

Hands-on ns-3 LTE tutorial

Zoraze Ali*, Thomas R. Henderson†

Centre Tecnològic de Telecomunicacions de Catalunya (CTTC/CERCA),

Castelldefels (Barcelona), Spain*

University of Washington Seattle, WA†

June 16, 2019



I. EXAMPLE PROGRAMS

A. *lena-simple.cc*

This example simulates a scenario with one eNB and a UE without EPC. It uses the LteRlcSm as the RLC model, and without supporting any IP networking. In other words, it will be a radio-level simulation involving only LTE PHY, MAC and the scheduler, with a saturation traffic model for the RLC.

The program options are as follows:

```
$ ./waf --run lena-simple --command-template="%s --PrintHelp"
Program Options:
  --simTime:      Duration of the simulation [+1050000000.0ns]
  --useCa:        Whether to use carrier aggregation [false]
  --rlcStatsEpoch: RLC stats epoch duration [+250000000.0ns]
  --numBearers:   Number of data radio bearers to be activated. [1]
```

The above output shows the available options and their default values, which are configurable through the command line. Let's run this example with these default parameters.

```
$ ./waf --run lena-simple
```

Once the simulation ends, run the following command to list the traces generated during this simulation.

```
$ ls
```

One could find various traces collected from the different layers of LTE protocol stack, e.g., "lena-simple.DlRlcStats.txt", "lena-simple.DlMacStats.txt", etc. The "DI" and "UI" prefix corresponds to down-link and uplink, respectively. For this example, we will be focusing on the RLC and MAC statistics.

```
$ less lena-simple.DlMacStats.txt
$ less lena-simple.DlRlcStats.txt
$ less lena-simple.UlMacStats.txt
$ less lena-simple.UlRlcStats.txt
```

Visit this page:

www.nsnam.org/docs/models/html/lte-user.html#simulation-output

Let's try to answer the following questions with help of traces.

- 1) What constitutes the delay values in DL and UL RLC statistics?
- 2) Why the MCS in DlMacStats.txt stays 0 for first few entries?

```
$ ./waf --run lena-simple --command-template="%s --connectCustomTraces=1"
$ less lena-simple.cqi.dat
```

- 3) Why the number of transmitted and received PDUs in DlRlcStats.txt are not equal?

```
$ ./waf --run lena-simple --command-template="%s --rlcStatsEpoch=0.001"
$ less lena-simple.DlRlcStats.txt
```

Let's add one more data radio bearer:

```
$ ./waf --run lena-simple --command-template="%s --rlcStatsEpoch=0.001 --numBearers=2"
```

4) Why the number of TX bytes for the first bearer in DlRlcStats.txt are lower this time?

```
$ less lena-simple.DlRlcStats.txt
```

Before moving on to the next example, run the following command to save the configuration steps used to build this example.

```
$ ./waf --run lena-simple 2> lena-simple-setup.dat
```

B. lena-simple-epc.cc

This example simulates a complete LTE network, i.e., EUTRAN + EPC. This simulation script, with the default parameters will generate the following topology Fig. 1.

