**T1. Performance and Reliability Analysis of Communication Networks**

Submit to srp@es.aau.dk with subject [NetPerf24][T1][Your Group Name]

**Task 1.** Outline differences between circuit-switching and packet-switching.

Et billede, der indeholder tekst, skærmbillede, Font/skrifttype

Automatisk genereret beskrivelse

In circuit switching a wired connection is established beforehand connecting two different machines together, whereafter communication can begin. In the early days of telephones connections were manually changed by operators to connect user, showcasing an example of circuit switching.

In packet switching the connection contains information regarding it’s destination, thus it can propagate a network and arrive at the correct location. Thus many packets can propagate at the same time and the exact connection from endpoint to endpoint is not needed as this can be handled by the network

**Task 2.** What is a counting process? Which distribution is used to model the counting

process?

A counting process is a process where we count the occurrences of some types of events. For example, you might have a random process N(t) that shows the number of packets received by a router by time t starting from time 0. For such a processes, we usually assume N(0)=0, so as time passes and packets arrive, N(t) takes positive integer values.

In the scope of electronics and communication we often use to Possion distribution for our counting processes. This is because Poisson process is one of the most widely-used counting processes. It is usually used in scenarios where we are counting the occurrences of certain events that appear to happen at a certain rate, but completely at random (without a certain structure). This is very applicable to scenarios such as packets arriving to a router.

**Task 3.** What are basic HTTP requests?

In order for information to be exchanged

Et billede, der indeholder tekst, skærmbillede, Font/skrifttype, nummer/tal

Automatisk genereret beskrivelse

The basic HTTP request is defined by:

Get: gets data form the server in focus

Options: request to know the servers functionalities: describe the communication options for the server

Post: sends the server in focus some data

Head: Gets only the metadata from the server

**Task 4.** Analyze the HTTP trace. You can use “Wireshark”.

• Get the HTTP trace:

https://kevincurran.org/com320/labs/wireshark/trace-http.pcap

• Filter the HTTP traffic.

• Find an HTTP request (basically, get the first GET packet and expand its block).

Write the source and destination IP.

• Show the headers and comment on what they mean. (e.g., “Host”: this is a

mandatory header that identifies the name and port of the server)

• Show response to the first GET in the trace; expand the block. Report “Status

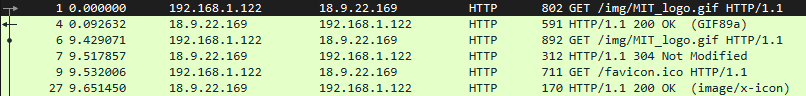
Code” and “Status Code Description”.

• Can you verify if the server needs to send fresh content for the second GET re-

quest? (Hint: check the third and fourth HTTP trace; check the “if-modified-since”

header.)

We find the first REQUEST:



With source and destination:



We look at the HTTP request:

Et billede, der indeholder tekst, skærmbillede, Font/skrifttype, software

Automatisk genereret beskrivelse

And see that the GET request also has the headers Host (mit.edu.dk) which is the name of the server.

We see connection, which tells the server to keep the connection alive.

We see cache control says the response is only fresh for 0 seconds.

And finally the accept headers which tell which content types can be used, such as language, typeset and encoding.

We now look at the response

Et billede, der indeholder tekst, elektronik, skærmbillede, software

Automatisk genereret beskrivelse

Where we see the response was ok, status code 200.

The request succeeded. The result meaning of "success" depends on the HTTP method:

* GET: The resource has been fetched and transmitted in the message body.

We note from the first picture that the content has not been modified and thus no new content needs to be sent. The server also response this with status code 304 Not Modified.

304 Not Modified

This is used for caching purposes. It tells the client that the response has not been modified, so the client can continue to use the same cached version of the response.

**Task 5.** Download the exercise program from https://luca31.github.io/HTTP-requests-

exercises/ and post the code and results obtained in the report. Print the POST requests

and the relative responses in full, including the URL, body, and headers. What request

code numbers do the actual packets use?

If you get an “unhandled error event” when you start the server, open index.js and

comment the three lines after the comment “copy to clipboard,” i.e., lines 101-103 in the

file. The server should work then.

**Result is in .ipynb file.**