Vienna LTE Simulators
System Level Simulator Documentation, v1.8r1375

Institute of Telecommunications
Vienna University of Technology, Austria
Gusshausstrasse 25/389, A-1040 Vienna, Austria
Email: {mtaranet}{mmueller}@nt.tuwien.ac.at
Web: http://www.nt.tuwien.ac.at/ltesimulator

Abstract

This document contains documentation on how to use the LTE System Level Simulator (LTE SL Simulator) [1] as well as some insight on its structure and the assumptions that were made while developing it. This document relates more on how to actually use the simulator. The concept and the structure of the simulator is described in more detail in [2].

I. Foreword

The LTE SL Simulator is published under a non-commercial academic use license. Please make sure that you understand the terms and conditions of the license before you use any of the available software packages. Would you require a license different to a non-commercial academic one please contact Josep Colom Ikuno or Martin Taranetz.

The detailed license agreement for the LTE SL Simulator can be found in Section XXIII. Please read it carefully and keep in mind that the usage of the simulator is subjected to compliance with the license terms.

Contents

I Foreword 1

II Running a simulation 4

III Simulation parameters 4
- III-A Debug options . . . 4
- III-B Plotting options . . . 4
- III-C General parameters . . . 4
- III-D Random number generation options . . . 5
- III-E Simulation time . . . 5
- III-F Cache options . . . 5
- III-G Network layout and macroscopic pathloss parameters . . . 5
  - III-G1 Generated network parameters . . . 5
  - III-G2 Capesso-imported network parameters . . . 6
- III-H Shadow fading (only for generated networks) . . . 6
- III-I Small-scale fading . . . 6
- III-J UE settings . . . 7
- III-K eNodeB settings . . . 7
- III-L Scheduler settings . . . 7
- III-M CQI mapper options . . . 8
- III-N Uplink channel options . . . 8
- III-O SINR averaging . . . 8
- III-P Saving the results . . . 8
- III-Q Optional configuration parameters . . . 8

IV Implementation issues 11
- IV-A How to read the pathloss maps . . . 11
- IV-B Antenna azimuth . . . 11
- IV-C Remote Radio Head (RRH) . . . 11
- IV-D Block Error Ratio curves and CQI mapping . . . 11

V Plotting results 13
- V-A eNodeB and UE position . . . 13
- V-B Throughput and aggregate results . . . 14

VI Fractional Frequency Reuse (FFR) 15

VII Coordinated Multipoint (CoMP) 15

VIII Femtocells and small cells 16

IX User Equipment (UE) distributions 16

X 3D antenna radiation patterns 16

XI Importing a network topology from Capesso$^{\text{TM}}$data 17

XII Using the Winner Phase II channel model reference implementation 17

XIII A note on the use of the CVX Matlab toolbox 17

XIV Employing UE traces for simulating 17

XV Matlab automatic code generation (codegen) 18
- XV-A Shadow fading generation . . . 18
- XV-B Calculation of interferer terms according to the precoders applied by the interfering eNodeBs and the interfering channel(s) . . . 19
- XV-C Receiver filter calculation according to the precoder . . . 19
- XV-D Capacity calculation for each precoder in a codebook for feedback calculation . . . 19

XVI Examples and Reproducible results 20
XVI-A Comparison with link level results: simulation over SNR range 20
XVI-B UE throughput scheduler comparison 20
XVI-C Closed Loop Spatial Multiplexing throughput comparison for several antenna configurations 21
XVI-D Adjusting antenna separation in the Winner II channel model 22
XVI-E Fractional Frequency Reuse performance results 23

XVII Changelog 25

XVIII Referencing 30

XIX Known issues 30

XX Questions 30

XXI Mailing List 30

XXII The People (so far) behind the development of the LTE System Level Simulator 30

XXIII License agreement 31
XXIII-A Academic Usage 31
XXIII-B Grant of copyright license 31
XXIII-C Grant of source code license 31
XXIII-D Exclusions from license grant 31
XXIII-E Warranty of provenance and disclaimer of warranty 31
XXIII-F Limitation of liability 31
XXIII-G Termination 32
XXIII-H Open source and code under other license terms 32
XXIII-I Appendix I 32

XXIV Acknowledgments 33

References 33

List of Figures 34

List of Tables 34

II. Running a simulation

The main file of the LTE System Level Simulator (LTE SL Simulator) is `LTE_sim_main.m`, though **you may normally run the simulation through a batch file**. Various demo batch files are available in the folder `sim_main_launcher_files` and can directly be executed from there. The following tasks are performed:

- Loading a configuration file of choice. See Section III for a list of configurable parameters.
- Executing the `LTE_sim_main.m` main simulation file.

The simulation parameters are loaded from the `LTE_load_params_*.m` script or alternatively from similarly-named files. Do note that the `LTE_load_params_dependant.m` script is used for automatically generating additional simulation parameters from the base parameters specified.

III. Simulation parameters

Below you can find a list of the parameters that can be configured in the configuration parameters files. You can find all of them in the `+simulation_config` simulator subfolder. The files are loaded via the `LTE_load_params` function. Check `LTE_sim_main_launcher_examples` and `LTE_load_params` function for a list of possible preconfigured simulation files.

When called with no arguments, `LTE_load_params` loads the `+simulation_config/hex_grid_tilted` file. The optional string argument allows you (among others) to load the following other preconfigured setups (see `LTE_sim_main_launcher_examples`):

- `tri_sector`
- `tri_sector_tilted`, `tri_sector_tilted_4x2` `tri_sector_tilted_4x4`.
- `tri_sector_plus_femtocells`
- `six_sector_tilted`
- `capesso_pathlossmaps`
- `omnidirectional_eNodeBs`

A. *Debug options*

- `LTE_config.debug_level`: configures how much debug text output is shown. Options are:
  - `0`: no output.
  - `1`: basic output.
  - `2`: extended output.

B. *Plotting options*

- `LTE_config.show_network.`: configures how much plots are shown. Options are:
  - `0`: no plots shown.
  - `1`: show some plots.
  - `2`: show all plots, which includes one showing the moving User Equipments (UEs), which may slow down simulations significantly.
  - `3`: show also the plots of the generated microscale fading traces.

C. *General parameters*

- `LTE_config.frequency.`: frequency in which the system is operating [Hz].
- `LTE_config.bandwidth.`: system bandwidth. Allowed values are 1.4 MHz, 3 MHz, 5 MHz, 10 MHz, 15 MHz, and 20 MHz. This bandwidths are equivalent to 6, 15, 25, 50, 75, and 100 Resource Blocks (RBs) respectively.
- `LTE_config.nTX`: number of transmit antennas. Used to generate the channel trace.
- `LTE_config.nRX`: number of receive antennas. Used to generate the channel trace.
- `LTE_config.tx_mode`: the transmission modes are defined in TS 36.213-820 Section 7.1, page 12 [3].
  - `1`: single antenna.
  - `2`: Transmission Diversity (TxD).
  - `3`: Open Loop Spatial Multiplexing (OLSM). Spatial multiplexing with Large Cyclic Delay Diversity (CDD).
  - `4`: Closed Loop Spatial Multiplexing (CLSM).
  - `5`: Multiuser MIMO.
  - `9`: Eight Layer Spatial Multiplexing

D. Random number generation options

These options allow you to actually reproduce the same exact simulation by means of resetting the random number generator to a known seed.

- `LTE_config.seedRandStream`: in order to allow repeatability, it is possible to seed MATLAB's default random number generator. Set it to either `true` or `false`.
- `LTE_config.RandStreamSeed`: if the above is set to `true`, it specifies the seed. Seeds must be an integer between 0 and 2^{32} [4].

E. Simulation time

- `LTE_config.simulation_time_tti`: length of the simulation in Transmission Time Intervals (TTIs).

F. Cache options

- `LTE_config.cache_network`: whether you want to save the generated eNodeBs, Pathloss map and Shadow fading map to a `.mat` file. Either `true` or `false`. All cache options work in the following way:
  - cache=`true` and file exists: read cache file.
  - cache=`true` and file does not exist: create and then store data in cache file.
  - cache=`false`: do not use cache at all.
- `LTE_config.network_cache`: the name of the cache file. set it to `auto` if you want the simulator to assign a name automatically.
- `LTE_config.delete_ff_trace_at_end`: since the microscala fading trace takes up large amounts of space, when doing the final `save` command, it is preferable to delete it, so as not to have too large result files.
- `LTE_config.delete_pathloss_at_end`: Further reduces the amount of space needed to store the traces by deleting the pathloss maps from the results file.
- `LTE_config.UE_cache`: whether to save the user position to a file. Either `true` or `false`.
- `LTE_config.UE_cache_file`: the name of the cache file. set it to `auto` if you want the simulator to assign a name automatically.

G. Network layout and macroscopic pathloss parameters

These parameters specify how the network layout is created. However, if the map is loaded, these parameters will be overwritten by the loaded map.

- `LTE_config.network_source`: Available options
  - `generated`: A hexagonal grid of equidistantly-spaced eNodeB sites with three sectors each will be created.
  - `capesso`: eNodeB position, configuration, and pathloss data are read from data exported from and written from the Capesso$^{\text{TM}}$planning tool (see Section XI). When using this source, shadow fading data is not generated, as the imported pathloss maps should already have it incorporated.
  - `fixed`: Generates N eNodeBs, each with a constant pathloss specified in the `LTE_config.pathlosses` vector, containing N values in dB. Useful for comparison with link level results, where a single pathloss value would be desired for all of the UEs.

1) Generated network parameters:

- `LTE_config.network_geometry`: Available options
  - `regular_hexagonal_grid`: Regular hexagonal grid with fixed inter-eNodeB distance, as defined in `LTE_config.inter_e`
  - `stochastic`: Stochastic eNodeB distribution with spatial density
  - `hybrid`: Hybrid spatial distribution with deterministic- and random parts
  - `predefined`: eNodeB locations as given in `LTE_config.eNodeB_positions`

  The functions are defined in `+network_geometry`.
- `LTE_config.map_resolution`: in meters/pixel. Also the resolution used for initial user creation.
- `LTE_config.nr_eNodeB_rings`: number of eNodeB rings. 0 rings specifies that just a single eNodeB will be created.
- `LTE_config.minimum_coupling_loss` (optional): describes the minimum loss in signal [dB] between Base Station (BS) and UE or UE and UE in the worst case and is defined as the minimum distance loss including antenna gains measured between antenna connectors. Recommended values [5] are 70 dB for urban areas, 80 dB for rural.
- `LTE_config.macroscopic_pathloss_model`: sets what macroscopic pathloss model is to be used. Depending on the choice, different choices are available for `LTE_config.macroscopic_pathloss_model_settings.environment`. The available macroscopic pathloss models are:

  - `free space`: free space pathloss. More for testing purposes than for actual use with simulations. $L = \left(\frac{4\pi d}{\lambda}\right)^{\alpha}$. d in meters. It allows for the following parameters to be specified in `LTE_config.macroscopic_pathloss_model_settings.environment`:
    * α: the α coefficient employed in the pathloss calcualtion.
  - `cost231`: COST231 pathloss model. The possible options for `LTE_config.macroscopic_pathloss_model_settings.environment` are:
    * `urban micro`: microcell LOS and NLOS pathloss based on the COST231 Walfish-Ikegami model, see TR25.996 and COST 231 book.
    * `urban macro`: urban macrocell pathloss based on the COST 231 extended Hata model, see 3GPP TR25.996 and COST 231 book.
    * `suburban macro`: suburban macrocell pathloss based on the COST 231 extended Hata model, see 3GPP TR25.996 and COST 231 book.
  - `TS36942`: see [5] for more information. Possible environments are:
    * `urban`: $L = 40\left(1 - 4 \cdot 10^{-3} \cdot \text{Dhb}\right) \cdot \log_{10}(R) - 18\log_{10}(\text{Dhb}) + 21\log_{10}(f) + 80\,\text{dB}$. Where R is the base station-UE separation in km, f the carrier frequency in MHz and Dhb is the base station antenna height in metres, measured from the average rooftop level.
    * `suburban`: $L = 69.55 + 26.16 \cdot \log_{10}(f) - 13.82 \cdot \log_{10}(\text{Hb}) + (44.9 - 6.55 \cdot \log_{10}(\text{Hb}))\log_{10}(R) - 4.78\left(\log_{10}(f)\right)^2 + 18.33 \cdot \log_{10}(f) - 40.94$. Where R is the base station-UE separation in km, f the carrier frequency in MHz and Hb is the base station antenna height above ground in metres.
  - `TS25814`: see [6] for more information. $L = I + 37.6 \cdot \log_{10}(R)$. Where R is the base station-UE separation in km and $I = 128.1$ when using a 2 GHz carrier and $I = 120.9$ for 900 MHz.
- `LTE_config.eNodeB_tx_power`: eNodeB's maximum transmit power, in Watts. Recommended by [7] are:
  - 43 dBm for 1.25, 5 MHz carrier
  - 46/49 dBm for 10, 20 MHz carrier.

2) Capesso-imported network parameters: See Section XI for a detailed list of the Capesso-related parameters.

H. *Shadow fading (only for generated networks)*

- `LTE_config.shadow_fading_type`:
  - `claussen`: It generates a lognormal-distributed 2D space-correlated shadow fading map, as in [8].
  - `none`: No shadow fading map. For simplicity reasons this is implemented as a shadow fading map with a constant value of 0 dB.
- `LTE_config.shadow_fading_map_resolution`: map resolution for the shadow fading pathloss map (metres/pixel).
- `LTE_config.shadow_fading_n_neighbors`: specifies the number of neighbors the algorithm takes into account when space-correlating the shadow-fading mapts. Possible options are `4` and `8`, which use R_5 and R_9 [8] respectively.
- `LTE_config.shadow_fading_mean`: mean (μ) of the lognormal distribution.
- `LTE_config.shadow_fading_sd`: standard deviation (σ) of the lognormal distribution.
- `LTE_config.r_eNodeBs`: inter-site shadow fading correlation. The correlaton between the sectors in a site is fixed to 1 (same shadow fading map).

I. *Small-scale fading*

Small-scale (Microscale) fading trace to be used between the eNodeB and its attached UEs.

- `LTE_config.channel_model.type`: which PDP to use for the channel generation. Available options are:
  - `PedA`: ITU Pedestrian A channel [9].
  - `PedB`: ITU Pedestrian B channel [9].
  - `extPedB`: Extension of the ITU channel models for wideband (OFDM) systems [10].
  - `VehA`: ITU Vehicular A channel [9].
  - `VehB`: ITU Vehicular B channel [9].
  - `winner+`: Winner II+-based channel model. The implementation utilizes the codebase from the Winner project [11]. For more information, refer to Section XII.
- `LTE_config.channel_model.trace_length`: length of the channel trace in seconds. Be wary of the size you choose, as it will be loaded in memory.
- `LTE_config.channel_model.correlated_fading`: `true` or `false`. Activates or deactivates the channel time correlation.

- `LTE_config.pregenerated_ff_file`: where to save the channel trace. If the specified file exists, it will be loaded. For the `auto` or unexistent filename cases, a new trace will be generated.
  e.g. `ff_60.0s_2x2_PedB_5.0MHz_5Kmph_20100205_121257`.
- `LTE_config.recalculate_fast_fading`: whether generate the trace even if the file already exists (force a new trace).

J. UE settings

- `LTE_config.UE.receiver_noise_figure`: receiver noise figure in dB. Set to 9 dB [5].
- `LTE_config.UE.thermal_noise_density`: thermal noise density in dBm/Hz.
- `LTE_config.UE_distribution`: how the UEs are generated over the Region Of Interest (ROI). See Section IX for a list of the configuration parameters related to each UE distribution.
- `LTE_config.UE_speed`: speed at which the UEs move. In meters/second.