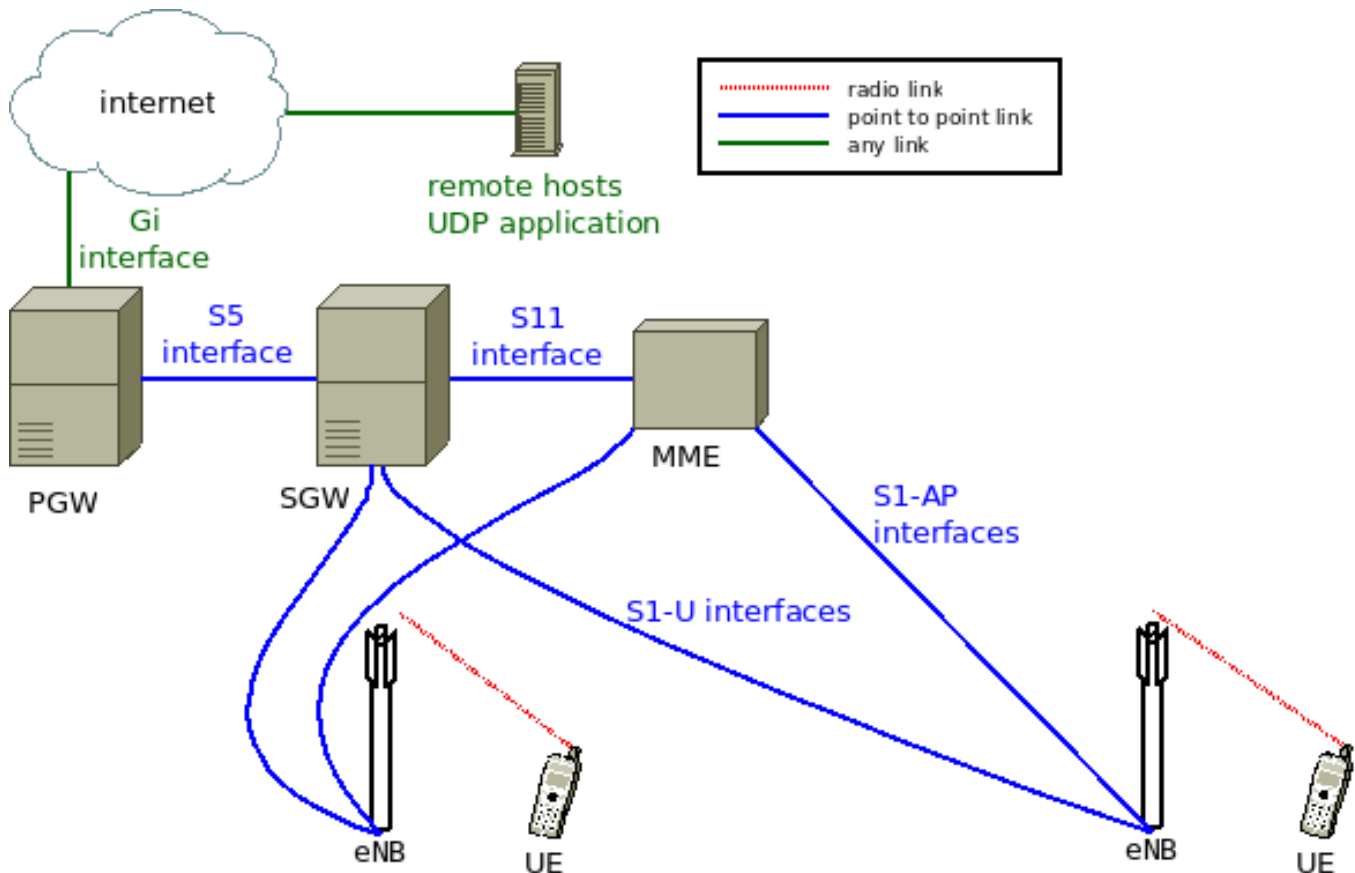


Fig. 1. lena-simple-epc topology

Go ahead, and run the following command.

```
$ ./waf --run lena-simple-epc --command-template="%s --PrintHelp"
```

Program Options:

```
--numNodePairs:      Number of eNodeBs + UE pairs [2]
--simTime:           Total duration of the simulation [+1100000000.0ns]
--interEnbDistance:  Distance between eNBs [m] [60]
--ueDistance:        Distance between a UE and a eNB [m] [10]
--interPacketInterval:  Inter packet interval [+100000000.0ns]
--useCa:             Whether to use carrier aggregation [false]
--disableDl:         Disable downlink data flows [false]
--disableUl:         Disable uplink data flows [false]
--disablePl:         Disable data flows between peer UEs [false]
--printPosition:     Flag to print the positions of the eNB and UE nodes [false]
```

Let's see the differences between the configuration steps followed to build the lena-simple, and this example.

```
$ ./waf --run lena-simple-epc 2> lena-simple-epc-setup.dat
```

```
$ pr -m -t --width=135 lena-simple-setup.dat lena-simple-epc-setup.dat
```

