Vienna LTE Simulators
System Level Simulator Documentation, v1.4r570

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

Abstract

This document contains documentation on how to use the Long Term Evolution (LTE) System Level 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 system level 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 System Level simulator can be found in Section XVII. Please read the license agreement carefully.

II. Running a simulation

The main file of the LTE Link Level Simulator is `LTE_sim_main.m`, though you may normally run the simulation through a batch file such as `LTE_sim_launcher.m`, which performs the following tasks:

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

III. Simulation parameters

Below you can find a list of the parameters that can be configured in the `LTE_load_params` file:

A. General parameters

- `LTE_config.debug_level`: configures how much debug text output is shown. Options are:
  - `0`: no output.
  - `1`: basic output.
  - `2`: extended output.
- `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.
- `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.UEs_only_in_target_sector`: defines whether UEs are created in the whole Region Of Interest (ROI) or just in the target sector. for the case where the UEs are just positioned in one sector, the other transmitters radiate at maximum power, just acting as interferers.
- `LTE_config.target_sector`: if UEs are only to be set in the target sector, this setting specifies which one that is. Set it to`center` for specifying the target sector to be the center one. `[eNodeB_id sector_id]` otherwise.
- `LTE_config.nTX`: number of transmit antennas. Used to generate the microscale fading trace.
- `LTE_config.nRX`: number of receive antennas. Used to generate the microscale fading 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 (not yet implemented).
- `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].
- `LTE_config.simulation_time_tti`: length of the simulation in Transmission Time Intervals (TTIs).
- `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 [5], 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`.

B. 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 (eg. `data_files/network_1_rings_5m_res_TS25814_2.00GHz_freq.mat`).
- `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.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 (eg. `data_files/UE_cache_1rings_target_sector_only_20UEs_sector_20100301_114247.mat`).

C. Network layout and macroscopic pathloss params

- `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™planning tool (see Section VII). When using this source, shadow fading data is not generated, as the imported pathloss maps should already have it incorporated.

1) Generated network:

- `LTE_config.inter_eNodeB_distance`: in meters. When the network is generated, this determines the distance between the eNodeBs.
- `LTE_config.map_resolution`: in meters/pixel. Also the resolution used for initial user creation.
- `LTE_config.nr_eNodeB_rings`: number of eNodeB rings.
- `LTE_config.minimum_coupling_loss`: 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 [6] 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. $L = \left(\frac{4\pi d}{\lambda}\right)^2$. d in meters.
  - `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 [6] for more information. Possible environments are:
  * `urban`: $L = 40\left(1 - 4 \cdot 10^{-3} \cdot \mathrm{Dhb}\right) \cdot \log_{10}(R) - 18\log_{10}(\mathrm{Dhb}) + 21\log_{10}(f) + 80\,\mathrm{dB}$. 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`: $L = 69.55 + 26.16 \cdot \log_{10}(f) - 13.82 \cdot \log_{10}(\mathrm{Hb}) + [44.9 - 6.55 \cdot \log_{10}(\mathrm{Hb})]\log_{10}(R) - 4.78\left(\log_{10}(f)\right)^2 + 18.33 \cdot \log_{10}(f) - 40.94$. 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 [7] 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 transmit poower, in Watts. Recommended by [8] are:
  - 43 dBm for 1.25, 5 MHz carrier
  - 46/49 dBm for 10, 20 MHz carrier.

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

D. 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 [9].
  - `none`: No shadow fading map. For simplicity reasons this is implemented as a shadow fading map with a constant value.
- `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 [9] 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).

E. Microscale fading

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 [10].
  - `PedB`: ITU Pedestrian B channel [10].
  - `extPedB`: Extension of the ITU channel models for wideband (OFDM) systems [11].
  - `VehA`: ITU Vehicular A channel [10].
  - `VehB`: ITU Vehicular B channel [10].
- `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.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.channel_model.correlated_fading`: `true` or `false`. Activates or deactivates the channel time correlation.
- `LTE_config.recalculate_fast_fading`: whether generate the trace even if the file already exists (force a new trace).

F. UE settings

- `LTE_config.UE.receiver_noise_figure`: receiver noise figure in dB. Set to 9 dB [6].
- `LTE_config.UE.thermal_noise_density`: thermal noise density in dBm/Hz.

- `LTE_config.UE_per_eNodeB`: number of UEs per sector.
- `LTE_config.UE_speed`: speed at which the UEs move. In meters/second.
- Traffic map related parameters: Related to the importing of Capesso-generated maps. See Section VII for a detailed explanation of the parameters.