K. eNodeB settings

- `LTE_config.antenna_gain_pattern`: gain pattern of the antenna attached to each sector. Only valid for `generated` networks. For Capesso-imported networks, these values are not used, as they are read from the cell description files. Available options are:
  - `berger`: $A(\theta) = -\min\left[12\left(\frac{\theta}{70^\circ}\right)^2, 20\,\mathrm{dB}\right]$, $-180 \leq \theta \leq 180$.
  - `TS 36.942`: $A(\theta) = -\min\left[12\left(\frac{\theta}{65^\circ}\right)^2, 20\,\mathrm{dB}\right]$, $-180 \leq \theta \leq 180$ [5].
  - `omnidirectional`: $A(\theta) = 0$.
  - `six-sector`: $A(\theta) = -\min\left[12\left(\frac{\theta}{35^\circ}\right)^2, 23\,\mathrm{dB}\right]$, $-180 \leq \theta \leq 180$.
  - `kathreinTSAntenna`: Antenna pattern to be read from an an antenna pattern file. See also Section X It needs of the following parameters:
    * `LTE_config.site_altiude`: Altiude of site (terrain altitude) [m]
    * `LTE_config.site_height`: Height of site [m].
    * `LTE_config.rx_height`: Receiver height [m].
    * `LTE_config.antenna.mechanical_downtilt`: Antenna mechanical downtilt [].
    * `LTE_config.antenna.electrical_downtilt`: Antenna electrical downtilt [].
    * `LTE_config.antenna.kathrein_antenna_folder`: Folder to scan for the antenna pattern files.
    * `LTE_config.antenna.file_format`: either `msi` or `txap`, depending on the format of your antenna pattern files.
    * `LTE_config.antenna.antenna_type`: The name of the antenna you want to use. e.g. `'742212'`.
    * `LTE_config.antenna.frequency`: The frequency at which the pattern should be used [MHz]. It is not automatically set to the frequency being used because it could happen that all you are interested is the pattern itself, not whether it would correspond with the actual frequency used.
- `LTE_config.max_antenna_gain`: antenna gain, in dB. Recommended values are: 15 dBi (rural area 900 MHz, urban area 2 GHz) and 12 dBi (urban area 900 MHz).

L. Scheduler settings

- `LTE_config.scheduler`: the type of scheduler to use. Supported schedulers are:
  - `round robin`: equally assigns physical resources to all UEs.
  - `best cqi`: each physical resource (RB) is assigned to the UE with the best channel conditions.
  - `prop fair Sun`: proportional fair scheduler, as in ”Reduced-Complexity Proportional Fair Scheduling for OFDMA Systems” by Z. Sun, C. Yin, and G. Yue [12].
  - `FFR`: See Section VI.
- `LTE_config.latency_time_scale`: the simulator keeps track of average UE throughput filtered with an exponential window. This averaged throughput is basically used by the proportional fair scheduler to obtain the average throughput. As in [13], the average throughputs $T_k(t)$ for each user k are updated using an exponentially weighted low-pass filter

$$T_k(t+1) = \begin{cases} \left(1 - \frac{1}{t_c}\right) T_k(t) + \frac{1}{t_c} R_k(t), & k \in k^*, \\ \left(1 - \frac{1}{t_c}\right) T_k(t), & k \notin k^*. \end{cases}$$

  Where k^* is the scheduled UE set and $R_k(t)$ the rate the k-th user got. t_c is the length of the window and is the value stored in `LTE_config.latency_time_scale`.
- `LTE_config.power_allocation`: only `homogeneous` is supported right now.

M. CQI mapper options

- `LTE_config.CQI_mapper.CQI2SNR_method`: `1` to use a less conservative method to map from Channel Quality Indicator (CQI) back to Signal to Noise Ratio (SNR), which uses the value in the middle of the SNR interval corresponding to a CQI instead of the lower boarder value. This value will just be used in connection with quantized CQI feedback.

N. Uplink channel options

- `LTE_config.feedback_channel_delay`: uplink delay in TTIs. When set to 0 TTIs, only the CQI reports experience zero delay. ACK reports have a minimum delay of one TTI.
- `LTE_config.unquantized_CQI_feedback`: when set to `true`, the sent CQI feedback is not rounded, and therefore represents a direct mapping against the post-equalization Signal to Interference and Noise Ratio (SINR). The default mode (CQI is an integer value) is `false`.

O. SINR averaging

since v1.5, just Mutual Information Effective Signal to Interference and Noise Ratio Mapping (MIESM) is supported as an averaging algorithm. Exponential Effective Signal to Interference and Noise Ratio Mapping (EESM) is not supported anymore and is not included in the simulator anymore. See Section XVII for more information.

- `LTE_config.SINR_averaging.algorithm`: what subcarrier averaging algorithm is to be used. For each option, the specific configuration parameters will vary. Possible options are [14]:
  - `MIESM`: use MIESM. Averages on the Mutual Information (MI) domain. See [15] for a more detailed explanation. A `.mat` file containing the Bit Interleaved Coded Modulation (BICM) capacity tables for the relevant modulations and bit mappings must be provided. One is included with the simulator:
    - `LTE_config.SINR_averaging.BICM_capacity_tables`: location of the BICM capacity tables. One is already provided: `data_files/BICM_capacity_tables_20000_realizations.mat`.
    - `LTE_config.SINR_averaging.betas`: MIESM β calibration values. Notice that anyway MIESM is much more robust to calibration errors than EESM, so even without using any calibration at all, the results will not differ much.

P. Saving the results

- `LTE_config.results_folder`: folder where to save the results.
- `LTE_config.results_file`: results filename. `auto` assings a filename automatically.
  eg. `2.00GHz_freq_5.00_bw_200TTIs_20100304_103218_proportional_fair_r230.mat`.

Q. Optional configuration parameters

The following parameters, to allow for backwards-compatibility, are optional. If you do not specify them, they will be set to a default value that will mimic the old behaviour (i.e. as in old versions of the simulator before that parameter was even defined).

- `LTE_config.trace_version`: Defaults to `v1`, which employs precalculated precoding matrices. A new mode was implemented which applies precoding at run-time. The same optimum Precoding Matrix Indicator (PMI) is still precalculated and included, but applied at run-time, so it could be replaced by any other precoding. Of course, given the amount of matrix multiplications involved, this is much slower than `v1`. Related it to `LTE_config.runtime_precoding`.
- `LTE_config.sector_azimuths`: The azimuths of the site sectors. The number of sectors created will be equal to `length(LTE_config.sector_azimuths)`. Defaults to `0:360/3:359`, i.e. `[0 120 240]`.
- `LTE_config.add_femtocells`: Whether to add an extra layer of cells with omnidirectional antennas. This is thought to add a layer of femtocells/small cells, but of course, one could assign this layer a higher transmit power than the macrocells. This extra layer is added on top of the ROI. Defaults to `false`. Extra parameters when using this option are listed in Section VIII.
- `LTE_config.minimum_coupling_loss`: See Section III-G1.
- `LTE_config.always_on`: If no UEs are attached to the eNodeB, when set to `false`, the eNodeB will not radiate power (i.e. not generate interference). See `LTE_config.signaling_ratio` also. Defaults to `true`.
- `LTE_config.manually_set_ROI`: By default the ROI is set to encompass all of the eNodeB sites. If set in this way, you can manually specify the ROI.
- `LTE_config.UE.antenna_gain`: Allows you to specify an antenna gain for the UE's omnidirectional antenna. Thought to used to add an antenna gain loss (e.g. -3 dB).
- `LTE_config.compact_results_file`: Even when using the `LTE_config.delete_ff_trace_at_end` and/or `LTE_config.delete_pathloss_at_end` parameters, the results files generated can be quite big (300+ MB), so it

can be inconvenient when long batch sets of simulations are performed. Not only due to the large storage space required, but also because of the much longer time to read and process the results. At the expense of not having the complex object structure in the saved results, the size of the results files can be reduced by a factor of at least approximately nine. The possible levels of reduction in results file saving are cumulative (3 combines the reduction of `1`, `2` and `3`) are:

  - `1` or `true`: save the results in a struct instead of an object, as well as clearing non-essential data from the objects stored in memory. results structs are created for both cell and UE results. The `UEs` vector is converted to a struct in order to save further space but make it still possible to employ the results Graphical User Interfaces (GUIs). Do note that further level of results file size reduction will make the results GUIs not work.
  - `2`: results are only saved for the UEs. Cell results are not saved. For the UEs, saving of the following variables is skipped:
    - `TB_size`
    - `ACK`
    - `TB_CQI`
    - `CQI_sent`
    - `TB_SINR_dB`
  - `3`: additionally, the following UE traces are not saved:
    - `RI`
    - `nCodewords`
    - `position`
    - `attached_site`
    - `attached_eNodeB`
    - `wideband_SINR`
    - The `UEs` variable is deleted.

- `LTE_config.traffic_map_upscaling`: When specifying traffic maps, this value upscales the traffic map. i.e. it allows you to test a UE density distribution and then upscale it without needing to generate a new map. Defaults to `1` (no upscaling).
- `LTE_config.delete_ff_trace_at_end`: To save some space, if set to `true`, it will delete the microscale fading traces from the results file. Defaults to `false`.
- `LTE_config.delete_pathloss_at_end`: To save some space, if set to `true`, it will delete the pathloss maps from the results file. Defaults to `false`.
- `LTE_config.unquantized_feedback`: The 0-15 CQI range [16] can be set to be transmitted in an unquantized way, so a mapping to SINR can be more precise at the eNodeB side. this is just intended for feedback testing purposes, if a more precise CQI signaling is needed. It defaults to `false`.
- `LTE_config.trace_SINR`: Whether to keep a trace of the subcarrier SINRs. Defaults to `false`. Please note that it will dramatically increase the size of the results file.
- `LTE_config.adaptive_RI`: Takes into account just a specified part of the spectrum if set to `2`. Defaults to `0` (no adaptive Rank Indicator (RI)). See [17] for more details.
- `LTE_config.keep_UEs_still`: Keeps the UEs still regardless of the set UE speed. Channel speed is maintained, though. Useful for snapshot simulations where it is undersirable that a UE would move out of the ROI and be randomly moved to another point in the ROI. Defaults to `false`.
- `LTE_config.default_shown_GUI_cells`: If set, when the result GUIs are called, these cells will be highlighted by default. See `LTE_sim_main_launcher_examples` for an example. Defaults to `[]`.
- `LTE_config.compute_only_UEs_from_this_eNodeBs`: As the SINR distribution of the border ring of eNodebs is not equally distributed as in the center ones, most of the cases it is desirable to take the results just from UEs attached to these central eNodeBs. In this cases, it can be specified to calculate results just from the UEs attached to a set of cells, which results in shorter simulation times. See `LTE_sim_main_launcher_examples` for an example. Defaults to `[]`.
- `LTE_config.reapply_MCL_to_cache_or_planning_tool`: Whether the MCL is re-applied at the end regardless of what is loaded from cache. If set to 'false', this extra step will not be performed when loading from cache or planning tool data. Useful if you would like to apply a set of different Minimum Coupling Losss (MCLs) without the need of generating a pathloss map each time. Defaults to `false`.
- `LTE_config.additional_penetration_loss`: It allows you to set an additional amount of macroscopic pathloss that will be applied, in addition to the set maps. Useful to add an `indoor` or similar pathloss to the UEs. Defaults to `0`. It also accepts the following strings as input, which are equivalent to the following values:
  - `deep indoor`: 23 dB
  - `indoor`: 17 dB
  - `incar`: 7 dB

  - `outdoor`: 0 dB
- `LTE_config.reuse_pregenerated_ff_trace_from_last_run`: if several simulations would need to be run but not enough memory to run them in parallel would be available, it is possible to decrease the simulation time by keeping the trace file in memory. Defaults to `false`. This will give gains if:
  - The simulation is short (few TTIs). So the time needed to load the channel trace (some 100-200 MB) is significant.
  - The simulations use the same channel trace (i.e. same speed).
  - If several traces are used, the execution order of the simulations should group them by trace filename.
- `LTE_config.output_filename_suffix`: Allows you to specify a suffix that will be appended to the end of the simulation results file. Defaults to `''` (empty string).
- `LTE_config.signaling_ratio`: Allows you to set a ratio of power that is to be dedicated non-data channels. It is set to simulate pilots and other control channels that, even if no UE is attached, would continue to be sent. i.e. a ratio of the total power which is always radiating and interfering other neighboring eNodeBs. Must be between 0 and 1. Defaults to `0`.
- Winner II optional parameters: if any of them is set, all of tehse parameters will have to be set. If no antenna parameter set is specified, the settings in Table I will employed.
  - `LTE_config.winner_antenna_params.TX_antenna_polarization`: vector containing for each of the transmit antennas, their polarization in degrees. e.g., for a two-transmit-antenna case `[-45 45]` would set the transmit antennas to a -45° and 45° polarization.
  - `LTE_config.winner_antenna_params.TX_antenna_position_in_lambdas`: distance of the transmit antennas measured in wavelength (λ). The previous example, combined with `[0 0]` would generate a 45/-45 X-Pol.
  - `LTE_config.winner_antenna_params.RX_antenna_polarization`: Analogous to the TX antenna case, this parameter sets the polarization of the receive antennas.
  - `LTE_config.winner_antenna_params.RX_antenna_position_in_lambdas`: Analogous to the TX antenna case, this parameter sets the position of the receive antennas in the antenna array.
- `LTE_config.channel_trace_id`: In cases, you may want that each simulatio would run a different channel trace realization. This parameter will append an id number to the end of the trace filename, so it will allow for multiple traces of the same type to be saved with different filenames.
- `LTE_config.non_parallel_channel_trace`: If you are executing multiple parallel simulations without previously creating a channel trace, you may want to use this option. This will disable the call to `matlabpool open` in the channel trace generation. This avoids the case of `matlabpool open` being called from a worker, which would result in a crash.
- `LTE_config.UE_distribution`: By setting this option, you can change the way the UEs are generated and placed over the ROI. When not specified, it defaults to `constant UEs per cell`, which places a constant number of UEs on each cell (eNodeB). When comparing scenarios with an overlay of small cells, this would make the total number of UEs proportional to the number of eNodeBs, which may not be the desirable case. See Section IX for details on how to specify other UE distributions.

IV. Implementation issues

A. How to read the pathloss maps

When reading the pathloss maps, the convention shown in Figure 1 has been used.

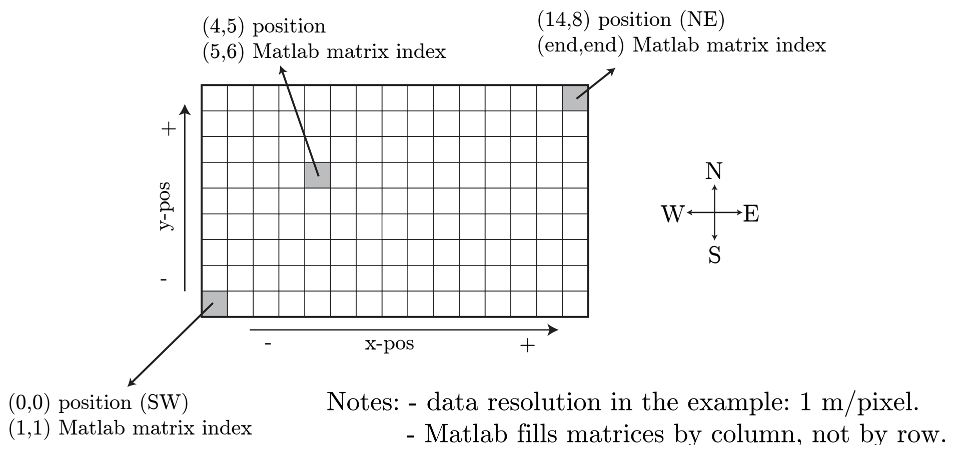

(4,5) position
(5,6) Matlab matrix index
(14,8) position (NE)
(end,end) Matlab matrix index
+
y-pos
-
N
W
E
S
-
x-pos
+
(0,0) position (SW)
(1,1) Matlab matrix index
Notes: - data resolution in the example: 1 m/pixel.
- Matlab fills matrices by column, not by row.

