Exercise 6

Distribution Fitting, Goodness of Fit Test

In exercise 5 you simulated a network, in which a client was constantly uploading a file via *FTP* to a server. If you think about other, more complex types of traffic, the question arises how to model this appropriately.

Take web browsing for example. Here you can not expect a constant data transmission, but a user which retrieves a website and needs some time for reading before making the next request. These time periods also need to be configured within your simulation model, so that the user behaviour is represented properly.

This exercise will explain the method of distribution fitting, which can be used to create a probability distribution based on empirical data. The goodness of the fit is afterwards evaluated by employing the so called χ^2 test.

1) Find out how to perform the χ^2 test for the *Poisson* distribution on a given series of packet arrival rates (packets per time unit). Give a detailed pseudo-code example[1] for the calculation and explain your solution.

2) Implement a *MATLAB* script to perform the χ^2 test on a given vector. Use the built-in functions and set the parameters appropriately, based on your knowledge from the lecture and task 1.

3) Reconfigure your network from the previous two exercises. Instead of TCP, now UDP is used between the client and the server. Please see the example *omnetpp.ini* file at the end of this exercise sheet. Perform simulations and export the vector of the incoming packet rate at the server.

4) Perform the χ^2 test for the *Poisson* distribution on the incoming packet arrival rates. Plot the sample data as well as the fitted distribution, so that you can see the differences between them. Are your simulation results accepted by the χ^2 test? If not, consider simulating again with a changed simulation time.

Advanced additional tasks:

5) Implement your pseudo-code example from task 1 in *MATLAB*, without using most of the built-in functions from task 2.

6) Compare the results of your own implementation with the results from task 4.

[1] independent of the programming language/tool used

omnetpp.ini: *Remove the TCP applications and use the following UDP configuration*

```
# Client settings
**.client.numUdpApps = 1
**.client.udpApp[0].typename = "UDPBasicApp"
**.client.udpApp[0].destAddresses = "server"
**.client.udpApp[0].destPort = 1000
**.client.udpApp[0].sendInterval = exponential(0.01s)
**.client.udpApp[0].messageLength = 50 B

# Server settings
**.server.numUdpApps = 1
**.server.udpApp[0].typename = "UDPSink"
**.server.udpApp[0].localPort = 1000
```

For more details on how "ThruputMeter" computes the throughput see the file inet/src/inet/common/misc/ThruputMeter.cc