G. 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`: $A(\theta) = -\min\left[12\left(\frac{\theta}{65^\circ}\right)^2, 20\,\mathrm{dB}\right]$, $-180 \leq \theta \leq 180$ [6].
  - `kathreinTSAntenna`: Antenna pattern to be read from an an antenna pattern file. 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.mean_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).
- `LTE_config.scheduler`: the type of scheduler to use. Supported schedulers are `round robin`, `best cqi` (Max C/I), and `proportional fair`. Please note that the proportional fair scheduler has not been throughly tested and may be buggy.
- `LTE_config.power_allocation`: only `homogeneous` is supported right now.

H. Uplink channel options

- `LTE_config.feedback_channel_delay`: uplink delay in TTIs. When set to 0 TTIs, only the Channel Quality Indicator (CQI) reports experience zero delay. ACK reports have a minimum delay of one TTI.
- `LTE_config.unquantized_CQI_feedback`: there is an option to send unquantized feedback, which is de-facto sending the measured Signal to Interference and Noise Ratio (SINR), as then afterwards the CQI is not mapped.

I. SINR averaging

- `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 [12]:
  - `EESM`: use Exponential Effective Signal to Interference and Noise Ratio Mapping (EESM). The following configuration parameters are needed:
    * `LTE_config.SINR_averaging.MCSs`: the Modulation and Coding Schemes (MCSs) defined in [13].
    * `LTE_config.SINR_averaging.betas`: the calibration β parameters that fit the EESM function to the Additive White Gaussian Noise (AWGN) Block Error Ratio (BLER) curves.
  - `MIESM`: use Mutual Information Effective Signal to Interference and Noise Ratio Mapping (MIESM). Please note that MIESM has not yet been throughly tested with the simulator. Some bugs may be present. 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_10000_realizations.mat`.

J. Saving of 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.

K. Values that should not be changed

- LTE_config.RB_bandwidth.: Bandwidth of a RB. 180 KHz and should not be changed. It basically used for throughput calculations.
- LTE_config.TTI_length: length of a TTI (subframe) in seconds.
- LTE_config.cyclic_prefix: set to normal. It is used to calculate the number of available bits in each subframe, so it will not realistically reflect the effect of using another cyclic prefix length.
- LTE_config.maxStreams: maximum number of codewords per TTI. Set to two.

L. Optional configuration parameters

The following parameters, to allow for backwards-compatibility, are optional. If you do not specify them, they will take a default value, which is described here.

- 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.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_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.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.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.

IV. Implementation Issues

A. How to read the pathloss maps

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

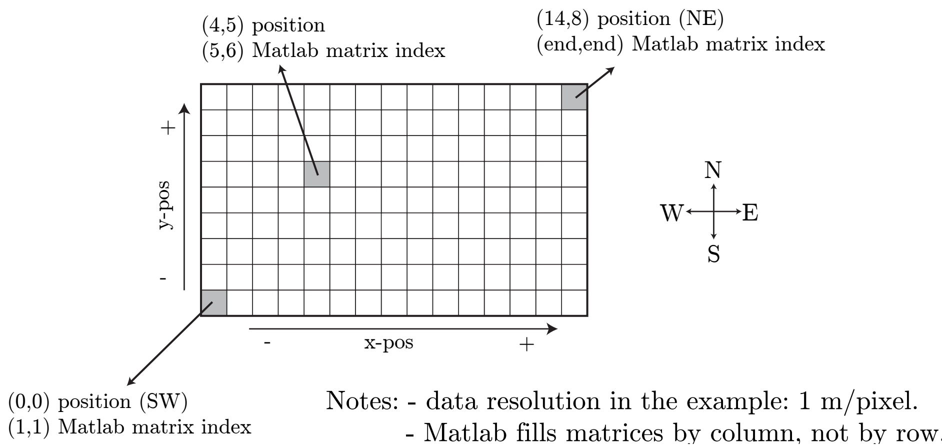


Fig. 1. Coordinate convention used in the LTE system level simulator.