Fig. 1. Coordinate convention used in the LTE SL Simulator.

It is because of this coordinate convention that in order to correctly display a pathloss map (without flipping it and with correct axes ticks) the following command is used:

```
imagesc([min_roi_x max_roi_x],[min_roi_y max_roi_y],pathloss_map);
set(gca,'YDir','normal');
```

B. Antenna azimuth

In the simulator, do note that an azimuth of 0°points towards 12 o'clock, with the azimuth value increasing clockwise, thus a 180°azimuth would point to 6 o'clock.

C. Remote Radio Head (RRH)

The current implementation allows to employ RRHs with a predefined number of antennas per node. A demo is provided in `sim_main_launcher_files/LTE_sim_main_launcher_demo_RRH`. The implementation requires run-time precoding, i.e., the setting `LTE_config.trace_version = 'v2'`, since the channel matrices are generated on the fly by stacking the single channels.

D. Block Error Ratio curves and CQI mapping

The Block Error Ratio (BLER) curve data files can be supplied to the simulator in the following forms:

- Long Term Evolution (LTE) link-level simulator [18] `.mat` results file.
- `.mat` file containing the following vector variables of equal length (needs less disk space):
  - `SNR`: SNR values.
  - `BLER`: BLER values.

When loaded, the CQI tables are used to generate the SNR-toCQI mapping, which is shown in Figure 2. The 15 CQI BLER curves are loaded (left), and from their 10 % BLER points, the (basically linear) mapping between for the calculation of the reported CQI is obtained (right).

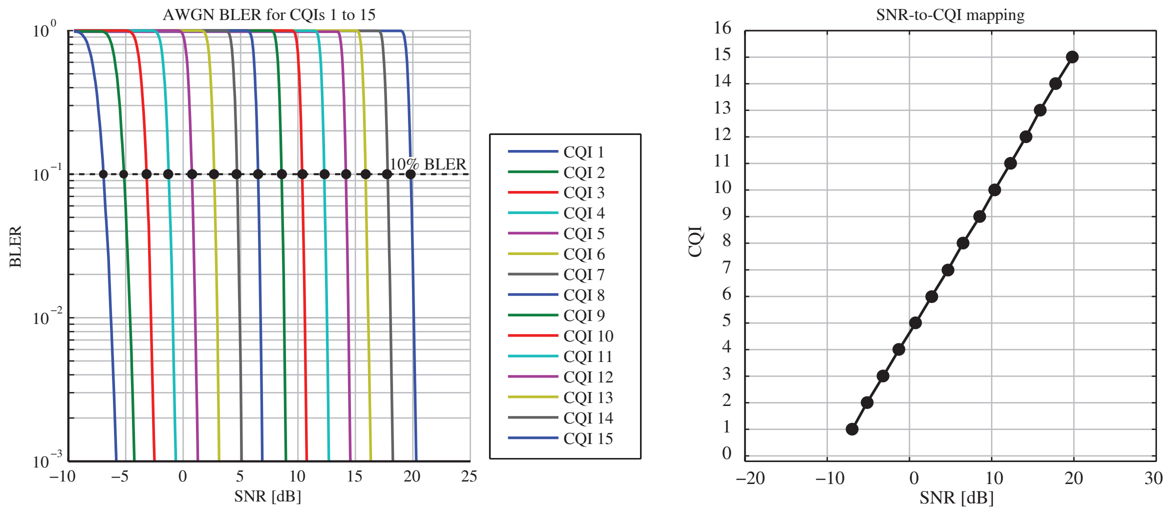
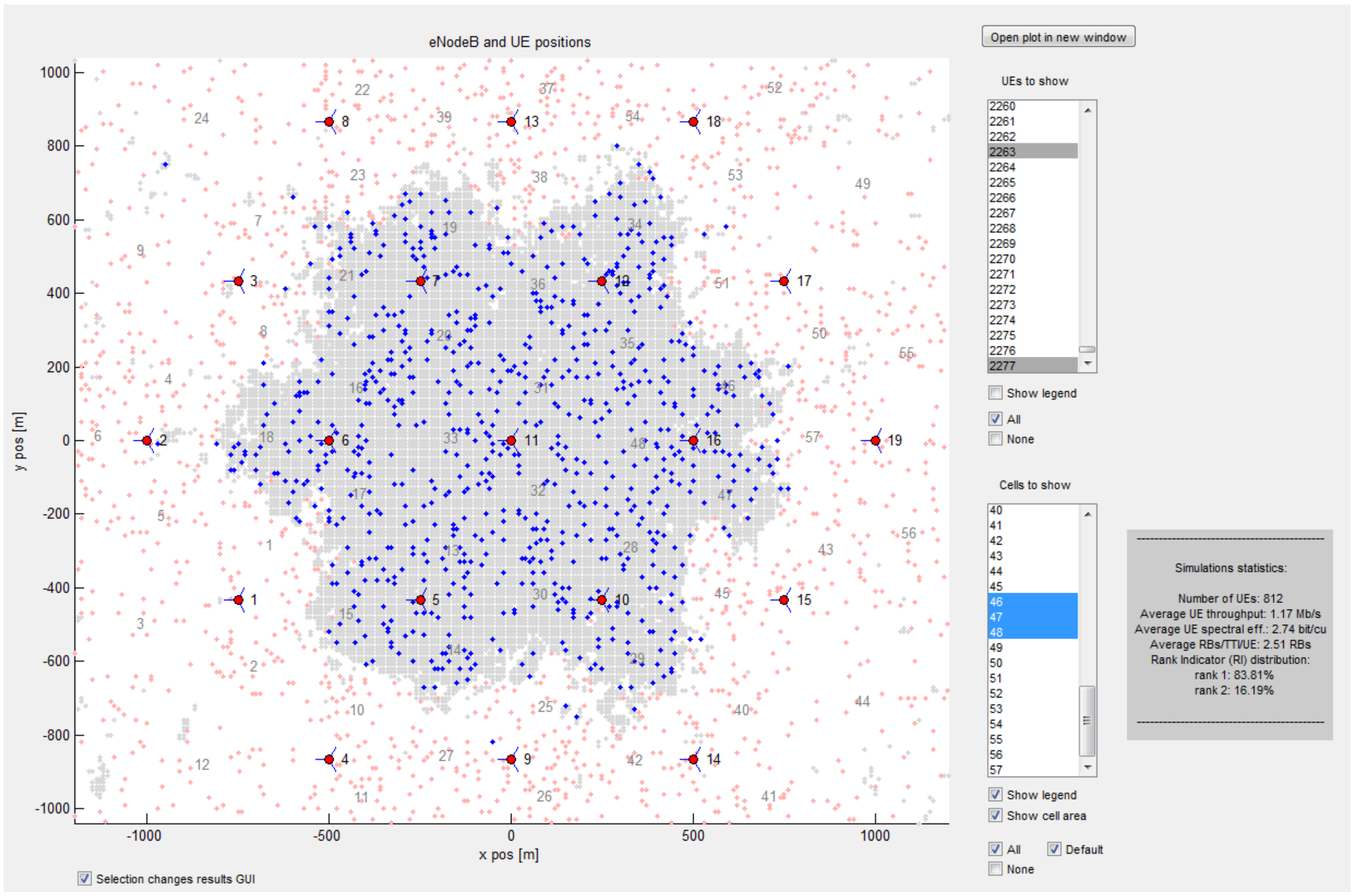


Fig. 2. SNR-to-CQI mapping. Left: CQI BLER curves. Right: CQI mapping obtained from the 10 % BLER points.

V. Plotting results

See the `LTE_sim_main_examples` script for a reference on how to call the functions plotting the simulation results.

A. *eNodeB and UE position*



Simulations statistics:

Number of UEs: 812
Average UE throughput: 1.17 Mb/s
Average UE spectral eff.: 2.74 bit/cu
Average RBs/TTI/UE: 2.51 RBs
Rank Indicator (RI) distribution:
rank 1: 83.81%
rank 2: 16.19%

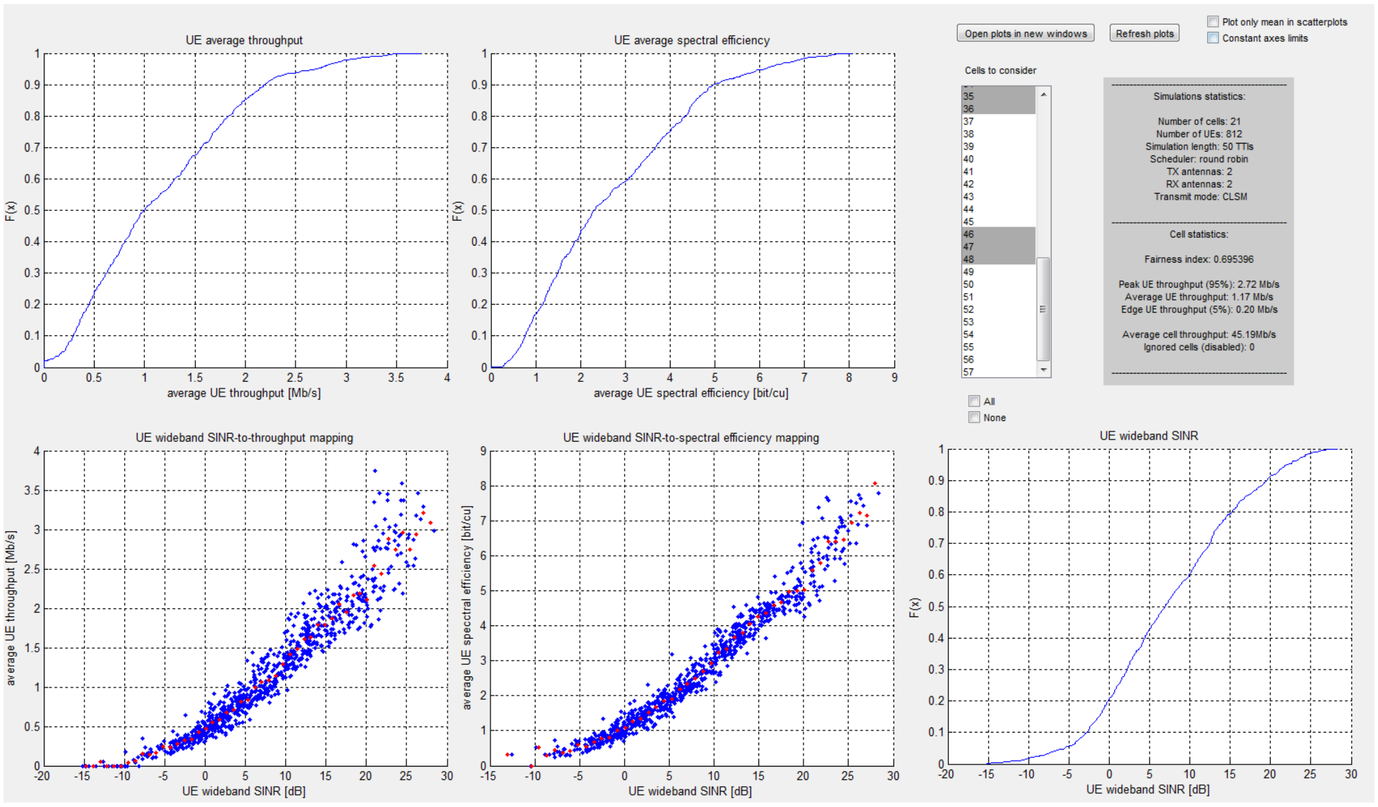
Fig. 3. Positions GUI. Shaded via the selction of the appropriate cells on the cell list: approximate cell area corresponding to the center eNodeBs. The UEs attached to the selected eNodeBs are shown black. Shaded are the rest of the UEs.

`LTE_GUI_show_UEs_and_cells` (Figure 3): GUI that shows the positions of all UEs, eNodeBs (cells) and eNodeB sites, as well as related information:

- Selected cells. When selecting one or more cells from the menu, the UEs associated to these cells are automatically selected. The UE selection can be changed a posterior (e.g. to locate a specific UE on the map).
- If selected, the area from selected cells is shadowed. All UEs are shown on the map. The ones not pertaining to any of the selected cells are showed shaded.
- If some UEs in the simulation are deactivated, these UEs will appear red on the map.
- If the simulation happens to use a Fractional Frequency Reuse (FFR) scheduler, Full Reuse (FR) UEs are shown as dots. Partial Reuse (PR) UEs are shown as crosses.
- If selected, a legend showing the eNodeB index and UE index of all seleted eNodeBs/UEs is displayed. The eNodeB index is displayed grey approximately in the middle of the cell area. The UE index is displayed next to the last position of each UE. The site id is always shown next to the site.
- For convenience, and if called as in `LTE_sim_main_examples`, the selection on this GUI will automatically change the cell selection on the second GUI (Section V-B).
- The grey text box contains:
  - Number of plotted UEs.
  - Average UE thorughput.
  - Average UE spectral efficiency. While for a round robin scheduler throughput and spectral efficiency will basically look the same, for any other scheduler in which the assignment of Physical (PHY) resources is not random and homogeneous, these would look different.

  - Aveage scheduled RBs per UE and TTI.
  - Distribution of the RI feedback.

B. *Throughput and aggregate results*


Open plots in new windows
Refresh plots
Plot only mean in scatterplots
Constant axes limits
Cells to consider
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
All
None
UE average throughput
F(x)
average UE throughput [Mb/s]
UE average spectral efficiency
average UE spectral efficiency [bit/cu]
UE wideband SINR-to-throughput mapping
average UE throughput [Mb/s]
UE wideband SINR [dB]
UE wideband SINR-to-spectral efficiency mapping
average UE spectral efficiency [bit/cu]
UE wideband SINR
UE wideband SINR [dB]

```
----------------------------------------
Simulations statistics:

Number of cells: 21
Number of UEs: 812
Simulation length: 50 TTIs
Scheduler: round robin
TX antennas: 2
RX antennas: 2
Transmit mode: CLSM

----------------------------------------
Cell statistics:

Fairness index: 0.695396

Peak UE throughput (95%): 2.72 Mb/s
Average UE throughput: 1.17 Mb/s
Edge UE throughput (5%): 0.20 Mb/s

Average cell throughput: 45.19Mb/s
Ignored cells (disabled): 0

----------------------------------------
```

Fig. 4. Aggregate results GUI. The same eNodeBs as in Figure 3 are selected. Plots show the throughput ECDF for the average UE throughput (upper-left), spectral efficiency (upper-right), wideband (lower-left) SINR and the mapping between between the wideband SINR and the average throughput for each UE. The results are computed from the UEs attached to the selected eNodeBs. Some overall statistics are shown on the grey text box.

`LTE_GUI_show_aggregate_results` (Figure 4): GUI that shows aggregate UE results, as well as some cell-related statistics. For the UE-related results, the UEs from which the results are obtained are the ones pertaining to any of the selected cells. Deactivated UEs (i.e. NaN values) are ignored. If the cell selection is linked to the first GUI (Section V-A), before the changes in a selction are shown, the "Refresh plots" button has to be pressed:

- Empirical Cumulative Distribution Function (CDF) (ECDF) of the UE average throughput.
- ECDF of the UE average spectral efficiency.
- ECDF of the UE wideband SINR.
- Scatterplot showing for each UE in the set the mapping between the wideband SINR and the throughput. Since many points could be overlapping, there is the option of showing a binned (over wideband SINR) mean throughput mapping (in red).
- Scatterplot showing for each UE in the set the mapping between the wideband SINR and the spectral efficiency. Since many points could be overlapping, there is the option of showing a binned (over wideband SINR) mean spectral efficiency (in red).
- The grey text box contains:
  - Number of cells (eNodeBs) from which these results are plotted.
  - Number of UEs pertaining to these eNodeBs. The UE results are obtained from these UEs.
  - Length of the simulation in TTIs.
  - Type of scheduler being used.
  - Number of TX and RX antennas, as well as the transmit mode, as specified in [19] (single TX antenna, TxD, OLSM, or CLSM).
  - Fairness index [20], as calculated from the UE average throughput values.
  - Obtained from the UE average throughput plot, the 95% (peak), average, and 5% (edge) throughput values.
  - Average cell throughput.

