

## Read Me

Mrinal Aich (cs16mtech11009)

### Steps:

1. `chmod +x shellScript.sh`
2. `./shellScript`

### Minor Modifications to run full/half-buffers:

In `shellScript.sh` at line 34:

For Full buffer: `--fullBufferFlag=1`

For Half buffer: `--fullBufferFlag=0`

### Files:

1. `*/ns3/ns-allinone-3.25/ns-3.25/scratch/sourceCode.cc` // Source code
2. `*/ns3/ns-allinone-3.25/ns-3.25/mergerScript.py` // Merges all runs into one-file
3. `*/ns3/ns-allinone-3.25/ns-3.25/automationScript.py` // Reads the data and generate plots
4. `*/ns3/ns-allinone-3.25/ns-3.25/shellScript.sh` // Shell script for all type of runs

### Logic:

1. Each run of shell script generates graphs for "Average Aggregate System throughput", "Throughput CDF plot" each for speed of 0 and 5 m/s, "SINR" and "Instantaneous throughput" for IMSI-1(UE-0) each at 0 and 5 m/s speed.
2. Different Runs are required each for full-buffer and half-buffer case.
3. During the each iteration, 'mergerScript.py' merges the contents of "DIRlcStats.txt" into a temporary-file "myDIRlcStatsAt0" or "myDIRlcStatsAt5" depending on the speed of the UE(s) (for speed 0 or 5 m/sec respectively).  
Also, it adds the name of the scheduler-type to the file, which is used by *automationScript.py* for statistical calculations.
4. DIRsrpSinrStats: This stat is required for only one RngRun, hence shellScript.sh creates the file using a special parameter to *mergerScript.py* (described there).

### Functionality of the files:

1. `*/ns3/ns-allinone-3.25/ns-3.25/mergerScript.py` :  
Appends the contents from "DIRlcStats.txt" into a temporary file "myDIRlcStats<0|5>.txt" taking input schedulerName and speed-value.  
Also, creates another temporary files "myDIRsinrStatsAt<0|5>.txt" for a specific RngRun.
2. `*/ns3/ns-allinone-3.25/ns-3.25/automationScript.py` :  
Statistical analysis on the data of the temporary files and creates the necessary plots (mentioned above).
3. `*/ns3/ns-allinone-3.25/ns-3.25/shellScript.sh` :  
Runs "sourceCode.cc" over mutiple RngRuns for different Schedulers and different speeds. It calls *mergerScript.py* in every iteration with scheduler names and speed-value which appends all the contents into a temporary file.  
Finally "automationScript.py" creates all the necessary plots.

**Output:**

For each run the following files are generated depending upon full/half-buffer scenario.

1. AggregateSysThroughput.png - Average Aggregate System throughput for four scheduling algorithms at speed 0 and 5 m/sec as X-axis.
2. UeThroughputCdfAt<0|5>.png - Throughput CDF plot for different schedulers at Speed (0 or 5) m/s.
3. SinrImsi\_UE1\_At<0|5>.png - SINR value for UE 0 (IMSI-1)  
X-axis: Time in sec, Y-axis: SINR Speed of 0 or 5 m/s for all four schedulers.
4. InstantThroughput\_UE1\_<0|5>.png - Instantaneous throughput values for UE 0 (IMSI-1).  
X-axis: Time in sec, Y-axis: Instantaneous throughputs for Speed of 0 or 5 m/s for all four schedulers.

**Temporary Files:**

1. myDIRlcStatsAt<0|5>.txt – Merged files containing DIRlcStats.txt data for all schedulers across all RngRuns at speed 0 or 5 m/s.
2. myDISinrStatsAt<0|5>.txt - Merged files containing DIRsrpSinrStats.txt data for all schedulers for one RngRun at speed 0 or 5 m/s.

**System Requirements:**

- ns3.25
- python 2.7