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. Loading the BLER curves

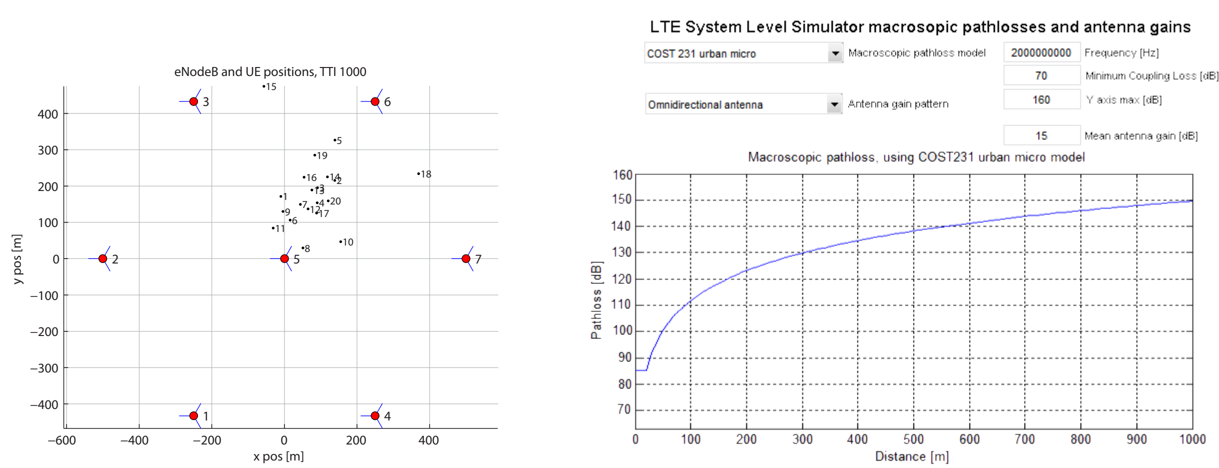
The BLER curve data files can be supplied to the simulator in the following forms:

- LTE link-level simulator [14] `.mat` results file.
- `.mat` file containing the following vector variables of equal length:
  - `SNR`: Signal to Noise Ratio (SNR) values.
  - `BLER`: BLER values.

V. Plotting Results

The `LTE_sim_results` provides some plotting of the result traces. The generated plots are the following:

- eNodeB and UE positions: self-explanatory. See Figure 2(a).
- `LTE_GUI_pathloss_antenna_info`: Graphical User Interface (GUI) that shows you the available macroscopic pathloss models and antenna gain patterns. Does not actually plot results. See Figure 2(b).



(a) UE position

(b) `LTE_GUI_pathloss_antenna_info` GUI

- Sector throughput and BLER: for every sector, shows the average throughput (summing both streams, when applicable) and overall BLER. If no users are assigned to the sector, `NaN` may be displayed as BLER.

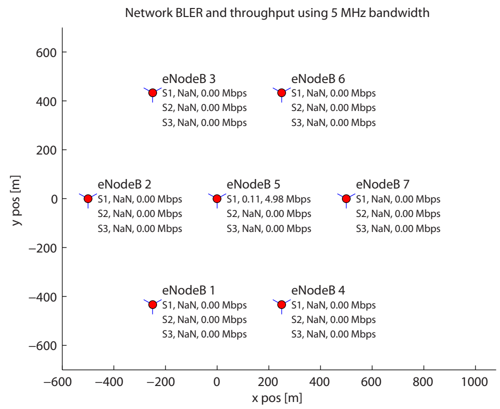


Fig. 2. Sector throughput and BLER

- `LTE_GUI_show_UE_traces`: this GUI shows you the following plots:
  - throughput and BLER over time. The blue line depicts the UE throughput in Mb/s for the selected stream and UE, as well as the BLER as measured by the ACK/NACK ratio (green line) and the BLER value applied by the link quality model. Although the system is calibrated to deliver BLERs$\geq$0.1, the actual results are influenced by the uplink delay and time variability of the channel.
  - UE position in the ROI.
  - sent CQI report for the selected RB and stream (blue), mean CQI for the whole frequency band (red) and CQI of to the Transport Block (TB) sent to the UE, if scheduled.
  - distribution of the CQIs for the selected UE and RB during the simulation time (blue), and of the TB CQIs (red).

  Note that all time-dependant data is averanged by using a rectangular window of configurable length.