- If some cells would contain inactive UEs, thus making its cell throughput not valid, they are ignored for the computation of the previous value. The number of ignored eNodeBs is shown here.

VI. Fractional Frequency Reuse (FFR)

FFR is implemented in the LTE SL Simulator as a new kind of scheduler (`FFR` scheduler). It allows you to specify a scheduler for the FR and PR parts independently.

When selecting the `FFR` scheduler, `LTE_config.FFR_active` is automatically set to `true`. Subsequently, you have to define each of the child schedulers that will hang from the FFR scheduler. So both

- `LTE_config.scheduler_params.FR_scheduler` and
- `LTE_config.scheduler_params.PR_scheduler`

have to be specified. An example for an FFR simulation in which both schedulers would be round robin would be

```
LTE_config.scheduler                               = 'FFR';
LTE_config.scheduler_params.FR_scheduler.scheduler = 'round robin';
LTE_config.scheduler_params.PR_scheduler.scheduler = 'round robin';
```

while if proportional fair scheduling with an exponential averaging window of length 25 would be desired, it would be

```
LTE_config.scheduler                               = 'FFR';
LTE_config.scheduler_params.FR_scheduler.scheduler = 'prop fair Sun';
LTE_config.scheduler_params.FR_scheduler.av_window = 25;
LTE_config.scheduler_params.PR_scheduler.scheduler = 'prop fair Sun';
LTE_config.scheduler_params.PR_scheduler.av_window = 25;
```

The FR and PR scheduler can of course be also different schedulers. No restriction is placed on the type of scheduler that can be attached to the FFR scheduler.

Any parameters specified in the `FR_scheduler` or `PR_scheduler` structs is copied to the scheduler initialization, so any scheduler could be used in conjunction with FFR. FFR requires of the following two additional parameters:

- `LTE_config.FFR_params.beta_FR`: in the $(0, 1]$ interval, it specifies the ratio of the total system bandwidth which is used for the FR part. A β_{FR} value of 1 sets the whole bandwidth for the FR part (as in reuse-1). Note that since reuse-3 is assumed for the PR part, each of the PR parts will use $1/3$ of the remaining bandwidth (ratio of $(1 - \beta_m athrmFR)$). When simulating, (obviously) only an integer-valued number of RBs can be scheduled to the FR/PR parts. Which means that for a 20 MHz bandwidth (100 RB), the minimum value of β_{FR} is 0.01, as 100 is not divisible by 3 (99 is).
- `LTE_config.FFR_params.SINR_threshold_value`: taking the SINR values from the pathloss maps (not to be confused with post-equalization SINR), at simulation begin at UE assignment to either the PR or the FR zone is performed. If the SINR is bigger than the threshold, the UE is set to be a FR UE, and viceversa.

VII. Coordinated Multipoint (CoMP)

The object `CoMP_site` is introduced, which holds a set of eNodeBs. It is configured by `LTE_config.CoMP_configuration`:

- `trivial` Is just in there for comparison, it equals NO CoMP, as each eNodeB is a different site.
- `global` All eNodeBs are in the same site, all are potentially cooperating.
- `others` Are not implemented yet

In the System Level simulator, CoMP is relevant for *receiver*, and *scheduling*. Since no particular receiver structure is implemented yet, the focus lies on scheduling. The `CoMP_site` employs the scheduler `CoMP`. It enables scheduling across multiple eNodeBs and is configured by `eNodeB.CoMP_handles_pre_and_postprocessing`:

- `false`: If this is set to false, the eNodeB calls the `LTE_scheduler` in the default manner.
- `true`: If the variable is set to true, the `CoMP` scheduler is called directly by the eNodeB. It has to take care of all preprocessing, scheduling and postprocessing. It might also do the pre- and the postprocessing, and call a regular LTE-scheduler for the scheduling. This mode provides freedoms in scheduling, however it requires to replicate all needed functionalities from the scheduling process.
  At first call, the `CoMP` scheduler calculates the `RB_grid` for all eNodeBs, and subsequently just provides the stored grid.

The struct `cooperation_rules` in `+CoMP.CoMP_site` stores the policy of cooperation. E.g., a `dont_schedule` can indicate the strongest interfering eNodeB on a given RB, which should therefore restrain its transmission on this resource. Consider, e.g., *Round Robin coordinated scheduling*.The `LTE_scheduler` is set to `CoMP`, and `CoMP_handles_pre_and_postprocessi` is set to false. Therefore, the `CoMP` scheduler requests RB grids from the `LTE_scheduler`. According to the cooperation rules, it then mutes certain RBs based on a wideband Signal to Interference Ratio (SIR) threshold.

VIII. Femtocells and small cells

The LTE SL Simulator allows for an additional layer of eNodebs to be placed over the standard eNodeB grid. This would be typically employed to add a layer of femtocells or small cells to the eNodeB layout. The femtocell layout can be added via the optional `LTE_config.add_femtocells` parameter, which when set to `true`, enables the following options to be employed:

- `LTE_config.femtocells_config.spatial_distribution`: specifies the spatial distribution of the femtocells. Possible values are:
  - `homogeneous density`: homogeneously spreads femtocells over the ROI with a given density in femtocells/km^2. The following parameters need to be specified:
    - `LTE_config.femtocells_config.femtocells_per_km2`: The femtocells are uniformly spread across the ROI with the specified density in average femtocells/km^2.
- `LTE_config.femtocells_config.tx_power_W`: transmit power of each of the femtocells in Watts.
- `LTE_config.femtocells_config.macroscopic_pathloss_model`: Which pathloss model to be used by the femtocells. Same options available as for the macro eNodeBs. If this parameter is not specified, the same values as in `LTE_config.macroscopic_pathloss_model` is employed.

IX. UE distributions

The `LTE_config.UE_distribution` allows for several UE distributions to be employed. Below are the allowed types of UE distribution, as well as their related parameters:

- `constant UEs per cell`: places a constant number of UEs per cell.
  - `LTE_config.UE_per_eNodeB`: number of UEs per cell.
- `LTE_config.radial`: places around each eNodeB site a series of concentric rings with UEs.
  - `LTE_config.UE_distribution_radii`: a vector specifying the radius of each of the UE rings around the eNodeB sites.
  - `LTE_config.UE_distribution_nUEs`: either a scalar specifying the number of UEs per ring or a vector specifying for each ring the number of UEs.
- `traffic map`: load a traffic map from a Capesso^{TM}file. See Section XI also.
  - `LTE_config.traffic_map_config.udtm_folder` : folder where the traffic map is located.
  - `LTE_config.traffic_map_config.udtm_filename`: name of the traffic map file.
  - `LTE_config.traffic_map_config.udtm_environment`: possible options are:

X. 3D antenna radiation patterns

Starting on v1.3r427, it is possible to use 3D antenna radiation patterns, in addition to the 2D patterns suggested in the standard [5]. The antenna files have been provided by KATHREIN-Werke KG, and represent measured antenna radiation patterns from commercial antenna models. The interpolation to a 3D radiation pattern has been done by the common component sum method [21]. Next versions will support importing pathloss maps from a network planning tool such as Atoll/Capesso and cross the pathloss data with 3D antenna radiation patterns, but for now, just the antenna pattern is supported. To calculate the vertical angle, a flat terrain surface is assumed.

Parameters that can be configured when using a 3D antenna radiation pattern are (these parameters are applied to each transmitter):

- `LTE_config.site_altiude`: altitude of the site. Set to a default of 0 m.
- `LTE_config.site_height`: height of the antenna pole. Set to a default of 20 m.
- `LTE_config.rx_height`: height of the receiver. Set to a default of 1.5 m.
- `LTE_config.antenna.mechanical_downtilt`: mechanical downtilt applied to the antenna. The effect of the mechanical downtilt is calculated.
- `LTE_config.antenna.electrical_downtilt`: electrical downtilt applied to the antenna. For different downtilts a suitable antenna pattern file must be present, so just integer values are supported.
- `LTE_config.antenna.kathrein_antenna_folder`: where the simulator should look fo rthe antenna files. Defaulted to `'./data_files/KATHREIN_antenna_files/msi'`.
- `LTE_config.antenna.file_format`: the simulator is capable of importing both `.msi` and `.txap` files. This sets which files are to be imported.
- `LTE_config.antenna.antenna_type`: the specific antenna type you want to use.
- `LTE_config.antenna.frequency`: sets the frequency at which the antenna is operating. This should be the same as `LTE_config.bandwidth`, but it is configured separately in case you would like to check a particular frequency pattern for an arbitrary frequency.

XI. Importing a network topology from Capesso^{TM}data

Since v1.4r570, it is possible to import a network layout from pathloss data as obtained from Symena's^{TM}Capesso^{TM}tool[1] automatic cell planning tool. Data from Forsk's^{TM}Atoll^{TM}[2] should also be importable, but the feature is not tested.

The data contained in the data files under `data_files/CapessoExample` contains a hexagonal cell layout stored in the same format as the network cell planning would, including eNodeB configuration.

The code may also serve as a gide to anyone interested in importing data from another network source. An example simulation can be run by means of the `LTE_sim_main__launcher_capesso.m` script.

You can find below the extra parameters specified in `LTE_load_params_example_capesso.m`. The data files contained in the `CapessoExample` folder contain a Digital Terrain Map (DTM) of the ROI (`dtm` folder), specifying the elevation of the terrain, the omnidirectional pathloss from each site (`.par` and `.los` files), and the eNodeB configuration (`exampleCluster_Cell.txt`, `exampleCluster_Sites.txt`, and `exampleCluster_Transmitter.txt`).

XII. Using the Winner Phase II channel model reference implementation

Starting with v.1.4r550, it is possible to use channels generated with the publichly-available Matlab implementation of the WINNER Phase II Channel Model [11]. Since the code is distributed under the GNU GPL, its files are not included in the simulator release. In order to use to be able to use it, you will have to download it yourself. For this, go to the WINNER Phase II Model website, download the `WIM2_3D_ant_ver064_220908.zip` file and unzip the `.mat` files in the `./Winner Channel Model` folder.

Unless the `LTE_config.winner_antenna_params.TX_antenna_polarization` optional parameters are set (see Section III-Q), the default antenna parameters will be loaded, which correspond to the values listed in Table I.

TABLE I
Default MIMO antenna settings employed with the Winner Phase II channel model.

	Number of antennas	Antenna	Position	Polarization	
TX	2	1	$0\,\lambda$	-45	X-Pol
		2	$0\,\lambda$	45	
	4	1	$0\,\lambda$	-45	XX-Pol
		2	$0\,\lambda$	45	
		3	λ	-45	
		4	λ	45	
RX	2	1	$0\,\lambda$	-45	X-Pol
		2	$0\,\lambda$	45	
	4	1	$-1\,\lambda$	-45	XX-Pol
		2	$-1\,\lambda$	45	
		3	λ	-45	
		4	λ	45	

XIII. A note on the use of the CVX Matlab toolbox

For some schedulers, it may be possible that the "CVX: Matlab Software for Disciplined Convex Programming" convex optimization toolbox may be needed [22]. It is available under http://cvxr.com/cvx/ under the GNU GPL 2.0 license. For the cases it may be needed, just download it and place its contents in the `cvx` folder.

XIV. Employing UE traces for simulating

Starting with v.1.7, it is possible to define a simulation scenario via UE traces instead of a pathloss map. The traces that need to be supplied, its format, and the necessary configuration parameters are described in this section.

The traces describe, for the simulated UEs, the the pathloss between the UE and N cells, which include its serving cell (interpreted from the traces as the cell with the lowest pathloss), and $N-1$ interfering cells.

The trace has the following format:

`time [s];UE_id; cell_id 1; pathloss 1; cell_id 2; pathloss 2;[...];cell_id N; pathloss N`

The beginning of an example trace extract can be found below:

```
0.0; 12; 137; 118.54; 207; 129.99; 129; 131.22; 130; 131.68; 186; 131.87; 157; 135.42; 133; 138.75; 19; 138.85; 179; 139.28; 91; 139.53
0.0; 13; 130; 122.43; 137; 123.03; 129; 134.23; 35; 135.94; 176; 136.63; 133; 136.65; 109; 138.36; 207; 138.38; 202; 138.94; 91; 139.11
```

[1]Symena and Capesso are the trademarks or registered trademarks of Symena Software & Consulting Gmbh . There is no relation between the Vienna LTE simulators and Symena

[2]Forsk and Atoll are the trademarks or registered trademarks of Forsk. There is no relation between the Vienna LTE simulators and Forsk

```
0.0; 14; 137; 118.54; 207; 129.99; 129; 131.22; 130; 131.68; 186; 131.87; 157; 135.42; 133; 138.75; 19; 138.85; 179; 139.28; 91; 139.53
0.0; 15; 137; 106.06; 130; 114.96; 129; 117.34; 133; 119.47; 157; 122.32; 207; 125.72; 19; 128.25; 176; 128.26; 199; 129.34; 109; 129.94
0.0; 16; 137; 118.54; 207; 129.99; 129; 131.22; 130; 131.68; 186; 131.87; 157; 135.42; 133; 138.75; 19; 138.85; 179; 139.28; 91; 139.53
0.1; 11; 137; 118.54; 207; 129.99; 129; 131.22; 130; 131.68; 186; 131.87; 157; 135.42; 133; 138.75; 19; 138.85; 179; 139.28; 91; 139.53
0.1; 12; 130; 122.43; 137; 123.03; 129; 134.23; 35; 135.94; 176; 136.63; 133; 136.65; 109; 138.36; 207; 138.38; 202; 138.94; 91; 139.11
0.1; 13; 137; 118.54; 207; 129.99; 129; 131.22; 130; 131.68; 186; 131.87; 157; 135.42; 133; 138.75; 19; 138.85; 179; 139.28; 91; 139.53
0.1; 14; 137; 106.06; 130; 114.96; 129; 117.34; 133; 119.47; 157; 122.32; 207; 125.72; 19; 128.25; 176; 128.26; 199; 129.34; 109; 129.94
0.1; 15; 137; 118.54; 207; 129.99; 129; 131.22; 130; 131.68; 186; 131.87; 157; 135.42; 133; 138.75; 19; 138.85; 179; 139.28; 91; 139.53
0.1; 16; 137; 118.54; 208; 90.99; 129; 131.22; 130; 131.68; 186; 131.87; 157; 135.42; 133; 138.75; 19; 138.85; 179; 139.28; 91; 139.53
```

In this example, the trace details, for each step of 0.1 seconds, which UE are in the system and the pathlosses they see from the ten strongest base stations.

Not all UEs need to be present at each step. For instance, UE 11 appears after 0.1 s. In the same way, any given UE/s can vanish from the simulation (they are then inactive) at any given time.

In this mode, the pathloss is read from the traces instead of from pathloss maps, with the necessary number of dummy `eNodeB` objects is created so as to ensure all of the eNodeBs mentioned in the trace exist in the simulation.