Output:

lena-simple.cc	lena-simple-epc.cc
1. Create an LteHelper object	1. Create an LteHelper object
2. Create Nodes: eNodeB and UE	2. Create a PointToPointEpcHelper object
3. Install Mobility Model	3. Tell the LTE helper that the EPC will be used
4. Create Devices and install them in the Nodes (eNB and UE)	4. Connect PGW to a remote host
5. Attach a UE to a eNB	5. Create Nodes: eNodeB and UE
6. Activate data radio bearer	6. Install Mobility Model
	7. Create Devices and install them in the Nodes (eNB and UE)
	8. Install the IP stack on the UEs
	9. Assign IP address to UEs and set the default gateway for the UEs
	10. Attach one UE per eNodeB
	11. Install and start applications on UEs and remote host

For this example, we will be focusing on the PDCP and RLC statistics.

```
$ less lena-simple-epc.DlPdcStats.txt
```

```
$ less lena-simple-epc.DlRlcStats.txt
```

```
$ less lena-simple-epc.UlPdcStats.txt
$ less lena-simple-epc.UlRlcStats.txt
```

C. lte-tcp-x2-handover.cc

The goal of simulating this example is to simulate the LTE X2 based handover, and to explore the performance evaluation of a long-running TCP flow from an Internet server. Fig. 2 illustrates the network topology considered for this simulation.

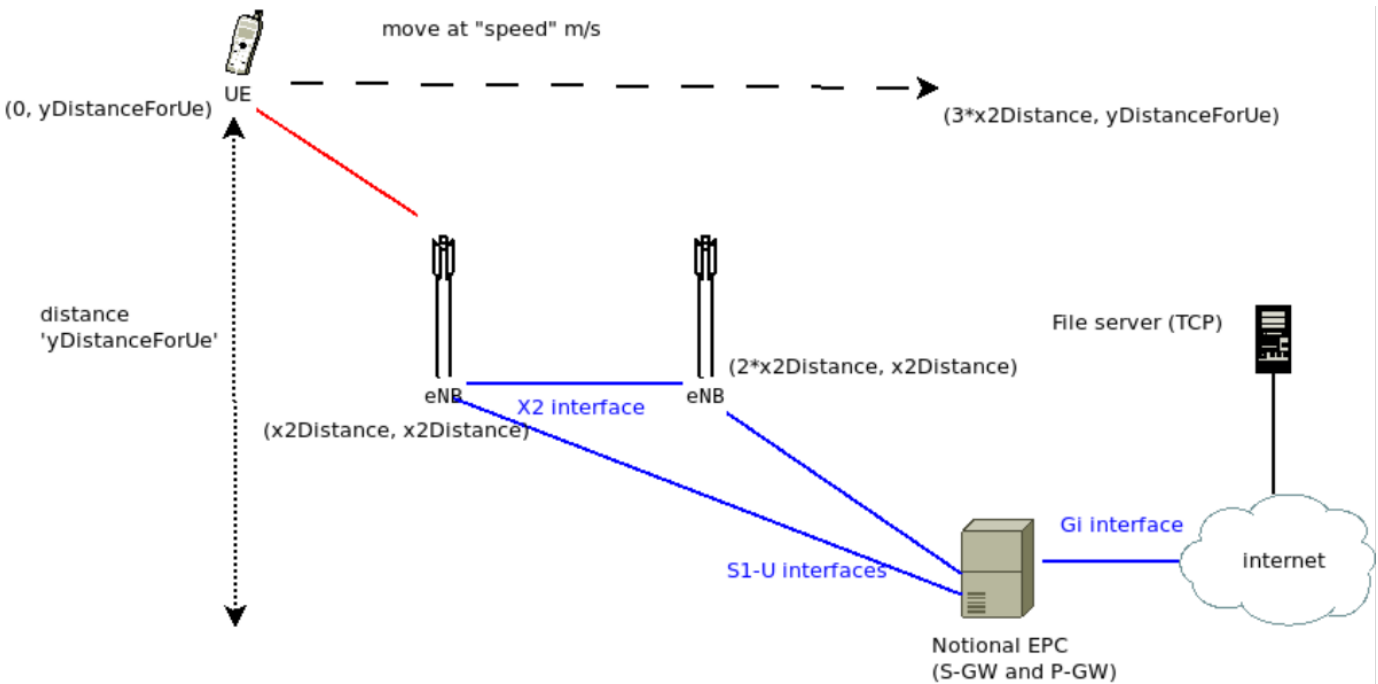


Fig. 2. lte-tcp-x2-handover topology

```
$ ./waf --run lte-tcp-x2-handover --PrintHelp
```