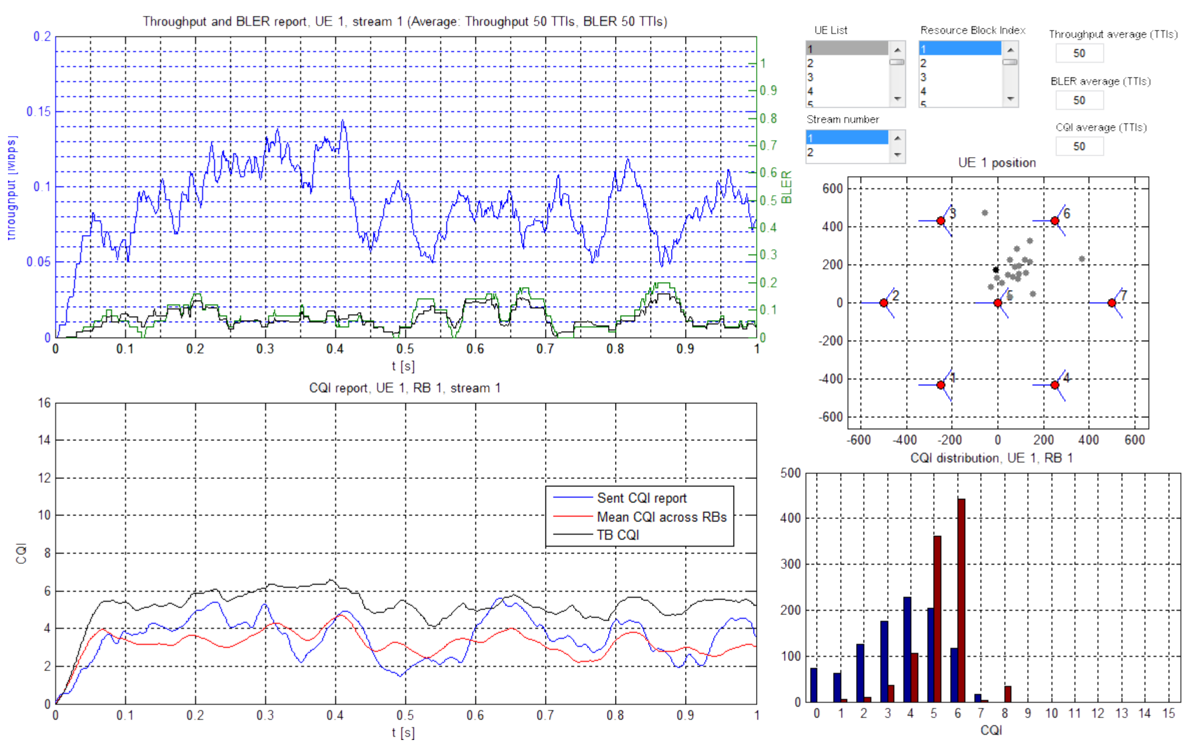


Fig. 3. `LTE_GUI_show_UE_traces` GUI

- `LTE_GUI_show_cell_traces`: GUI depicting the cell traces for the selected eNodeB/sector pair. Contains the following figures:

- graphical depiction of the RB allocation for the selected TTI.
- evolution of the number of assigned RBs to each UE during the selected TTI range.
- throughput and BLER for the selected stream number. Throughput and BLER are averaged using a rectangular window of configurable length. In order to make the post-processing faster, the cell throughput is calculated with the ACKed data from the UEs instead of checking the throughput of every attached UE. Thus, the uplink delay makes you lose the value for some TTIs.

