Basic Tracing

In this assignment you will perform a basic tracing of two different scenarios using LTTng tracer and will analyze the trace files in Tracecompass. You should identify the target application among the collected trace file and the sequence of execution it goes through. You will explain this along with appropriate screenshots of the trace analysis. You will also have to discuss potential performance bottlenecks. Your mark will be awarded based on the following equally weighted criteria:

- Correctness and completeness of the report (1500 ~ 2000 words and six ~ eight figures).
- Organization of the report.
- Professionalism illustrated through the report.
- Creativeness and meticulousness in your investigation and discussions.
- Oral in class presentation of your findings.

Get familiar with Netcat:
https://www.digitalocean.com/community/tutorials/how-to-use-netcat-to-establish-and-test-tcp-and-udp-connections

LTTng:
https://lttng.org/docs/v2.12/

and Tracecompass:
https://www.eclipse.org/tracecompass/

Before you do any tracing make sure you can easily execute the steps and see the packets being generated on wireshark. Once you fully understand the workflow start performing the experiments.

Experiment I

Create a tracing session with vpid,vtid,procname context information. You should enable all system calls as well as the following events:

Scheduler events so you know which thread is running in each cpu at any given time:
`sched_switch,sched_waking,sched_wakeup`
Network events: `net_*`
Socket buffer events: `skb_*`
Interrupt related events to see when a packet has been received and an interrupt is raised:
`irq_*`

Start tracing:

Using netcat open a tcp socket and listen on it:
```
nc -l 127.0.0.1 4000
```

On a different terminal create a tcp client and connect to the nc server:
```
nc -p 4444 127.0.0.1 4000
```

Then you can type in on the client side and the data will be sent over to the server when you press enter. Close the connection using Ctrl+C on the client or server side.

Stop trace collection
Visualize in Tracecompass and perform your analysis of the trace as instructed.

Experiment II

Now use telnet to www.google.com and issue an HTTP GET request as follows:

Start tracing

```
telnet www.goog.com 80
GET / HTTP/1.1
Host: www.google.com
Accept: */*

Ctrl+C
```

Stop tracing

Do trace analysis and see the difference between the two traces. In particular you should see the wifi or ethernet interface and drivers getting involved in the second experiment compared to the first one.

Experiment III

Repeat the same experiments as before but this time perform a ping to your default gateway. Change the interval and packet size. What change do you see in the collected trace if the flood option is used (ping -f)? What change do you see when the packet size is set to 64KB (ping -s 65507)?