



LTE Emulator

(downlink connection)

version 1.0

**Authors: Zsolt A. Polgar,
Mihaly Varga,
Vasile Bota**

**Data Transmission Laboratory,
Communication Department,
Faculty of Electronics, Telecommunications and
Information Technology
Technical University of Cluj-Napoca**

Contents

| | |
|--|----|
| 1. Requirements of the emulator project..... | 2 |
| 2. The OFDMA multi-user access technique. Definition of the transport frames..... | 3 |
| 3. Operating principle and basic structure of the LTE emulator..... | 5 |
| 3.1. Functional blocks of the emulator. Parameters of the functional blocks..... | 6 |
| 3.1.1. The signal to noise ratio computation block..... | 6 |
| 3.1.2. The command and control block..... | 8 |
| 3.1.3. The emulation statistics display block..... | 10 |
| 3.2. Main operations performed in the emulation process..... | 10 |
| 3.2.1. Interception and filtering of the processed packets..... | 11 |
| 3.2.2. Delay of the intercepted packets and error insertion in these packets..... | 12 |
| 4. The parameters of the LTE simulator..... | 14 |
| 4.1. Parameters that characterize the cell in the simulation process..... | 14 |
| 4.2. Parameters that characterize the users in the simulation process..... | 15 |
| 4.3. Parameters that define the modulation and FEC coding techniques and the OFDM access method in the simulator..... | 16 |
| 4.3.1. Parameters of the adaptive modulations employed on OFDM sub-carriers..... | 16 |
| 4.3.2. Parameters of the error-correcting codes employed..... | 16 |
| 4.3.3. Parameters of the data structures employed in transmission..... | 18 |
| 4.4. The main parameters of the simulation..... | 18 |
| 4.5. The output data delivered by the simulator..... | 19 |
| 5. The principle of the simulation process..... | 20 |
| 5.1. The channel model employed in the simulation process..... | 21 |
| 5.2. The structure of the simulation program and main processing performed..... | 23 |
| 5.2.1. Allocation of radio resources. The scheduler block..... | 24 |
| 5.2.2. Generation of statistics tables. Table formats..... | 26 |
| 6. Compatibility with 3GPP standards. Possible developments..... | 30 |
| 6.1. Compatibility with E-UTRA standards..... | 30 |
| 6.2. Possible developments of the present version..... | 32 |
| 7. Installation of the emulation program. Computational resources required..... | 34 |
| 8. Adjustment of the bit rate transferred through the emulator. Remarks..... | 35 |
| 9. Appendix: Multipath propagation models for different frequency bands..... | 37 |
| 10. References..... | 39 |

1. Requirements for the emulator project

The „LTE Emulator– downlink connection” program emulates the downlink connection of a transmission over the LTE radio interface between a base–station computer and mobile–station computer. It must also allow the establishing of the uplink connection, without affecting the transmitted data (quasi-ideal link). The transmission scenarios considered assume a single cell and a single channel carrier.

The main functions that must be performed by the emulator are:

- a. to intercept the packets of an Ethernet connection between a base-station computer (or computers) and a mobile-station computer (or computers);
- b. to emulate the delays and errors introduced by the transmission on the radio interface between the two computers, in the defined scenario, on the downlink connection;
- c. to allow the modification of the emulated transmission scenario’s parameters;
- d. to display the parameters of the defined scenario and the performances of the connection provided;
- e. to employ files containing statistics of the delays and errors introduced by the transmission in the defined scenario, in the emulation process;
- f. to ensure the transparent Ethernet connection on the uplink between the mobile-station computer and the base-station one;