Program Options:

```
--speed:           Speed of the UE (m/s) [20]
--x2Distance:      Distance between eNB at X2 (meters) [500]
--yDistanceForUe: y value (meters) for UE [1000]
--enbTxPowerDbm:  TX power (dBm) used by eNBs [46]
--useRlcUm:       Use LTE RLC UM mode [false]
--handoverType:   Handover type (A2A4 or A3Rsrp) [A2A4]
--generateRem:    If true, will generate a REM and then abort the simulation [false]
--pcap:           Enable pcap tracing [false]
--verbose:        Enable verbose logging [false]
```

The UE moves horizontally at a speed, default 20 m/s. The simulation time is configured to correspond

to the time it takes for the UE to move from initial position for a distance of $3 \times x2Distance$. This allows the UE to move to the halfway point between the eNBs around the halfway point of the simulation.

Let's start by plotting the Radio Environment Map (REM) of the scenario.

```
$ ./waf --run lte-tcp-x2-handover --command-template="%s --generateRem=1"
output:
Initial positions: UE: (0,1000), eNB1: (500,500), eNB2: (1000,500)
Simulation time: 75 sec
```

This simulation script is capable to generate three gnuplot compatible scripts. The two files for plotting the points indicators for the eNBs and the UE positions, and the third one to plot the REM generated by "RadioEnvironmentMapHelper". Go ahead, and run the following command:

```
$ gnuplot -p x2-ho-rem-ue-position.dat x2-ho-rem-enb-position.dat x2-ho-gnuplot-rem-script.dat
```