To employ traces, set `LTE_config.trace_simulation_mode` to `true`. The fllowing additional parameters are employed (an example configuration file is included in the `+simulation_config` folder):

- `LTE_config.trace_filename`: the zip file containing the text file with the traces or directly the text file. See the included trace file in `./data_files/UE_pathloss_trace` for an example of how the naming convention of the files is.
- `LTE_config.eNodeB_tx_power`: as only the pathlosses are specified, the TX power for the eNodeBs needs to be specified. The possibility to specify the eNodeB transmit power (or an RSRP-equivalent metric) in the traces may be added in future releases.
- `LTE_config.TTI_per_trace_step`: To specify how many TTIs each step in the trace represents. Can be set to lower numbers to speed-up simulation time.

As the reading of large traces may be slow, a caching of the UE traces based on a hash of the cache file name is provided. See the `LTE_sim_main_launcher_trace` file, included with the simulator release, for an example of how this feature is used.

XV. MATLAB AUTOMATIC CODE GENERATION (CODEGEN)

As mentioned in the changelog, in v.1.7 we employed for the first time the automatic code generation capabilities of the MATLAB coder (a.k.a. codegen) [23]. The following subsection/s detail the portions of code that are accelerated by means of automatically-generated MEX functions, as well as the parameters employed when creating the included MEX files (only Win64 versions are included). Although the `LTE_aux_mex_files` script contains the commands necessary for compilation calling the `codegen` compiler, the subsections below detail the input parameters for each of the (time-consuming) functions that can be compiled to MEX functions.

For architectures other than Win64 or if for some reason the MEX file would not run on your MATLAB setup, you can then recompile the following function/s by calling the `LTE_aux_mex_files` script[3]:

Note: for all files, the following configuration has been employed:

```
mex_config = coder.MexCodeConfig;
mex_config.ExtrinsicCalls            = false;
mex_config.SaturateOnIntegerOverflow = false;
mex_config.IntegrityChecks           = false;
mex_config.ResponsivenessChecks      = false;
```

A. Shadow fading generation

- Function name: `channel_gain_wrappers.shadowFadingMapClaussen.spatiallyCorrelateMap`
- Function folder: `./+channel_gain_wrappers/+shadowFadingMapClaussen`
- Employ the following parameters
  - Input variables (variable name and type)
    ```
    n_neighbors         double(1 x 1)
    offsets_neighbors   double(:inf x 2)
    L_tilde             double(:inf x :inf)
    a_n_matrix          double(:inf x :inf x :inf)
    lambda_n_T          double(1 x :inf)
    ```
  - Output file: `spatiallyCorrelatedMap_mex`

[3]Note that we cannot assist you in setting up the MATLAB coder. If you have inquiries in this respect, you can forward them to the customer service of the MathWorks or ask in the many available forums such as StaackOverflow (http://stackoverflow.com/questions/tagged/matlab).

B. Calculation of interferer terms according to the precoders applied by the interfering eNodeBs and the interfering channel(s)

- Function name: `network_elements.UE.calculate_per_layer_interference_power`
- Function folder: `./+network_elements/+UE/`
- Employ the following parameters
  - Input variables (variable name and type)

`P`	`complex(double(:inf x :inf x :inf))`
`H_i`	`complex(double(:inf x :inf x :inf x :inf)`
`precoding_codebook`	Precoder `struct` vector (see `LTE_aux_mex_files` and/or note below)

  - Output file: `calculate_per_layer_interference_power_mex`

Note: The `struct` defining the precoders can be generated with the following code snippet:

```
interfering_precoders = ...
  coder.newtype('struct', struct('W',coder.typeof(complex(0), [inf inf])), [inf inf]);
```

Also note that as the complexity of the precoder variable is fixed, even if it is not a complex number, for use with the `codegen` version, the precoder should always be (if needed) converted to a complex variable.

C. Receiver filter calculation according to the precoder

- Function name: `network_elements.UE.calculate_effective_channel_and_receiver`
- Function folder: `./+network_elements/+UE/`
- Employ the following parameters
  - Input variables (variable name and type)

`H`	`complex(double(:inf x :inf x :inf))`
`W`	`complex(double(:inf x :inf x :inf))`

  - Output file: `calculate_effective_channel_and_receiver_mex`

D. Capacity calculation for each precoder in a codebook for feedback calculation

- Function name: `feedback_calculation.codebook_capacity.non_sparse`
- Function folder: `./+feedback_calculation/+codebook_capacity/`
- Employ the following parameters
  - Input variables (variable name and type)

`H_t_all`	`complex(double(:inf x :inf x :inf))`
`W_all`	`complex(double(:inf x :inf x :inf))`
`receiver`	`char(1 x :inf)`

  - Output file: `non_sparse_mex`

XVI. Examples and Reproducible results

One of the main points of the simulator is to allow you to reproduce and review our results, as well as the algorithms that produce them. For each of the following scenarios, reproducible results are included with the simulator. Only results which are obtained directly from the released version of the simulator are included here. For other papers, altered or trimmed versions may have been used. In such cases, those results are not included here, but in a separate file, so check the author's site for the link to those files

A. Comparison with link level results: simulation over SNR range

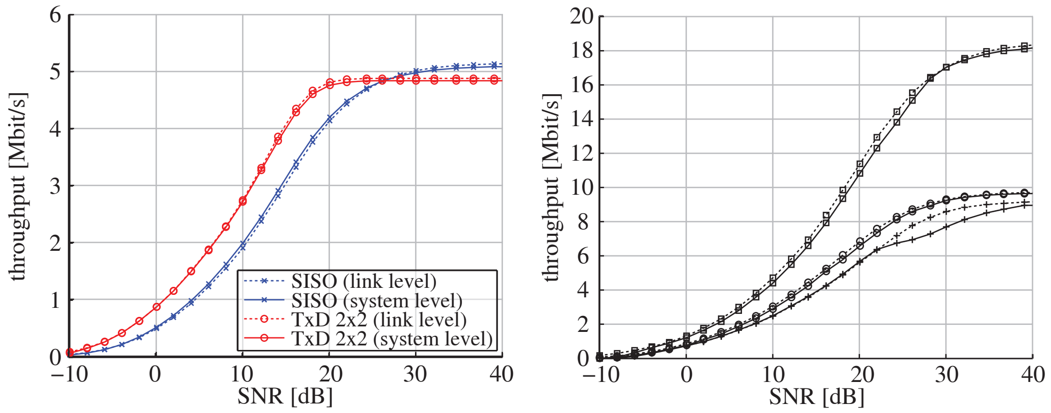


Fig. 5. Left: SISO and 2×2 TxD. Right: OLSM.

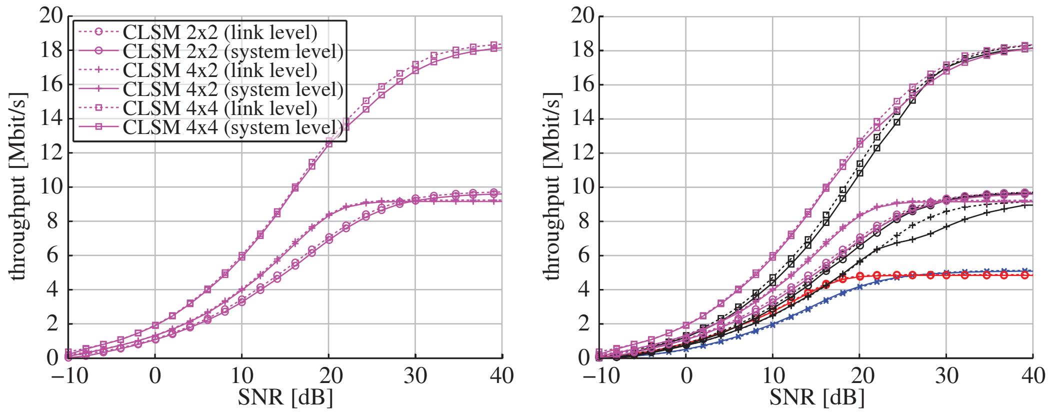


Fig. 6. LefT: CLSM. Right: All antenna configurations.

B. UE throughput scheduler comparison

Since the publication of the paper presenting this simulator in 2010 [2], many changes, improvements, bugfixes, new features and even structural changes have made the LTE SL Simulator evolve. Hence, the decision to delete the original script files for the plots shown in the original paper. The plots would now give a wrong impression of the level of maturity of this tool, so a new version of equivalent plots is now included since.

These plots are analogous to the ones shown in [2], and show how to obtain UE throughput CDF curves, as well as a comparison in terms of mean, edge, and peak throughput, as well as fairness [20].

To reproduce these plots, you will need to run the MATLAB script in the `reproducibility/VTC_v2` folder. For your convenience, the generated results files are also included in the`reproducibility/included_simulation_results` subfolder (`CLSM_2x2_RR.mat`, `CLSM_2x2_BESTCQI.mat`, and `CLSM_2x2_PROPFAIR.mat`), so if you are only interested in the plots, you can just run part of the script skipping the calls to the simulator.

The resulting plots are shown in Figure 7, and depict a comparison between round robin, proportional fair, and best CQI scheduling algorithms for a 2×2 CLSM LTE setup.

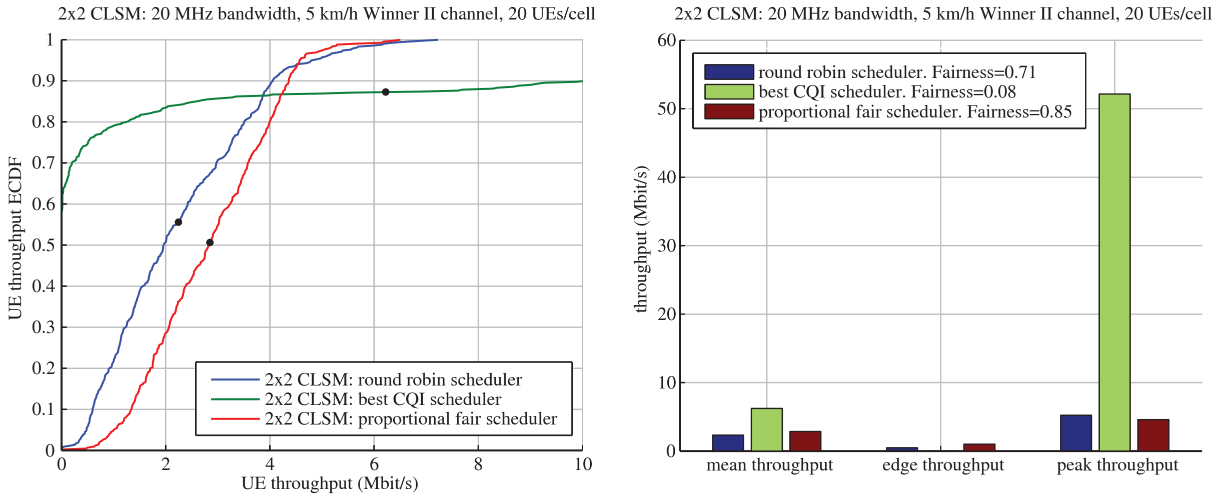


Fig. 7. UE throughput comparison for different types of scheduling: round robin, proportional fair, and best CQI. Left: UE throughput empirical CDF for each of the scheduler setups. The mean value is marked in the CDF as a black dot. Right: Scheduler comparison in terms of mean, edge, and peak UE throughput. Fairness is shown in the legend.

C. *Closed Loop Spatial Multiplexing throughput comparison for several antenna configurations*

To reproduce these plots, you will need to run the MATLAB script in the `reproducibility/CLSM_throughput_comparison` folder. For your convenience, the generated results files are also included in the `reproducibility/included_simulation_results` subfolder (`CLSM_2x2_RR.mat`, `CLSM_4x2_RR.mat`, and `CLSM_4x4_RR.mat`), so if you are only interested in the plots, you can just run part of the script skipping the calls to the simulator.

The resulting plots are shown in Figure 8, and depict a comparison between CLSM with a 2×2, 4×2, and 4×4 antenna setup.

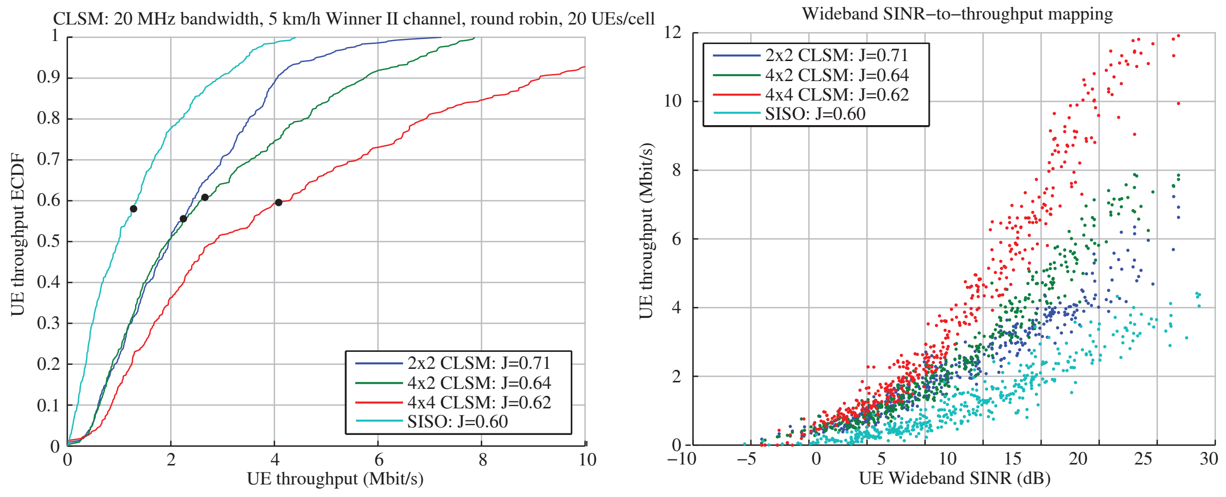


Fig. 8. UE throughput comparison for CLSM with a 2×2, 4×2, and 4×4 antenna configuration. Left: UE throughput empirical CDF for each of the scheduler setups. The mean value is marked in the CDF as a black dot. Right: Mapping between the UE wideband SINR and the throughput achieved by each UE. Fairness is shown in the legend.

D. Adjusting antenna separation in the Winner II channel model

This simulation show how the antenna separation in the Winner II channel trace can be adjusted. The simulation results can be reproduced by means of the `LTE_sim_main_launcher_antenna_separation` script in the `reproducibility/antenna_separation` folder.

Table II lists the antenna configurations input into the Winner II channel model to generate the channel traces which, afterwards, were employed during the simulation. Please do note that the channel model was not implemented by us, so we are just providing the means to configure certain parameters of the channel trace generation and them applying it to the simulation/s. So the results presented here are, in a sense, as precise as the model employed.

TABLE II
ANTENNA CONFIGURATIONS FOR EACH OF THE SIMULATIONS IN LTE_SIM_MAIN_LAUNCHER_ANTENNA_SEPARATION. SEVERAL TRANSMIT ANTENNA CONFIGURATIONS OF THE WINNER II CHANNEL MODEL ARE TESTED. AT THE RECEIVER SIDE, CROSS-POLARIZED ANTENNAS WERE ALWAYS USED.