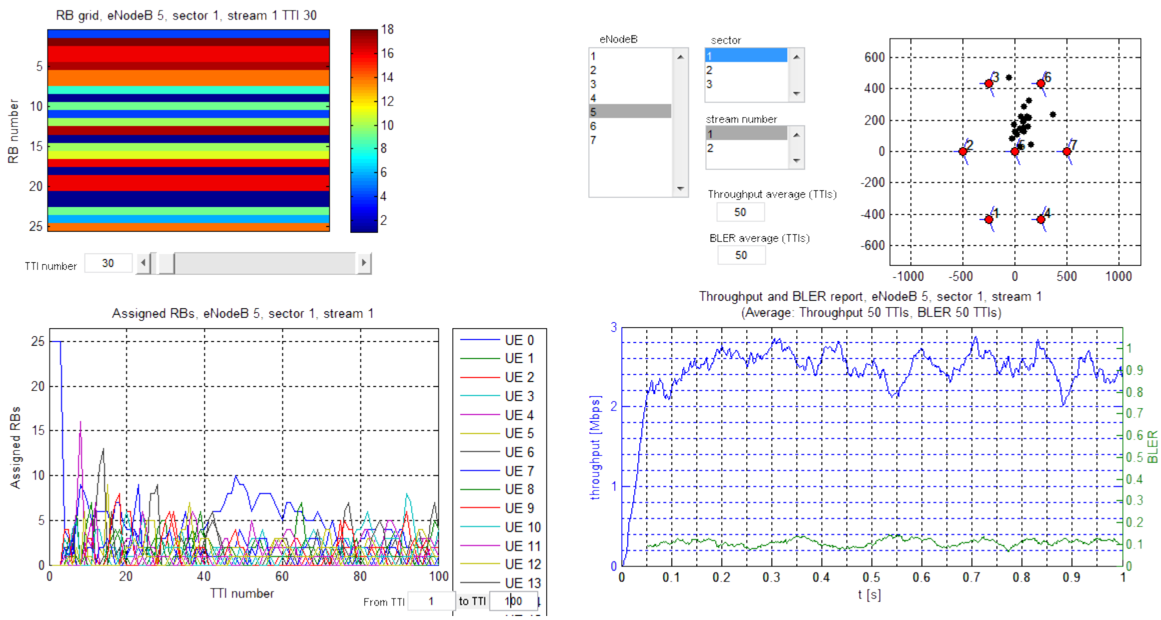


Fig. 4. `LTE_GUI_show_cell_traces` GUI

VI. 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 [6]. 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 [15]. 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.

VII. 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`).

VIII. 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 [16]. 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.

IX. 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. 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.

X. Reproducing the results/plots presented on a paper

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 publications, a set is configurable that will reproduce the results in the corresponding paper:

- 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 [2]:
  In the `reproducibility/VTC2010` forlder, the `plot_ECDFs` script allows you to reproduce the CDF plots shown in the paper using the already pre-processed results. If you would like to reproduce the simulation traces from which the result files were extracted (∼8 GB of traces), you can do so by moving the `LTE_sim_VTC`, `LTE_sim_batch_2x2_*_*`, and `LTE_load_params_2x2_*_*` scripts to the root folder of the simulator and executing the `LTE_sim_VTC` script. The result files can then be post-processed with the `LTE_plot_UE_throughput_CDF` script (it must be also copied to the simulator root folder). Please note that you will have to adapt the folder-related variables to suit where the result files are located.
  The simulations from [2] were done with the following basic configuration (Table I):

[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

Parameter	Value
Frequency	2.0 GHz
Bandwidth	5 MHz
Thermal noise density	-174 dBm/Hz
Receiver noise figure	9 dB [6]
nTX×nRX antennas	2×2
TX mode	TxD and OLSM
Simulation length	500 TTIs
Number of simulations	200 per scenario
Inter eNodeB distance	500 m
Minimum Coupling Loss	70 dB [6]
Macroscopic pathloss	$128.1 + 37.6 \log_{10}(R)$ [7]
Shadow fading	lognormal, space-correlated [9], $\mu = 0, \sigma = 10\,(dB)$
Shadow fading correlation	Inter-site: 0.5, Intra-site: 1 [6]
eNodeB TX power	43 dBm [6]
Microscale fading	PedB uncorrelated, time-correlated [17]
UEs position	Homogeneous. UEs located in target sector only, 20UEs/sector
UE speed	5 KM/h
BS Antenna pattern	$A(\theta) = -\min\left[12\left(\frac{\theta}{65^\circ}\right)^2, 20\,\mathrm{dB}\right], -180 \le \theta \le 180$ [6]
BS antenna gain	15 DBi [6]
Scheduler	Round Robin, Max C/I (Best CQI)
Subcarrier averaging algorithm	EESM
Uplink delay	3 TTIs

TABLE I
CONFIGURATION PARAMETERS USED FOR [2]

XI. CHANGELOG

- 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 up 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 for pointing up 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 VII 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 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 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 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-B 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.

XII. Referencing

A version of the LTE System Level 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
```

XIII. Known issues

- The LTE 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.

XIV. 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).

XV. 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.

XVI. The People (so far) behind the development of the LTE System Level simulator

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

XVII. 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.

Acknowledgment

The authors would like to thank the whole LTE research group for continuous support and lively discussions. This work has been funded by mobilkom austria AG, the Christian Doppler Laboratory for Wireless Technologies for Sustainable Mobility, as well as the Institute of Communications and Radio-frequency Engineering. The views expressed in this paper are those of the authors and do not necessarily reflect the views within mobilkom austria AG.

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

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

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

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

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

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

[11] 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.

[12] 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.

[13] Technical Specification Group RAN, "E-UTRA; multiplexing and channel coding," 3GPP, Tech. Rep. TS 36.212, March 2009.

[14] 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.

[15] 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.

[16] 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

[17] Y. R. Zheng and C. Xiao, "Simulation models with correct statistical properties for rayleigh fading channels," *Communications, IEEE Transactions on*, June 2003.