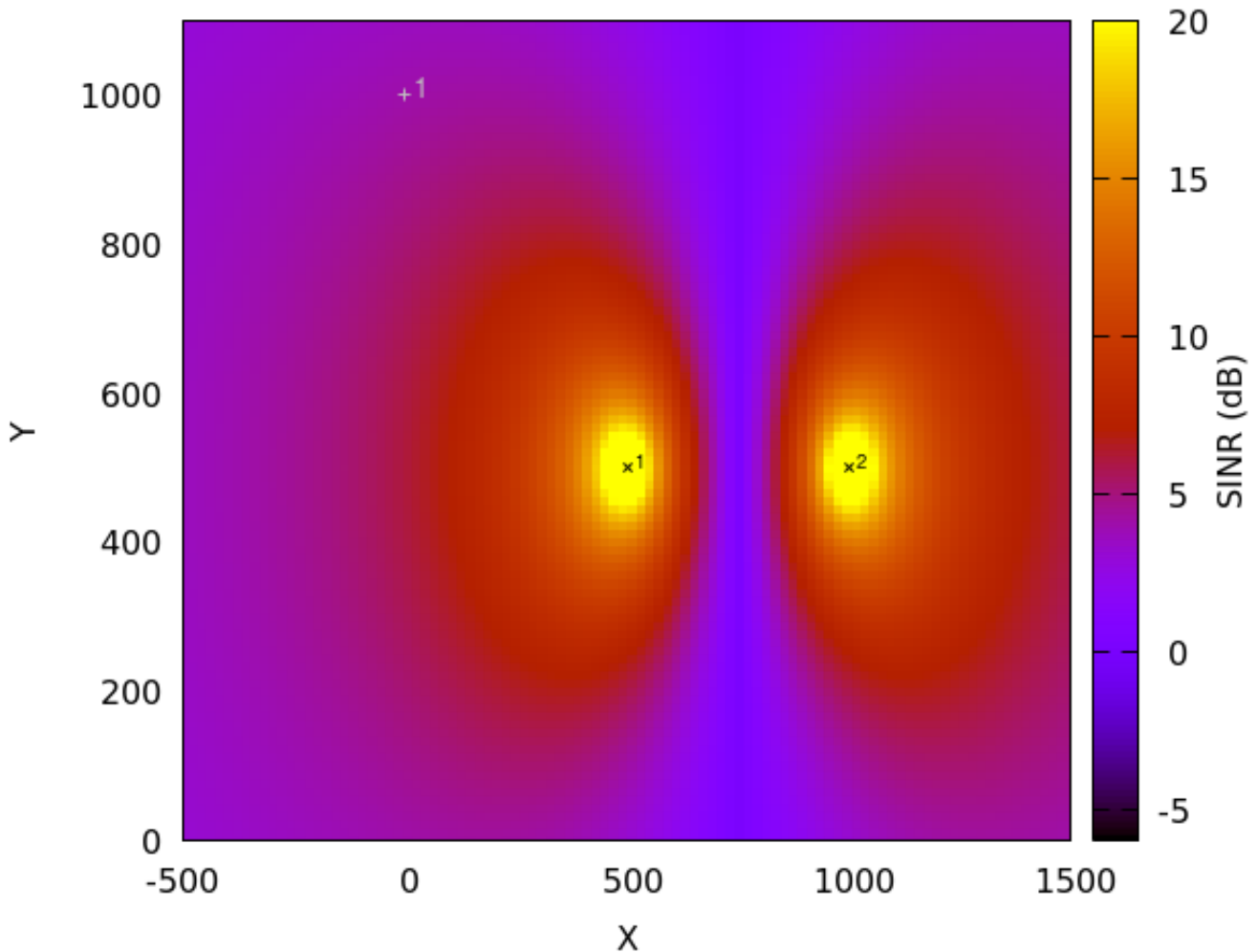


Fig. 3. Radio environment map

The REM in Fig. 3 is generated using the DL control channel and shows the SINR (dB) distribution in this scenario.

Three additional options shown above can further tune the simulation parameters. The “enbTxPowerDbm” can be used to change the downlink eNB power. The “useRlcUm” can be used to force the use of the RLC Unacknowledged Mode (by default, RLC Acknowledged Mode is used). Finally, the “handoverType” argument allows the user to change between the A2A4 or A3Rsrp algorithms. The last two command line arguments can enable PCAP traces (for use in Wireshark) and some ns-3 logging (if the simulator has been built in debug mode). Upon running the program, a number of data files are generated, as described below. The output to terminal is as follows:

```
$ ./waf --run lte-tcp-x2-handover
Initial positions: UE: (0,1000), eNB1: (500,500), eNB2: (1000,500)
Simulation time: 75 sec
0.0202143 node 4 UE IMSI 1: connected to CellId 1 with RNTI 1
0.0202143 node 2 eNB CellId 1: successful connection of UE with IMSI 1 RNTI 1
*** Simulation time: 10.0s
*** Simulation time: 20.0s
*** Simulation time: 30.0s
38.2 node 2 eNB CellId 1: start handover of UE with IMSI 1 RNTI 1 to CellId 2
38.2 node 4 UE IMSI 1: previously connected to CellId 1 with RNTI 1, doing handover to CellId 2
38.2 node 4 UE IMSI 1: successful handover to CellId 2 with RNTI 1
38.2 node 3 eNB CellId 2: completed handover of UE with IMSI 1 RNTI 1
*** Simulation time: 40.0s
*** Simulation time: 50.0s
*** Simulation time: 60.0s
*** Simulation time: 70.0s
```

The first two statements print out the initial positions and the simulation time, respectively. Simulation progress is also noted by statements preceded by asterisks. The first events near the start of the simulation indicate initial cell acquisition of cell 1 by the UE. At time 38.2, the handover algorithm initiates handover, which is quickly completed. The simulation ends at time 75 seconds.

The example uses the Round Robin scheduler (`RrFfMacScheduler`) that has been slightly modified to output a wideband CQI trace, and in general, other ns-3 LTE defaults.

1) *LTE stats*: Let’s have a look at the “DlMacStats.txt”.

```
$ less DlMacStats.txt
```

The scheduler allocates, at MCS value 28, a transport block of size 2196. If one searches the trace, one will find that at most 2196 bytes are allocated each subframe (1 ms), leading to a peak throughput of

$2196 \times 8 / 0.001 = 17.6 \text{ Mb/s}$. In actuality, the long-running average TCP throughput that can be achieved will be somewhat less than this due to overhead.

