The MatplotViz tool has the big advantage that it doesn't need a lot of configuration. Simply install or setup a python environment with some basic libraries and you're good to go. The data will simply be stored in csv files and can be plotted in either a real time manner or in a replay manner. A big disadvantage is the basic visualization constraints. Although efforts were made to make the visualization as adaptable and customizable as possible it is not possible to achieve the same visual results as for example by using a tool like Thingsboard.

MatplotViz uses matplotlib to plot real time data or data that was stored in advance. The modified adaptive\_mbsfn will take the clientside role. Running this mbsfn module is the same as the original adaptive mbsfn module. The eNodeB will now try to connect with the server. If the connection is successful it will send the required data to the server. If the connection isn't successful, nothing will happen and the parameters will just be printed onto the terminal, just as before. Remark: the IP of the server is default set to localhost but can be set in the network\_components/client.cc file. If you do not wish to run the mbsfn program you can also run the generators/client.py to generate and send random parameters to the server.

If you wish to capture and store data, the network\_components/server.py should be running on the machine with the IP that the client is sending data to. This server program will be listening for incoming tcp connections and for incoming messages. When it recieves a message, it will increase the frame index and it will store this index alongside the recieved parameters in a csv file of choice.

If you want to plot the data stored in a csv file, you have to run the monitors/main.py. This will either start a piechart or bardiagram monitor. The monitor program is capable of replaying an existing static csv file or a csv file that is updating without any configuration changes. Just run the command. The program will replay the existing data until it catches up with the last row stored in the file. If a new row is added it will automatically detect this and plot this new value. Note that filename, horizontalwindowSize, sleeptime and columnToPlot can all be set. Note that the server doesn't need to be running to run the monitor.

**HOW TO RUN**

**clientside:**

* Option 1: run src/generators/client.py to generate and send random values
* Option 2: run the modified adaptive\_mbsfn which sends the parameters to the server to get real life parameters

**serverside**:

* CAPTURE DATA: run src/network\_components/main.py --addressOfServer "insertaddreshere" or just default "127.0.0.1" (same with client.py)
* VISUALIZE DATA: run src/monitors/main.py --monitorType "insert bardiagram or piechart" or just default "bardiagram"

**REMARKS**

!!! MAKE SURE THAT THE IP OF THE SERVERADDRESS IS CONFIGURED CORRECTLY ON BOTH THE SERVER AND CLIENTSIDE!! --addressOfServer "insert ip"

!!! MONITOR ADJUST SLEEPTIME FOR FRAMERATE => 0 MEANS ASAP

!!! Filename arguments just last filename part without "csv\_files"

**REQUIREMENTS**

Matplotlib: “Pip install matplotlib”

Pandas: “Pip install pandas”

Filelock “Python -m pip install filelock”