2. The OFDMA multi-user access technique.

Definition of the transport frames

The OFDMA multi-user access technique is usually based on a frequency-time signal structure (see fig. 2.1.), which can be adjusted to various propagation conditions and is able to cope with the frequency selectivity generated by the multipath propagation and the time variability generated by the motion of the mobile station. Such a signal structure, called „bin” or „chunk”, is proposed for the LTE („Long Term Evolution”, also called 3.9G) mobile communication systems, [And05] [Bar06] [Chi06] [Hye06] [Kat06] [Pel05] [Sie06], which are still being standardized, as well as for the future 4G mobile communication systems [Win05_1]. Within these chunks, coded or non-coded adaptive QAM modulations are employed [Bot04] [Ste03] [Win05_1], based on the channel measurements and channel-state prediction, made by the mobile station. These operations require the use of a number of spread pilot symbols within the chunk, [Ste03] [Win05_1]; some certain sub-carriers are also only used for synchronization and channel measurement operations. The allocation of chunks to different users on the downlink employs methods that usually take into consideration the predicted instantaneous signal to noise ratio in the frequency bands of the different chunks. The allocation function is performed by a scheduler block.

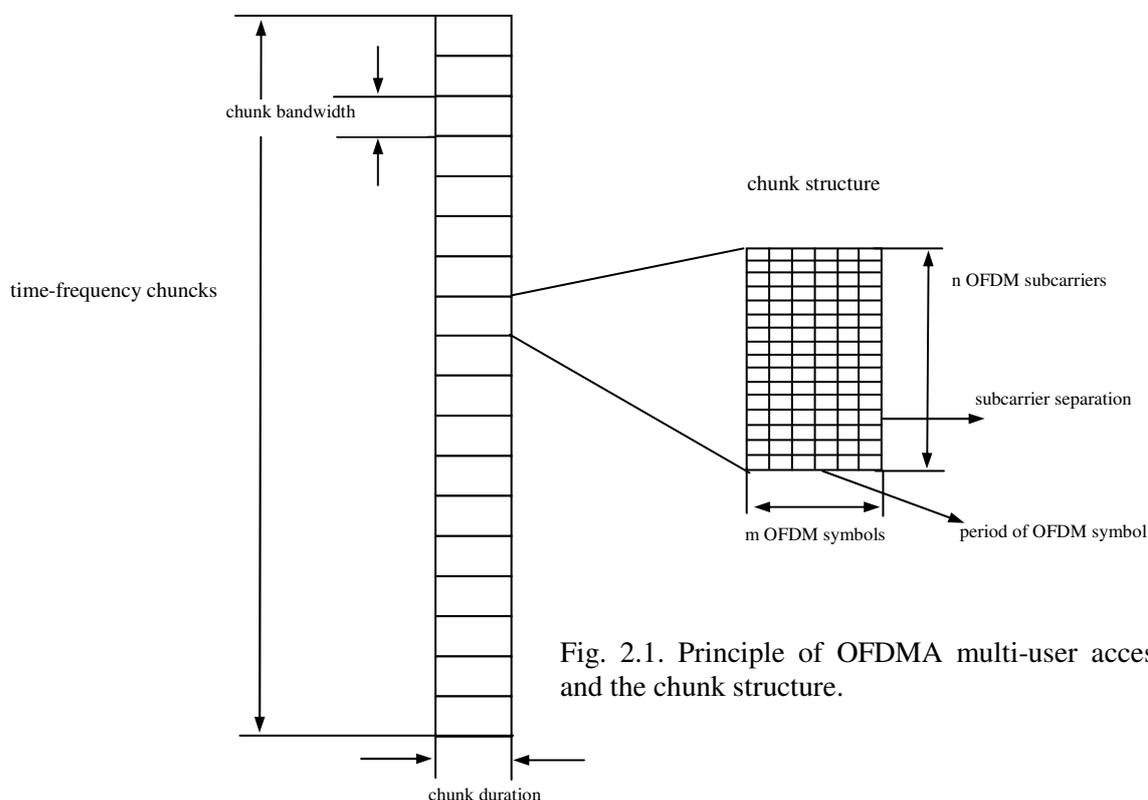


Fig. 2.1. Principle of OFDMA multi-user access and the chunk structure.