2) *TCP stats*: TCP statistics are gathered in two files. The first is ‘lte-tcp-x2-handover.tcp-receive.dat’, which lists the number of bytes and timestamp for each socket read operation at the receiver (UE). This file can be used to compute end-to-end throughput (a plotting file described below does this). The second is ‘lte-tcp-x2-handover.tcp-state.dat’ which lists changes to the TCP congestion control state machine. The key event to look for in this file is any example of a ‘RECOVER’ or ‘LOSS’ event.

For instance, the following example shows a TCP fast recovery event:

```
# time    congState
38.190 CA_DISORDER
38.192 CA_RECOVERY
38.413 CA_OPEN
```

The first ‘DISORDER’ event represents TCP detecting a sequence number gap. This leads to a ‘RECOVERY’ event, then an ‘OPEN’ event signifying the end of the loss recovery.

A TCP timeout looks similar, but is identified by a ‘LOSS’ event:

```
# time    congState
50.887 CA_DISORDER
51.852 CA_LOSS
52.155 CA_OPEN
```

3) *CQI data*: The wideband CQI data is written to the file ‘lte-tcp-x2-handover.cqi.dat’.

4) *Position data*: The position of the UE is written to the file ‘lte-tcp-x2-handover.position.dat’ every 5 seconds.

5) *UeMeasurements*: The RSRP and RSRQ reported values are written to the file ‘lte-tcp-x2-handover.ue-measurements.dat’. The file has the following format:

```
# time    cellId    isServingCell?  RSRP (dBm)  RSRQ (dB)
0.200    1         1                -74.718     -4.471
0.200    2         0                -78.701     -8.453
0.400    1         1                -74.684     -4.69
0.400    2         0                -78.693     -8.458
```

The UE reports the values of all cells that it hears. The third column of the trace file indicates which cell the UE belongs to (the ‘isServingCell?’ field). For instance, at time 0.200, cell 1 is serving and cell 2 is not.

The following table is provided as a general guide as to how to interpret the values of RSRP and RSRQ, in practice:

TABLE I
LTE RSRP AND RSRQ

channel state	RSRP (dBm)	RSRQ (dB)
excellent	≥ -80	≥ -10
good	-80 to -90	-10 to -15
mid cell	-90 to -100	-15 to -20
cell edge	≤ -100	< -20

(table is drawn from <https://www.cablefree.net/wirelesstechnology/4glte/lte-rsrq-sinr/>)

D. Experiment directory

The program run can be orchestrated with a shell script in the directory `contrib/training-2019/experiments/lte-tcp-x2-handover/`.

This automates the generation of a number of plots:

- TCP throughput over time
- CQI over time
- MCS over time
- RSRP over time
- RSRQ over time

The following figures provide the plots generated from the default program configuration. When values are plotted in red, this is due to the UE associating with the cell 1, and values plotted in blue are for times for which the UE is associated with cell 2.

The TCP throughput in Fig. 4 shows a disruption to throughput performance at the handover time; the rate drops by half (or more) and then must rebound after the loss event. The handoff is not ‘hitless’ as there must be some packet drop that the RLC AM is not able to recover.

The mobility-induced RSRP and RSRQ changes are clearly seen in Figures 5 and 6.