Setting	N_{TX}	Polarization	Position (x-axis)	TX Label	N_{RX}	RX conf.
1	4	[-45 45 -45 45]	$[\ -\lambda\ -\lambda\ \lambda\ \lambda\ ]$	XX	4	XX
2		[-45 45 -45 45]	$[\ -5\lambda\ -5\lambda\ 5\lambda\ 5\lambda\ ]$	X-X		
3		[90 90 90 90]	$[\ -3\lambda\ -\lambda\ \lambda\ 3\lambda\ ]$	||||		
4		[90 90 90 90]	$[\ -15\lambda\ -5\lambda\ 5\lambda\ 15\lambda\ ]$	|−|−|−|		
5	2	[-45 45]	[0 0]	X	2	X
6		[90 90]	$[\ -\lambda\ \lambda\ ]$	||		
7		[90 90]	$[\ -5\lambda\ 5\lambda\ ]$	—–—		
8	4	[-45 45 -45 45]	$[\ -\lambda\ -\lambda\ \lambda\ \lambda\ ]$	XX		
9		[-45 45 -45 45]	$[\ -5\lambda\ -5\lambda\ 5\lambda\ 5\lambda\ ]$	X-X		
10		[90 90 90 90]	$[\ -3\lambda\ -\lambda\ \lambda\ 3\lambda\ ]$	||||		
11		[90 90 90 90]	$[\ -15\lambda\ -5\lambda\ 5\lambda\ 15\lambda\ ]$	|−|−|−|		

The script not only depicts how to modify the Winner parameters but also illustrates how to run multiple simulations and then aggregate the results. The script employs a hexagonal grid (two rings) with shadow fading (simulations for 2.6 GHz or 800 MHz frequency were performed) with a 20 MHz bandwidth and a channel speed of 5 km/h. Only the UEs in the center rings (seven sites, thus 21 eNodeBs) are considered, and one UE per eNodeB was placed per eNodeB.

In order to obtain enough UEs results to generate a throughput CDF, each 50-TTI-long simulation case was repeated 20 times, each with a different channel trace. Thus, the shown CDFs are obtained from 420 UE throughput values.

The resulting figures are saved to the `reproducibility/included_simulation_results/antenna_separation` folder, together with the simulation results files.

Figures 9 and 10 depict the simulation results using the antenna configurations in Table II for an 800 MHz and 2×2 comparison and a 2.6 GHz and 4×4 comparison, respectively.

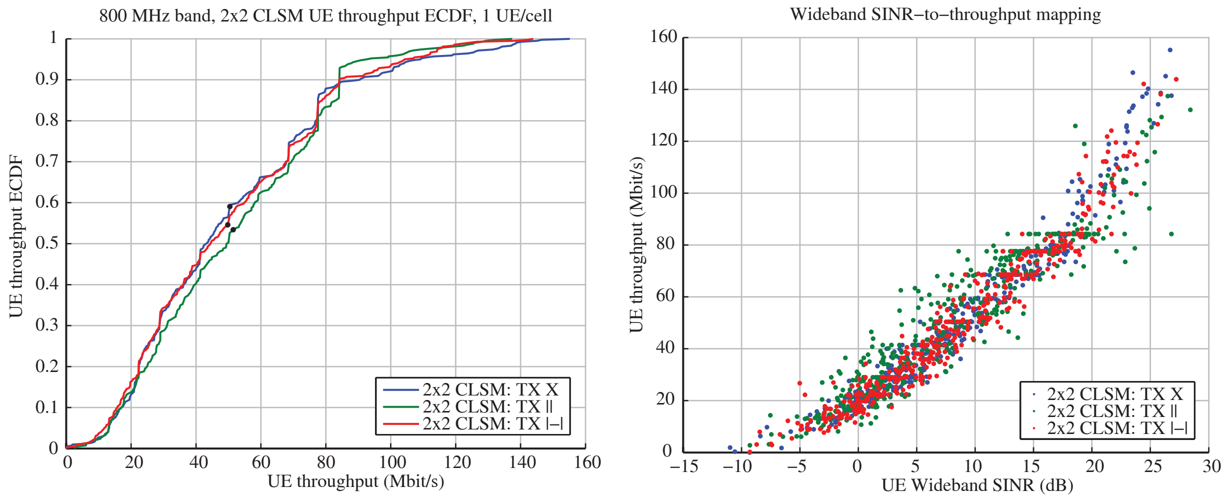


Fig. 9. UE throughput comparison for CLSM with a 2×2 antenna configuration on the 800 MHz band. Left: UE throughput empirical CDF for each of the scheduler setups. The mean value is marked in the CDF as a black dot. Right: Mapping between the UE wideband SINR and the throughput achieved by each UE.

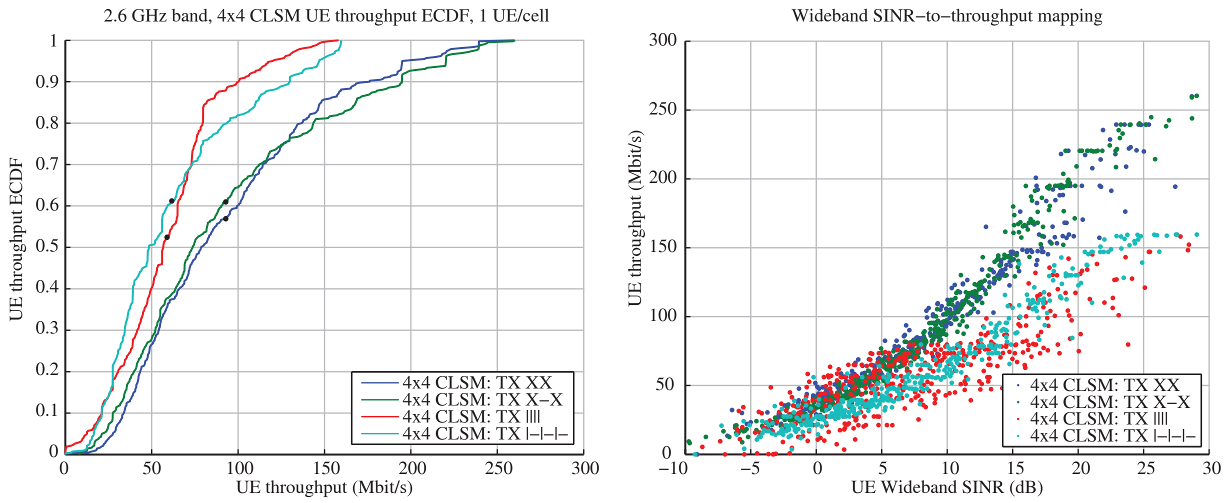


Fig. 10. UE throughput comparison for CLSM with a 4×4 antenna configuration on the 2.6 GHz band. Left: UE throughput empirical CDF for each of the scheduler setups. The mean value is marked in the CDF as a black dot. Right: Mapping between the UE wideband SINR and the throughput achieved by each UE.

E. Fractional Frequency Reuse performance results

This section shows some example FR results, taken from te results shown in [24], and reproducible by running the appropriate scripts included with the simulator release.

The `Launcher_FFR_performance_simulations` script in the `reproducibility/FFR` folder contains the batch file that will allow you to perform FFR simulations over a range of FR bandwidth allocation ratios β_{FR} and SINR thresholds Γ_{thr}, as shown in Figure 11 [24].

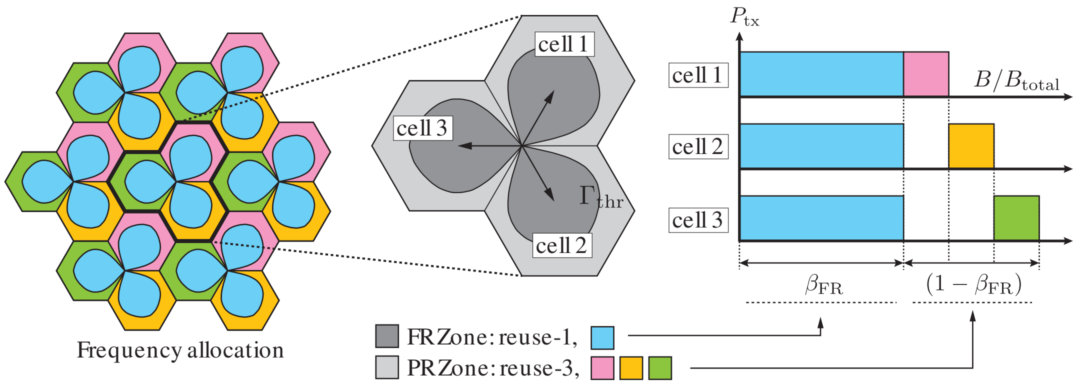


Fig. 11. FFR partitioning employed in the LTE system level simulator. The FR bandwidth is defined by a β_{FR} parameters, while the separation between the FR and PR is determined by an SINR threshold Γ_{thr}.

After performing the simulations (`parfor` use is recommended), you can call the `Postprocess_FFR_results` script to extract just the needed informations (it may take a while to read all of the results file every time you need to generate the figures new) and plot them with the `Plot_postprocessed_FFR_results` script.

In each of the simulations, a cell layout such as the one shown in Figure 12 is employed (the simulation parameters can be found in the paper of the configuration file itself).

As output from the `Plot_postprocessed_FFR_results` script, plots such as the ones in Figure 13 (also included in [24]) are generated.

Each of the points in the surface plot corresponds to a system level simulation, to which a UE throughput CDF is associated, as shown in Figure 14, where for the plots showing fairness [20] for the following FFR configurations defined by their corresponding β_{FR}-Γ_{thr} parameter points: (i) reuse-1, (ii) reuse-3, (iii) $\beta_{\text{thr}} = 0.34$, $\Gamma_{\text{thr}} = 10\,\text{dB}$ (increased fairness relative to the reuse-1 case).

Aggregate plots such as the ones comapring round robin and proportional fair scheduling in [24] are also generated by the `Postprocess_FFR_results` script.

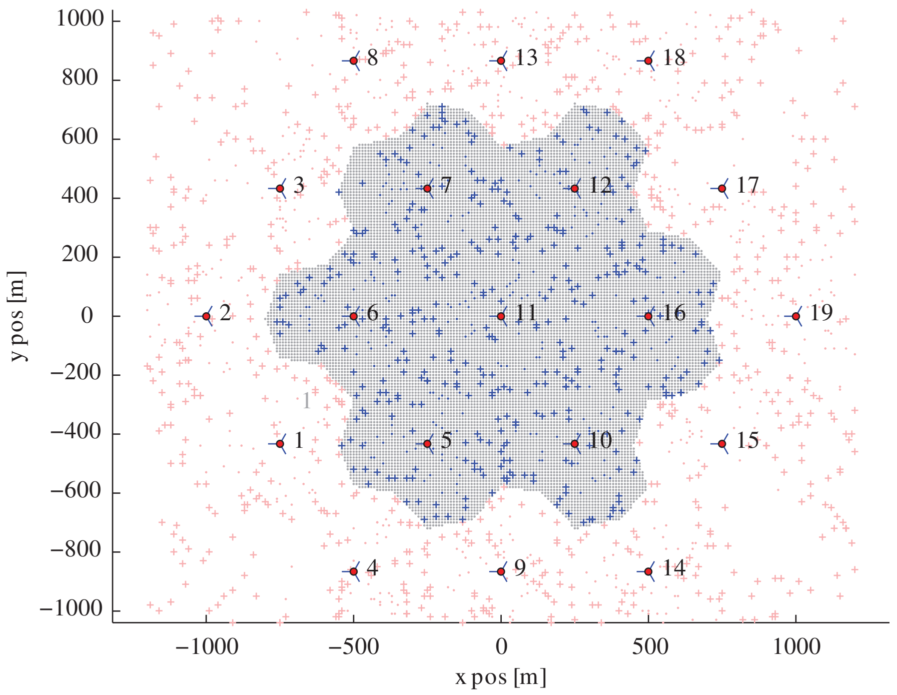


Fig. 12. Cell layout for FFR simulations. Highlighted in gray are the cells taken into account for throughput calculation.

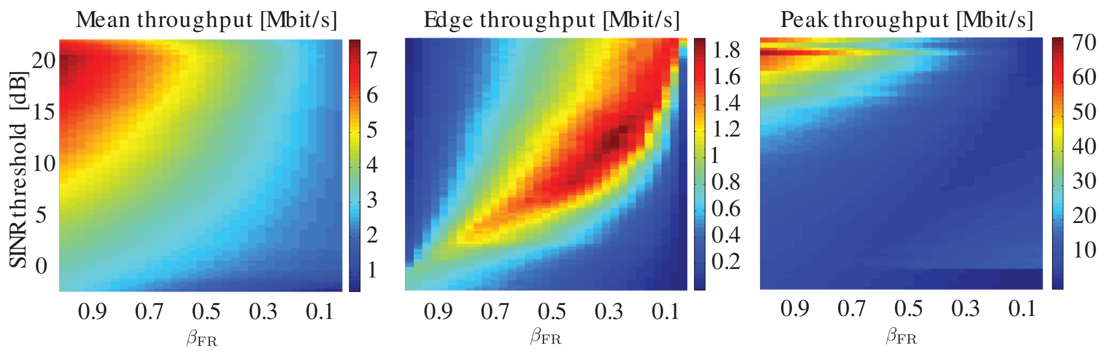


Fig. 13. Mean, edge, and peak throughput of the simulated FFR configurations. Round robin scheduling applied to both the FR and PR zones.

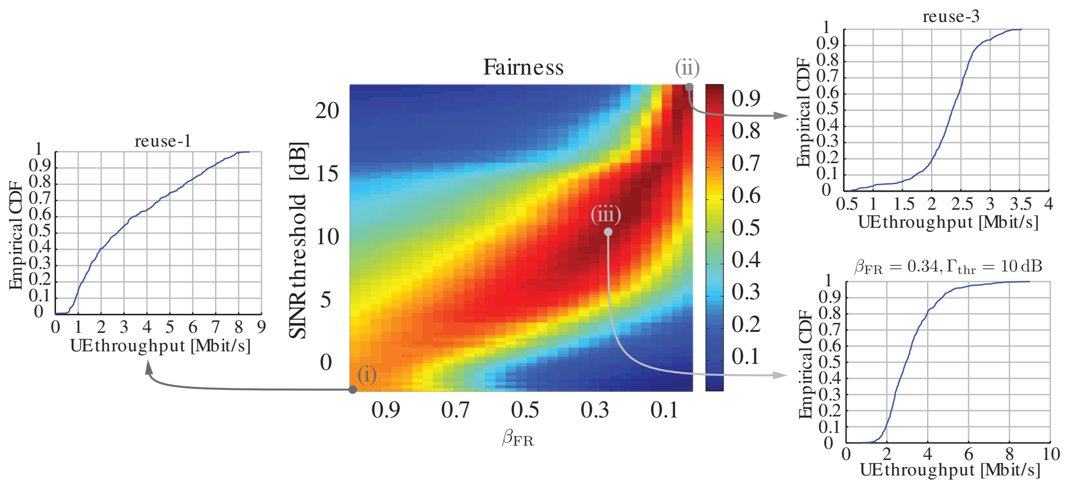


Fig. 14. Fairness results, round robin scheduling.