The OFDMA multi-user access technique is employed in several wireless and high-speed mobile transmission systems. The IEEE 802.16.x (x=a,...,e) standards specify such OFDMA multi-user access techniques (called SOFDMA – „Scalable OFDMA”) [Yag04]. This technique is based on a structure of time-frequency signal, called burst, whose size is variable, as a function of the channel parameters. In this case, the scheduling operation is based on a special type of „frequency hopping” technique (called „distributed sub-carrier permutation”), which is capable of successfully managing the diversity in frequency and variability in time of the channel [Yag04]. The duplex transmission mode could be TDD or FDD adapted to the „half-duplex” operation mode. The above mentioned standards specify as modulation techniques, the coded QAM adaptive modulations, [Yag04].

Another competitive technology for mobile high bit rate transmissions, called FLASH-OFDM, also based on the OFDMA multi-user access, is proposed by Flarion Technologies, [Lar01] and employs a type of „frequency hopping (FH)” allocation of user chunks.

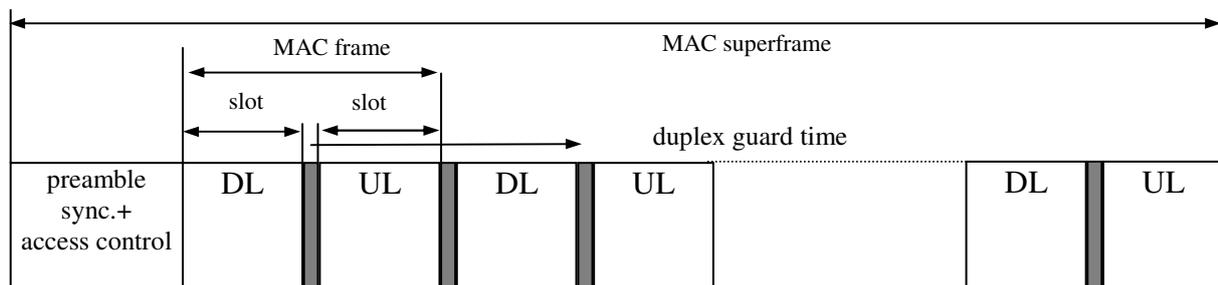
Besides the definition of the data structure (chunk) associated to the OFDMA access scheme, the following definitions must also be considered for the description of the transmission system:

- Transmission Time Interval (TTI) [Bar06] [Chi06] – notion also used in 3G, represents the time interval within which the transmission parameters remain constant, i.e. the modulation and encoding techniques do not change. Considering the definition of the OFDMA chunk and the manner in which the adaptive coded modulations are employed within these chunks, the TTI can be considered equal to a chunk period.
- The transport frame (or transport block) [Fau06] [See06] [Val06], represents the frame (block of data) with a duration of one TTI in which the user’s data is actually loaded. The transport frame can be a MAC radio frame or a fraction of it – see fig. 2.2. [Win05_1]; the MAC frame includes information regarding synchronization and access control; besides, it may also include data structures used in the management of uplink and downlink connections and data structures employed for signaling.

A straightforward option is to equalize a transport frame to a chunk (solution proposed by WINNER), which will lead to severe limitations of the transport frames’ lengths (the set of coded modulations that can be used within a chunk is limited) and to a higher variability of the length of this frame as a function of the available radio resources at a certain moment. However, it will allow an easier employment of adaptive coded modulations – one codeword within a chunk with an imposed modulation.

Another possibility is the construction of a transport frame composed of several chunks; this approach would allow longer frames and would reduce the frame length’s variability with available resources (a frame of a certain length could be loaded in several ways on the OFDMA chunks), but it will generate problems in the implementation of the coded modulations.

A possible solution for these problems would be the separation of the coding and the adaptive modulation. The bits of a frame are first encoded with a certain error correction code and the obtained sequence is mapped on several chunks employing different modulations. The modulation used is independently adapted to the particular transmission conditions of each chunk. The adaptive change of codes is possible in this case as well, but it must be performed based on certain parameters that characterize the transmission performances in several chunks, in which different modulations are used.



one time slot = 1 chunk/transport frame period in FDD mode; 3 chunktransport frame period in TDD mode
 one radio superframe = 8 or 16 radio frames

Fig. 2.2. MAC frame and superframe structures associated to the OFDMA multiuser access technique.

