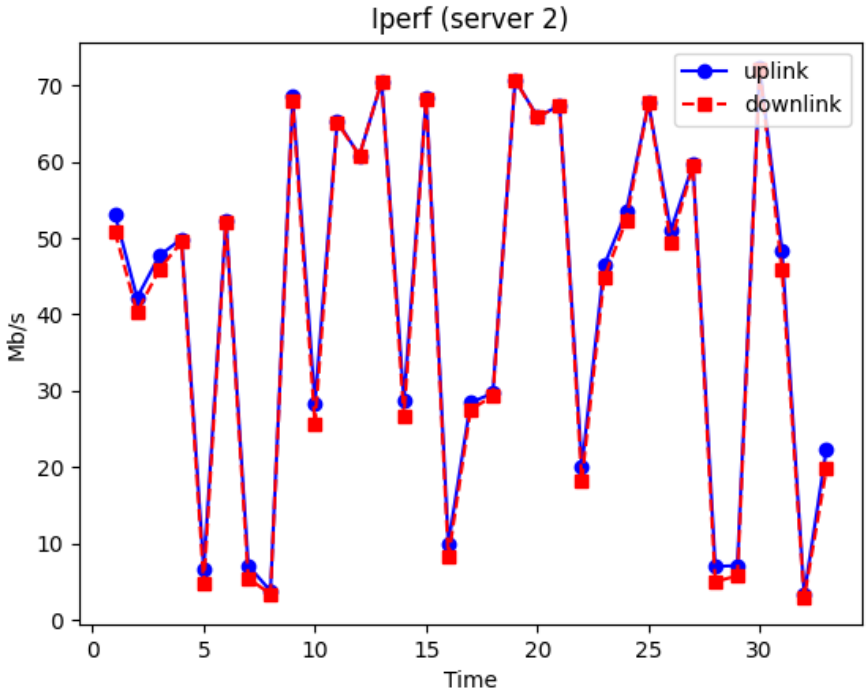
3) Method: Measurement service Speedtest

2. Table of measurement results

Key	HTTP (server1)	HTTP (server2)	Iperf UL (server 1)	Iperf DL (server 1)	Iperf UL (server 2)	Iperf DL (server 2)	ST UL	ST DL
2023-9-30 21:54:01	8.251	2.463	537.0	536.0	46.1	44.4	801.52	560.45
2023-10-2 9:54:01	4.436	1.904	462.0	460.0	28.7	26.6	771.35	663.99
Mean	4.509	4.520	461.555	460.966	42.717	41.712	776.07	562.52
Median	4.291	4.351	534.0	533.0	46.4	45.1	773.09	561.98
Min	2.229	2.229	56.6	56.6	3.81	3.32	701.67	441.64
Max	8.258	8.258	744.0	743.0	70.7	70.7	887.32	663.99
Avg deviation	1.064	1.047	156.433	156.499	19.945	20.385	43.131	54.909

3. Conclusions





Aalto University
School of Electrical
Engineering

1) Are the results between methods in line with each other?
   - No. The measurement results of curl and iperf3 are significantly different because they focus on different things and how they measure. Curl measures the speed of specific application layer downloads such as the throughput of a web browser downloading files, while iperf3 measures raw throughput on a network link.

2) Did some method have a lot of deviation? What do you think might cause this?
   - Yes. The method of iperf3 have a lot of deviation. Iperf3 performs raw network measurements, which may introduce additional overhead and limitations, causing deviation in throughput results.

3) Was there some method that gives higher values than others? What do you think might cause this?
   - Yes. Measurement service (e.g. Speed Test) gives higher values than others. Because different measurement methods may use different protocols or techniques. Some protocols may be more efficient than others, resulting in higher throughput. Also, different measurement tools can have different implementations and may have different optimizations for data processing and calculations. These implementation differences may result in differences in measurement results.

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?
   - Yes. Network throughput can fluctuate throughout the day. In general, the throughput during the day is greater than at night. This may be because the network quality is better during the day, but at night the network quality decreases due to the increase in access volume.

5) Were there any anomalies observed? For example, no connection or very different capacity.
   - Yes. At some point the throughput suddenly decreases. It may be because the number of users has increased significantly, which has reduced the quality of the network.