XVII. Changelog

- v.1.8r1375, 2014-05-15
  - Support for Remote Radio Heads (RRHs) in run-time precoding mode added.
  - The pathloss maps are now stored in linear instead of in dB. This was done so as to part with so many log operations in the SINR calculations.
  - Significant automatic code generation improvements. You can now generate them by calling the `LTE_aux_mex_files`. See Section XV for a list of the `codegen`-enabled functions in the simulator.
  - Improvements in the runtime precoding mode.
  - New network geometry options to generate *predefined*- and *stochastic* network scenarios.
  - Maximum number of users to schedule no longer limited to number of RBs.
  - Round Robin (RR) scheduler for Multiuser MIMO (MU-MIMO) implemented.
  - Basic support for CoMP transmission. New object `CoMP_site` acts as a central entity to organize the scheduling of multiple eNodeBs.
  - Support for transmission mode 6 and 9 (8-layer spatial multiplexing).
  - Various bug-fixes as reported by our online-community (www.nt.tuwien.ac.at/forum).
- v.1.7r1119, 2013-03-07
  - Increased speed of the `LTE_feedback` function for the CLSM (and less but also for OLSM) modes. Instead of looping over all possible precoders, all of them are simultaneously calculated by means of a block diagonal matrix.
  - Added a trace mode for running simulations where you may not have a pathloss map but would nevertheless know the pathlosses at the positions and times the UEs are transmitting. See Section XIV for a more detailed description of the mode.
  - Fixed bug in which after cell change, the fast fading trace starting points are not reset.
  - As for networks with a large number of eNodeBs (e.g., with femtocells) shadow fading generation tended to be fairly sow, a new MEX version combining `parfor` is now included. The MEX version is created using the MATLAB coder [23]. A Win-64 verison is included, but if you have the proper compilers and the code generation capabilities in your MATLAB installation, you can generate, if needed, a new version of the code by replicating the steps detailed in Section XV. In case the included version does not fit your machine or for some reason crashes, the original MATLAB version is run by default.
  - Added the option to specify a fixed pathloss value for each site. You can use it by setting the config variable in `LTE_config.network_source` to `fixed pathloss`. After that, set `LTE_config.pathlosses` to a vector consisting of the pathloss values you want to apply to each eNodeB. An according number of sites will be generated accordingly. Understandably, this setting overrides the pathloss, shadow fading and network cache parameters (it is actually faster to generate this each time than to read a file from disk).
  - Related to the addition above, the simulator now allows you to create a fixed number of UEs over the whole ROI area. Set `LTE_config.UE_distribution` to `constant UEs per ROI` and `LTE_config.nUEs` to the desired number of UEs to create.
  - Added the option to change the 3 dB lobe width for the `TS 36.942` 2D antenna. It can be modified by changing the `LTE_config.TS_36942_3dB_lobe` variable.
  - Added `parfor` support to TxD trace generation.
  - Added option to enable the calculation of the pathloss taking into account the elevation (i.e., the ”3D distance”). By default it is set to take only the 2D projection as the distance, as it should be. However, this can be changed by setting the config variable `LTE_config.calculate_3D_pathloss` to `true`.
  - Resized the results GUIs so that they fit in smaller screens (e.g., a laptop).
  - Added 3D antenna model according to [7], table A.2.1.1-2- 3GPP Case 1 and 3 (Macro-cell).
  - Added comparison with link level simulations in Section XVI-A.
  - Improved the interference structure. Until now, only the closest N sites, as set by a configurable threshold, were taken into account. this, however, proved to be inadequate for simulations with femtocells, where the femto power is very different to that of the macro tier. Now, the set of interfering eNodeBs considered for the calculation of the post-equalization SINR are only those from which the average received power is at most 45 dB lower than that of the received signal, calculated as a sum over all RBs. As this fix required modifying the `eNodeB` class, this fix also requires that you re-generate your network cache files. Thanks to Felipe Gómez Cuba (Centro Tecnolóxico de Telecomunicacións de Galicia, Spain) for pointing out this bug.
  - Added the option to reuse cached network files, so as not to have to load them from the disk when performing several snapshot simulations with the same network topology. Do note that because of this change, old pathloss map caches should be generated again.
  - Added a `radial` UE distribution for use with the Link-to-System (L2S) validation scripts.
  - Moved the SINR averaging configuration to `LTE_load_parameters_dependant`, as now just MIESM is

supported and is not a parameter that would be commonly changed.
  - Added extension to the shadow fading calculation. Additionally to the modes in [8], 12 neighbors are also now selectable.
  - Deleted the ζ and ξ parameters from the channel trace. While this would make it not possible to apply a fixed channel estimation error, the channel estimation error is anyway SINR dependant, so storing those two parameters did not bring any extra benefit. The memory footprint of the simulator should now be reduced (trace size in memory reduced to approximately half).
  - Fixed crash in `phy_modeling.channelTraceFactory_v1` when the parallel toolbox in not present. Thanks to Zoraida Frías Barroso (Universidad Politécnica de Madrid, Spain) for pointing out this bug and kindly providing a fix.
  - Added Single-Input Single-Output (SISO) case to the example in Section XVI-C (CLSM comparison).
  - In order to lighten up FFR simulation results storage, an even more reduced subset of results data can now be saved.
  - Fixed bug in the proportional fair scheduler. The scheduler was wrongly calculating the metric when more than one layer were present.
- v.1.6r885, 2012-05-17
  - For older MATLAB releases, the `RandStream.setDefaultStream` function is called instead of the newer `RandStream.setGlobalStream` in `LTE_load_parameters_dependant`. Requested by some users with older MATLAB versions.
  - The `edge` function is now not employed in `LTE_plot_loaded_network` when not available (requested from some users without the Image Processing Toolbox).
  - Added tracing of the average radiated energy per correctly received bit on a per-UE basis.
  - Improved the way the parameter files are loaded. You now find the parameter files in the `+simulation_config` subfolder. Each file is now loaded through the `LTE_load_parameters` function.
  - Fixed several bugs in the importing of Capesso pathloss maps.
  - Some code reorganization. The release should now be more compact.
  - A different pathloss than the one for macrocells can now be used for femtocells.
- v.1.5r855, 2012-05-07
  - Added the option of controlling additional parameters of the Winner II channel model trace generation, such as antenna spacing and polarization. Example simulation results can be found in Section XVI-D.
  - Fixed inconsistencies in the `attachUser`, `deattachUser` and `userIsAttached` functions in the `eNodeb_sector` class. The `dlnode` therefore not needed anymore. Thanks to Szu-ying Hu (National Chung Hsing University, Taiwan) for pointing out this bug.
  - Added simple handover functionality to exemplarily illustrate how such functionality can be further implemented. A UE can now request a handover to the eNodeB indicated in `cell_change.target_eNodeB` by setting its `cell_change.requested` variable to `true`. By default, this variable is set to `0`. After the cell change is realized, the values of `cell_change.requested` and `cell_change.target_eNodeB` are set to `false` and `[]` respectively.
  - Added some example results and instructions on how to reproduce them (see Section XVI). This is the main reason for the (big) increase in the size of the simulator release.
  - When using the winner II channel model, the polarization and spacing of the antennas can now be modified.
  - Added results equivalent to those in the original VTC paper [2], which included a fix of the best CQI scheduler.
  - Substituted the calls to the `rad2deg` and `wrapTo360` functions in `LTE_init_generate_network`. As a result, the Mapping Toolbox is not needed anymore to run the simulator. Thanks to Marion Richelmy (National Institute of Applied Sciences, France) for pointing out the usage of a Toolbox for just these two easily-substitutable functions.
  - Discontinued the use of the `print_log` function.
  - Chages in the simulation traces: The UE trace now includes the RI feedback. The traces are no longer organized in `eNodeB` and `sector` traces. Now just eNodeB (in the sense of cell) traces are included.
  - FFR simulations are now possible.
  - `LTE_config` is now loaded as a function to allow for parallel runs of the simulator via `parfor` loops (but beware that quite a lot of RAM may be needed). Similarly, `LTE_load_parameters_dependant` is now called inside of the `LTE_sim_main` function, which was sometimes leading to some confusions when reusing configuration parameters structs. The parallelization is realized by means of `parfor` calls to the simulator. Check the FFR simulations included for an example on how to call the simulator in a loop.
  - Fixed bug in the link quality model due to an incorrect usage of `repmat` + `reshape`. the bug just applies to cases when the power distribution is inhomogeneous across RBs, such as in FFR, so it was not noticed until now.
  - Fixed bug that caused a scheduling malfunction when a UE went out of the ROI.
  - Changed the results-plotting scripts to adapt them the the new code updates (some had stopped working) and to make

  them more useful. See Section V for more information.
  - Significantly speed-up when using the `prop fair Sun` scheduler. The old implementation of the `proportional fair` scheduler is not part of the simulator any more.
  - Added runtime precoding (i.e. changed the link quality model). This means the trace format is also changed. The old operation is still supported. However, new functions will be implemented having in mind the flexbility of the new link quality model. The old link quality code has basically been rewritten.
  - Added examples of how to read the simulation results via the use of the new `utils.resultsFileReader` object.
  - Reorganized the whole trace generating code.
  - Executing the `clear` command at the end of a simulation took an enourmous amountof time for some simulations, probably due to the high amount of cross-handles between many UEs and eNodeBs (why exactly is actually unclear to me). This was specially troublesome when performing several simulations in chain. This should now be solved. The variables are now efficiently pre-cleared after writing the results, saving MATLAB the work.
  - `global` is no longer used for `LTE_config`. It was anyway bad practice.
  - Optimized feedback calculation code in `UE.m`. Due to this optimizations, EESM is no longer supported for SINR averaging. Only MIESM is now supported.
  - Added necessary layer mappings for simulations using up to four transmit layers in CLSM and OLSM. i.e. 4×4 transmissions are now supported forboth Spatial Multiplexing (SM) modes.
  - Fixed bug regarding the interfering channel trace that caused incorrect behaviour for 4 transmit antennas due to an overestimation of the interfering power. The mistake was present just for the OLSM and CLSM modes and is now corrected. Old channel traces should therefore be generated again.
  - Parallel processing of the channel trace generation is now supported again (OLSM and CLSM modes only, though). The parallelization works on a per-layer basis except for CLSM, where more extensite parallelization has been applied due to the long time the feedback calculation requires. However, as a result memory the requirements for CLSM trace generation are now higher due to the higher number of parallel MATLAB workers. While a 2×2 trace remains reasonable in size (10 GByte of RAM for a trace length of 10 s), creating a 4×4 trace requires around 20 GB of RAM.
  - The trace generation code is now moved to the `phy_modeling.channelTraceFactory_v1` object. A bug regarding the feedback calculation with an incorrect dimension handling of the channel matrix for the CLSM case in `LTE_feedback` has also been resolved.
  - Changed the pathloss map structure so a tri-sector structure is not needed anymore. The old pathloss map cache format will however NOT work anymore, so cached pathloss maps should be generated again.
  - Simulations with femtocells are now possible. Refer to the `LTE_sim_main_launcher_examples` script for more details.
  - Simulations for a six-sector setup are now included in the `LTE_sim_main_launcher_examples` batch file.
- v.1.4r570, 2011-08-26.
  - Importing of pathloss data from Atoll™/Capesso™data is now possible.
  - Fixed bug in the SISO part of the link performance model that caused a crash when a single-enodeB seturp with zero-delay was used. Thanks to Zoraida Frías Barroso (Universidad Politécnica de Madrid, Spain) for pointing out this bug and kindly providing a fix.
  - Fixed bug in the proportional fair scheduler that caused the recources to be incorrectly assigned. Thanks to Leticia Almansa (Universidad Politćnica de Madrid, Spain) for pointing out this bug and kindly providing a fix.
  - Fixed error in `LTE_init_generate_users.m` that caused the simulation to crash when no UEs were present.
  - Minumum Coupling Loss can be now applied to loaded pathloss maps.
  - Added tracing of an overall SINR similar in concept to the SINR RS, although different, as the RS SINR should not include precoding and this does.
  - Separated the CQI and RI feedback calculation from the link quality model, so as to improve code readability.
  - Added functions to help in generating KML files for plotting results.
  - Antenna gain for the receive antenna can now be set by means of the optional `LTE_config.UE.antenna_gain` parameter. This gain is assumed to be omnidirectional.: e.g. setting it to `−3` sets an additional loss of 3 dB because of the receiver antenna.
  - Added option to output smaller result files, in case many simulations are performed (i.e. throughput ECDFs). To use it, set `LTE_config.compact_results_file` to `true`.
  - Improved code for feedback sending and the link between the link quality and link performance models.
  - Performance increase and better code in the MIESM averager, link quality model, and link performance model. Simulations should now be noticeably faster.
  - Fixed bug in the channel trace generation in which the code would not run if the parallel or distributed toolbox would not be installed.

  - Added support for using Winner+ channel traces. Deprecated the option to use uncorrelated microscopic fading and moved all channel trace generation routines to objects.
  - Corrected bugfix that slip past the last release: Fixed bug in SINR calculation in the link performance model. The `zeta` (ζ) parameter was incorrectly used twice, although since it is an identity matrix the result was unchanged. Thanks to Taka Sakurai (University of Melbourne, Australia) for pointing out this bug.
  - Slightly improved, updated, and revised the parameter description in the documentation.
  - Pathloss files can now be imported from Capesso network planning tool maps. See Section XI for more information.
  - Fixed small bugs in the MIESM SINR averaging implementation and small performance improvements.
  - Added a configurable penetration loss. It will only make a difference when there are not many interferers, though. You can configure it by means of the `LTE_config.additional_penetration_loss` variable. It can be set to a number, which will be the penetration loss in dB or to the following, with its equivalent value explained next to the text:
    * `deep indoor`: 23 dB
    * `indoor`: 17 dB
    * `incar`: 7 dB
    * `outdoor`: 0 dB
  - Added a configurable signaling part (always radiated) of the eNodeB power. When no UEs are attached to a given eNodeB, this ratio of the overall power will still be radiated by the eNodeB. Thus, the overal power will be assigned as follows: `LTE_config.signaling_ratio` of the maximum power which will be always radiating and the rest will be assigned by the scheduler.
- v.1.3r427, 2010-11-05.
  - Fixed bug in the throughput calculation. The two time slots in a subframe were not correctly considered.
  - Fixed normalization of the precoding matrices.
  - Optimized shadow fading map generation.
  - Added "Companies (no matter profit-oriented or not) are not allowed for free usage and have to contact the licensor before usage." to the license agreement (Section A).
  - The channel trace now only contains the current TX mode, instead of all of them. While this prevents in-simulation mode changing, it does reduce trace size.
  - Fixed bug in SINR calculation in the link performance model. the `zeta` parameter was incorrectly used twice, although since it is an identity matrix the result was unchanged.
  - Added option to use 3D antenna patterns instead of the typical 2D antenna patterns. The included antenna patterns were provided by KATHREIN-Werke KG.
  - Added option to configure whether the eNodeBs are always transmitting (interfering) or not. Setting `LTE_config.always_on=false` will cause the allocated power to be zero when no UEs are attached to the scheduler.
  - Added Traffic Models according to RAN R1-070674; they are just used with the constrained scheduler (but this one still needs some work).
  - Added Rank Indicator Feedback to OLSM.
  - Fixed Transport Block (TB) size calculation issues in the scheduler files.
  - Moved common scheduling code to the `lteScheduler` class to improve code readability and decrease indivudial scheduler algorithm implementation complexity.
  - Improved simulator speed by changing how microscale fading traces are stored. The old trace format will not work anymore.
  - Changed MIESM SINR averaging to a faster implementation that also supports calibration via β parameters. It uses up more memory but is much faster due to it not using the `interp1` function in every call. MIESM is now: $\gamma_{\text{eff}}\left(\underline{\gamma}\right) = \beta\, I_{m_j}^{-1}\left(\frac{1}{N}\sum_{\gamma_i \in \underline{\gamma}} I_{m_j}\left(\frac{\gamma_i}{\beta}\right)\right)$,where β is calibrated for each Modulation and Coding Scheme (MCS). Since the old implementation is easier to understand implementation-wise, it is still present in the release. It does NOT support calibration though.
  - In `LTE_init_generate_FF_tracev2` the frequency was incorrectly hardcoded to be $2110 \cdot 10^6$ instead of being read from `LTE_config.frequency`. Thanks to Yanjun Ma (Xidian University, China) for pointing out this bug.
  - Fixed bug: in order to calculate the SINR in the link quality model, the interference from your neighboring sectors attached to your sme eNodeB site was not considered. Thanks to Wang Bo (Beijing University of Posts and Telecommunications, China) for pointing out this bug.
  - Fixed crash in the `proportionalFairScheduler` when the average throughput `T_k` is zero.
  - Fixed simulator crash when shadow fading was not present (e.g. when loading data from an external source that has

macroscopic and shadow fadings already combined).
  - The configuration option `LTE_config.use_fast_fading` has been removed. It could be replaced by the use of an Additive White Gaussian Noise (AWGN) channel as channel matrix for the trace generation, but for such simulations the LTE link level simulator would be in most cases much more appropriate. Hence this configuration parameter has been dropped.
- v.1.2r310, 2010-06-25.
  - Fixed bug in the scheduler implementation that made it impossible to simulate scenarios where users are not only positioned in the center cell. Changed default scenario to UEs positioned all around the ROI also.
- v.1.1r295, 2010-04-12.
  - Updated AWGN BLER curves to more precise ones which also use less space. See Section IV-D for more information.
  - Fixed bug that caused the simulator to crash when storing the exponentially-averaged throughput when using less than two data streams.
  - Fixed an error in the generation of the OLSM traces. The used CDD D matrix was erroneously $\begin{bmatrix}1 & 0 \\ 0 & e^{-j2\pi/2}\end{bmatrix}$ instead of $\begin{bmatrix}1 & 0 \\ 0 & e^{-j2\pi i/2}\end{bmatrix}$. So it was not changing over time.
- v.1.0r247, 2010-03-15.
  - First publicly available version of the LTE System Level Simulator.