3. Operating principle and basic structure of the LTE emulator

The operating principle of the emulator consists of the interception of packets containing the data destined to (generated by) the main user, the one being tested, and the delay and/or error insertion in these packets, similarly to the delay and error insertion produced by a transmission through a radio system with the specified parameters. The error and delay insertion is made according to delay and error statistics (probability density functions - pdf) obtained from the simulation of the radio transmission over the specified system or from processing certain measurements, if these are available. A schematic representation of the chain of processing that must be performed by the emulator is presented in fig. 3.1.

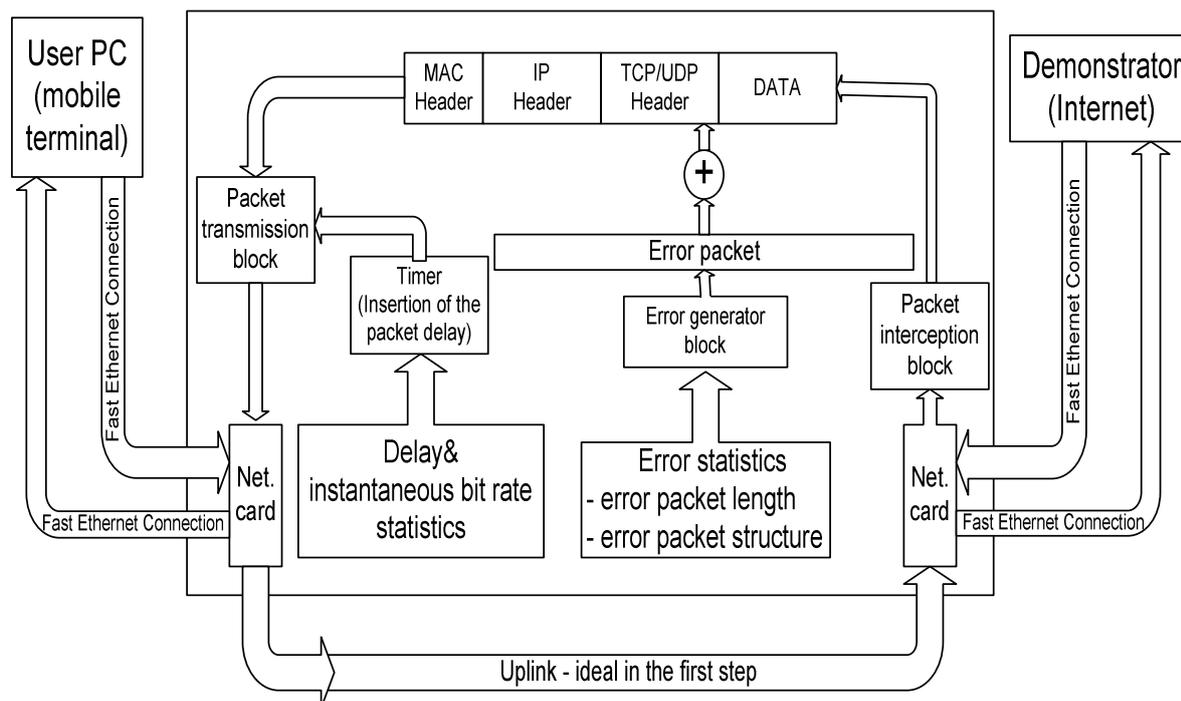


Fig. 3.1. Chain of processing performed by the LTE emulator.

An important problem of this approach is the large number of data structures (files, tables) containing statistics that have to be employed, each scenario and each system configuration requiring other statistics (distributions – pdf functions). This generates problems regarding not only the storage of this data, but also their accessibility. Decreasing the number of these statistics is, therefore, a fundamental requirement.

The main idea regarding the statistics generation (error and delay distributions) is the separation of the parameters regarding the transmission scenarios and configurations into two groups, out of which only one is explicitly used during the simulation (statistics generation) process. Then, only during the emulation process, the two categories of parameters are considered in a combined manner.