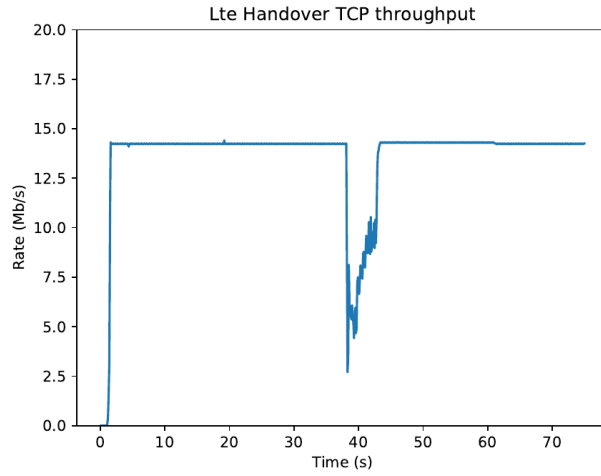


Fig. 4. TCP throughput vs. time

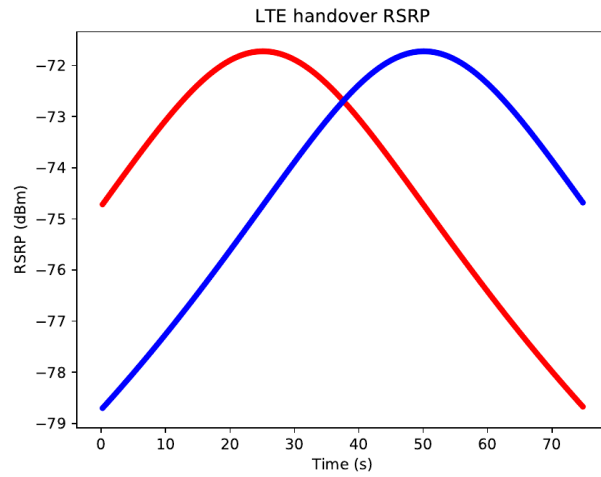


Fig. 5. RSRP vs. time

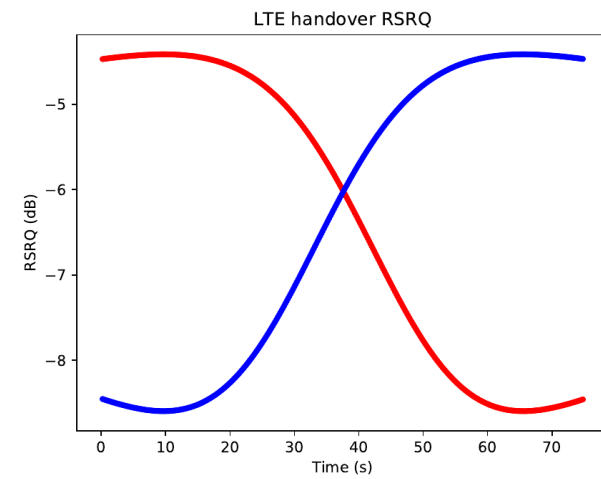


Fig. 6. RSRQ vs. time

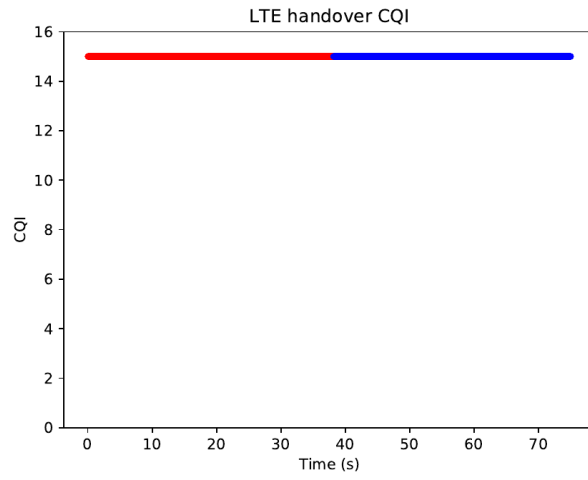


Fig. 7. CQI vs. time

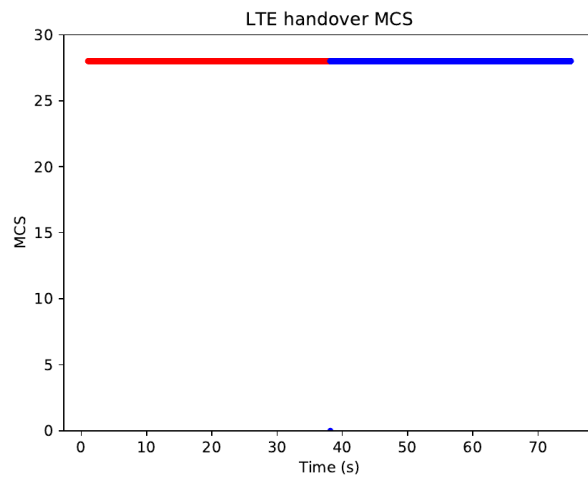


Fig. 8. MCS vs. time