XVIII. Referencing

A version of the LTE System Level Simulator (LTE SL Simulator) paper is available in our publication data-base **here**. If you are using the simulator for your scientific work, please use the refence below:

```
@InProceedings{VTC2010,
  author =        {Josep Colom Ikuno and Martin Wrulich and Markus Rupp},
  title =         {System level simulation of {LTE} networks},
  booktitle =     {Proc. 2010 {IEEE} 71st Vehicular Technology Conference},
  month =          may,
  year =           2010,
  address =       {Taipei, Taiwan},
  url =           {http://publik.tuwien.ac.at/files/PubDat_184908.pdf},
}
```

```
J. C. Ikuno, M. Wrulich, and M. Rupp, System level simulation of LTE networks,
in Proc. 2010 IEEE 71st Vehicular Technology Conference, Taipei, Taiwan, May 2010. [Online]
Avaialable: http://publik.tuwien.ac.at/files/PubDat_184908.pdf
```

XIX. Known issues

- LTE SL Simulators make use of the new Object-Oriented capabilities of Matlab (available since R2008a), the simulators will not run under older Matlab releases without extensive changes.
- Please note that MEX-files generated using Microsoft Visual C++ 2008 require that Microsoft Visual Studio 2008 run-time libraries be available on the computer they are run on. The runtime files can be downloaded **here** (x86) or **here** (x64). If you would need to compile for another arquitecture, the `LTE_aux_mex_files` script can be used.

XX. Questions

For questions please check our **forum**, where you will be able to post your questions/comments/bug reports. It makes it easer for you to see what other people asked and also makes it easier for us to answer you (when we have time).

XXI. Mailing List

If you want to receive information about future updates you can subscribe to our LTE simulator mailing list **here**. Note that you can change the display language to english in the selection panel to the right.

XXII. The People (so far) behind the development of the LTE System Level Simulator

- Josep Colom Ikuno
- Martin Klaus Müller
- Markus Rupp
- Stefan Schwarz
- Martin Taranetz
- Martin Wrulich

XXIII. License agreement

These terms (license for the LTE system level simulator) refer to the use of the *LTE system-level simulator* (the "Original Work"), developed by the Institute of Telecommunications, Vienna University of Technology (the licensor).

A. Academic Usage

Academic Usage in the context of this license describes the use of the Original Work in scientific projects without any reimbursement or financial claims that bear on results derived by the Original Work, but subject however to the restrictions provided for in Clause B hereinbelow. The main goal in the sense of Academic Usage shall be to obtain cientifically significant results that can be used for publication.

Companies (no matter profit-oriented or not) are not allowed for free usage and have to contact the licensor before usage.

B. Grant of copyright license

Licensor grants You a worldwide, royalty-free, non-exclusive, non-sublicensable license, restricted to non-commercial use, for the duration of the copyright, to install the Original ork and any Derivative Works thereof on one personal computer. The license allows You to:

1) Use the Original Work only for Academic Usage. Any usage of the Original Work, entirely or in part or modified, requires the proper citation, e.g. as reference in a publication.
2) Translate, adapt, alter, transform, modify, or arrange the Original Work, thereby creating derivative works ("Derivative Works") based upon the Original Work. Distribution, either royalty-free or commercially, in parts or in modified form of the Original Work, i.e. also of Derivative Works, is prohibited and not covered by "Academic Usage".
3) Display results derived from the Original Work, or in modified form, publicly, without commercial usage.

C. Grant of source code license

The term "Source Code" means the preferred form of the Original Work for making modifications to it and all available documentation describing how to modify the Original Work. Licensor agrees to provide a machine-readable copy of the Source Code of the Original Work along with each copy of the Original Work that Licensor distributes. Licensor reserves the right to satisfy this obligation by placing a machine-readable copy of the Source Code in an information repository reasonably calculated to permit inexpensive and convenient access by You for as long as Licensor continues to distribute the Original Work.

D. Exclusions from license grant

Neither the names of Licensor, nor the names of any contributors to the Original Work, nor any of their trademarks or service marks, may be used without express prior permission of the Licensor, except as expressly provided otherwise in Clause B1 hereinabove. Except as expressly stated herein, nothing in this License grants any license to Licensor's trademarks, copyrights, patents, trade secrets or any other intellectual property. No license is granted to the trademarks of Licensor even if such marks are included in the Original Work. Nothing in this License shall be interpreted to prohibit Licensor from licensing under terms different from this License any Original Work that Licensor otherwise would have a right to license.

E. Warranty of provenance and disclaimer of warranty

Licensor warrants that the copyright in and to the Original Work is owned by the Licensor or is sublicensed to You under the terms of this License with the permission of the contributor(s) of those copyrights and patent rights. Except as expressly stated in the immediately preceding sentence, the Original Work is provided under this License on an "AS IS" BASIS and WITHOUT WARRANTY, either express or implied, including, without limitation, the warranties of noniinfringement, merchantability or fitness for a particular purpose. THE ENTIRE RISK AS TO THE QUALITY OF THE ORIGINAL WORK IS WITH YOU. This DISCLAIMER OF WARRANTY constitutes an essential part of this License. No license to he Original Work is granted by this License except under this disclaimer.

F. Limitation of liability

Under no circumstances and under no legal theory, whether in tort (including negligence), contract, or otherwise, shall the Licensor be liable to anyone for any indirect, special, incidental, or consequential damages of any character arising as a result of this License or the use of the Original Work including, without limitation, damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses. This limitation of liability shall not apply to the extent applicable law prohibits such limitation.

G. Termination

If, at any time, You infringe upon the grants of this License, it shall terminate immediately and You may no longer exercise any of the rights granted to You by this License.

H. Open source and code under other license terms

The original work also contains work licensed under other licenses other than the license for the LTE system-level simulator. The terms and conditions described in this document are only applicable to the parts of the Original Work not under other licenses. A list of the parts of the Original Work not under the license for the LTE system-level simulator can be found in Appendix I.

I. Appendix I

The following parts of the original work are not under the terms of the license for the LTE system-level simulator, and are thus excluded from the terms and conditions stated by this license.

The usage and adaptation of these sections for use with the original work is done in compliance with the license terms they are released under. Any translation, adaptation, alteration, transformation, modification, or further use of the hereinbelow stated parts of the original work must be done under the terms of the applicable licenses for that specific part, which are also included in the package.

- Hash calculation in the `utils.hashing` class. The `DataHash` function provides MD5/SHA1 hash functionality and is under the BSD license [25]. The license agreement for the `DataHash` code is found in the `documentation/` folder. The code was retreieved from the Matlab File Exchange on 27.04.2012 [26].

XXIV. Acknowledgments

The whole LTE research group for continuous support and lively discussions. This work has been funded by the Christian Doppler Laboratory for Wireless Technologies for Sustainable Mobility, KATHREIN-Werke KG, and A1 Telekom Austria AG. The financial support by the Federal Ministry of Economy, Family and Youth and the National Foundation for Research, Technology and Development is gratefully acknowledged. The views expressed in this paper are those of the authors and do not necessarily reflect the views within any of the entities which financially supported this work.

References

[1] [Online]. Available: http://www.nt.tuwien.ac.at/ltesimulator/

[2] J. C. Ikuno, M. Wrulich, and M. Rupp, "System level simulation of LTE networks," in *Proc. 2010 IEEE 71st Vehicular Technology Conference*, Taipei, Taiwan, May 2010.

[3] Technical Specification Group RAN, "E-UTRA; physical channels and modulation," 3GPP, Tech. Rep. TS 36.211 Version 8.7.0, May 2009.

[4] Random number stream - MATLAB. [Online]. Available: http://www.mathworks.com/access/helpdesk/help/techdoc/ref/randstream.randstream.html

[5] Technical Specification Group RAN, "E-UTRA; LTE RF system scenarios," 3GPP, Tech. Rep. TS 36.942, 2008-2009.

[6] ——, "Physical layer aspects for E-UTRA," 3GPP, Tech. Rep. TS 25.814, 2006.

[7] ——, "E-UTRA; further advancements for E-UTRA physical layer aspects," 3GPP, Tech. Rep. TS 36.814, 2010.

[8] H. Claussen, "Efficient modelling of channel maps with correlated shadow fading in mobile radio systems," Sept. 2005.

[9] "Recommendation ITU-R M.1225: Guidelines for evaluation of radio transmission technologies for IMT-2000," Tech. Rep., 1997.

[10] T. B. Sorensen, P. E. Mogensen, and F. Frederiksen, "Extension of the itu channel models for wideband (ofdm) systems," in *Proc. 2005 IEEE 62nd Vehicular Technology Conference*, 2005.

[11] L. Hentilä, P. Kyösti, M. Käske, M. Narandzic, and M. Alatossava, "MATLAB implementation of the WINNER Phase II Channel Model ver1.1," December 2007. [Online]. Available: https://www.ist-winner.org/phase_2_model.html

[12] Z. Sun, C. Yin, and G. Yue, "Reduced-complexity proportional fair scheduling for ofdma systems," in *Proc. 2006 International Conference on Communications, Circuits and Systems Proceedings*, june 2006.

[13] P. Viswanath, D. Tse, and R. Laroia, "Opportunistic beamforming using dumb antennas," *IEEE Transactions on Information Theory*, vol. 48, pp. 1277–1294, 2002.

[14] M. Moisio and A. Oborina, "Comparison of effective sinr mapping with traditional avi approach for modeling packet error rate in multi-state channel," in *Proc. NEW2AN 2006*, St. Petersburg, Russia, 2006.

[15] L. Wan, S. Tsai, and M. Almgren, "A fading-insensitive performance metric for a unified link quality model," in *Proc. IEEE Wireless Communications and Networking Conference (WCNC2006)*, vol. 4, Apr. 2006, pp. 2110–2114.

[16] Technical Specification Group RAN, "E-UTRA; physical layer procedures," 3GPP, Tech. Rep. TS 36.213, March 2009.

[17] J. C. Ikuno, M. Taranetz, and M. Rupp, "Fractional frequency reuse as a fairness-adjusting mechanism in LTE networks," *submitted*, 2012.

[18] C. Mehlführer, M. Wrulich, J. C. Ikuno, D. Bosanska, and M. Rupp, "Simulating the long term evolution physical layer," in *Proc. of the 17th European Signal Processing Conference (EUSIPCO 2009)*, Glasgow, Scotland, Aug. 2009.

[19] Technical Specification Group RAN, "E-UTRA; LTE physical layer – general description," 3GPP, Tech. Rep. TS 36.201 Version 8.3.0, March 2009.

[20] R. K. Jain, D.-M. W. Chiu, and W. R. Hawe, "A Quantitative Measure Of Fairness And Discrimination For Resource Allocation In Shared Computer Systems," Digital Equipment Corporation, Tech. Rep., Sept. 1984.

[21] L. Thiele, T. Wirth, K. Brner, M. Olbrich, V. Jungnickel, J. Rumold, and S. Fritze, "Modeling of 3D Field Patterns of Downtilted Antennas and Their Impact on Cellular Systems," in *International ITG Workshop on Smart Antennas (WSA 2009)*, Berlin, Germany, Feb. 2009.

[22] M. Grant and S. Boyd. (2012) CVX: Matlab software for disciplined convex programming. [Online]. Available: http://cvxr.com/cvx/

[23] T. MathWorks. (2013) MATLAB coder. [Online]. Available: http://www.mathworks.de/products/matlab-coder/

[24] J. C. Ikuno, M. Taranetz, and M. Rupp, "A fairness-based performance evaluation of fractional frequency reuse in LTE," in *17th International ITG Workshop on Smart Antennas (WSA2013)*, Stuttgart, Germany, Mar. 2013.

[25] "BSD license." [Online]. Available: http://opensource.org/licenses/BSD-2-Clause

[26] J. Simon, "DataHash." [Online]. Available: http://www.mathworks.com/matlabcentral/fileexchange/31272

List of Figures

1 Coordinate convention used in the LTE SL Simulator. 11
2 SNR-to-CQI mapping. Left: CQI BLER curves. Right: CQI mapping obtained from the 10 % BLER points. . . . 12
3 Positions GUI. Shaded via the selction of the appropriate cells on the cell list: approximate cell area corresponding to the center eNodeBs. The UEs attached to the selected eNodeBs are shown black. Shaded are the rest of the UEs. 13
4 Aggregate results GUI. The same eNodeBs as in Figure 3 are selected. Plots show the throughput ECDF for the average UE throughput (upper-left), spectral efficiency (upper-right), wideband (lower-left) SINR and the mapping between between the wideband SINR and the average throughput for each UE. The results are computed from the UEs attached to the selected eNodeBs. Some overall statistics are shown on the grey text box. 14
5 Left: SISO and 2×2 TxD. Right: OLSM. 20
6 LefT: CLSM. Right: All antenna configurations. 20
7 UE throughput comparison for different types of scheduling: round robin, proportional fair, and best CQI. Left: UE throughput empirical CDF for each of the scheduler setups. The mean value is marked in the CDF as a black dot. Right: Scheduler comparison in terms of mean, edge, and peak UE throughput. Fairness is shown in the legend. 21
8 UE throughput comparison for CLSM with a 2×2, 4×2, and 4×4 antenna configuration. Left: UE throughput empirical CDF for each of the scheduler setups. The mean value is marked in the CDF as a black dot. Right: Mapping between the UE wideband SINR and the throughput achieved by each UE. Fairness is shown in the legend. 21
9 UE throughput comparison for CLSM with a 2×2 antenna configuration on the 800 MHz band. Left: UE throughput empirical CDF for each of the scheduler setups. The mean value is marked in the CDF as a black dot. Right: Mapping between the UE wideband SINR and the throughput achieved by each UE. 22
10 UE throughput comparison for CLSM with a 4×4 antenna configuration on the 2.6 GHz band. Left: UE throughput empirical CDF for each of the scheduler setups. The mean value is marked in the CDF as a black dot. Right: Mapping between the UE wideband SINR and the throughput achieved by each UE. 23
11 FFR partitioning employed in the LTE system level simulator. The FR bandwidth is defined by a β_{FR} parameters, while the separation between the FR and PR is determined by an SINR threshold Γ_{thr}. 23
12 Cell layout for FFR simulations. Highlighted in gray are the cells taken into account for throughput calculation. 24
13 Mean, edge, and peak throughput of the simulated FFR configurations. Round robin scheduling applied to both the FR and PR zones. 24
14 Fairness results, round robin scheduling. 24

List of Tables

I Default MIMO antenna settings employed with the Winner Phase II channel model. 17
II Antenna configurations for each of the simulations in `LTE_sim_main_launcher_antenna_separation`. Several transmit antenna configurations of the Winner II channel model are tested. At the receiver side, cross-polarized antennas were always used. 22