Practically, the main parameter used during the simulation process is the value of the signal to noise ratio of the first arrived path of the multipath model, SNR_0 , (details are presented in chapter 5. paragraph 5.1. describing the simulation process and the used radio channel model), separate statistics being generated for each value of SNR_0 ; then, the required value for the SNR_0 is computed in the emulator and the data structures with their corresponding statistics are selected. Therefore, the simulation process involves the radio channel model (multipath propagation model and the fading model due to motion) and the load of the channel carrier in the cell, i.e. the number of active users and their characteristics (the service offered, average rate). The present version of the emulator and the corresponding simulator employ only a single channel carrier.

The second group of parameters is also considered in the emulation process, i.e. the ones referring to the cell dimension (the cell is assumed circular, with a given radius), type of cell (urban/rural, small/large), the noise and/or interference distribution, mobile speed and its motion law and characteristics of the base station transceiver (carrier frequency, transmission power, height of the antenna). The value of the SNR_0 is computed by using these parameters and tables with bit error and delay distributions are selected, according to the SNR_0 value, out of a group of tables corresponding to the carrier frequency, multipath propagation profile and the cell carrier load.

The speed of the user can be set both in the simulation and emulation processes. The two speeds do not have to be equal, if the synchronization/equalization operations are considered to be ideal, because the speed does not affect the probability density functions, if the simulation is performed within a sufficiently large time interval. Therefore, different values of the mobile's speed do not require tables with different statistics. The velocity of the user is used in the simulation process, as well as in the emulation one, for computing the coherence time of the channel [Rap01] [Sk197], which has to be greater than the TTI interval. The channel characteristics are considered constant within the coherence time interval and, during this interval, only a single value is employed both for the instantaneous rate (number of bits/TTI) and the number of errors/TTI (the error distribution can vary within the data packet).

3.1. Functional blocks of the emulator. Parameters of the functional blocks

3.1.1. The signal to noise ratio computation block

This block has the purpose of reading the parameters of the cell (the geometry of the cell), of the base station transceiver and of the user's mobility and to compute the value of the SNR_0 . Meanwhile the block provides the display of the main user's trajectory within the cell. The parameters that are processed by this block, divided into 4 groups, are the following (see also fig. 3.2.):

- cell radius (circular cell) expressed in km;
- cell dimension: small cell (radius ≤ 1 km) and large cell (radius > 1 km);
- cell type: urban, suburban, rural;
 - ❖ Note: the last two parameters select the formulae used for the computation of the attenuation, referred to the base station (center of the cell), i.e. the formulae that are used to compute the large scale fading, which depends on the cell-type.
- channel carrier frequency, expressed in MHz;
- transmission power of the base station, expressed in dBm;
- noise power expressed in dBm;
 - ❖ Note: in this emulator version, the noise spectral power density is considered constant in the entire cell.
- height of the antenna of the base station transceiver, expressed in m.
 - ❖ Note: height of the user's antenna is considered constant and equal to 1.5m;
- speed of the user, expressed in km/h;
 - ❖ Note: the speed of the mobile must be chosen such that the coherence time of the mobile radio channel is greater or equal to the duration of the transport frame. The computation of the coherence time is made according to relation (2). *In the present version, the emulator does not check the fulfillment of this condition; this task is to be accomplished by the user, when setting the emulation parameters.*
- the mobile's motion law. It involves the function that describes the position of the mobile in time, within the cell, as a function of the cell's dimension and the mobile

speed (of the main user that is tested). The four motion laws that are implemented are defined bellow:

- random motion – a linear movement with random change of the motion direction at each second;
- linear motion – o linear continuous movement up to the edge of the cell where the motion direction is randomly changed;
- movement on a spiral [Spi] – the mobile moves on a spiral starting at the center of the cell and ending at the edge of the circular cell. The number of loops of the spiral is established randomly. When the mobile reaches the edge of the cell, it turns back to the center of the cell on the same spiral. In the moment when the mobile reaches the center of the cell a new spiral is generated with a random number of loops;
- circular movement on a circle with imposed radius;

The mobile position is modified every second, this being also the time interval when the SNR_0 signal to noise ratio is recomputed.

The motion of the mobile can be suppressed, for different testing operations, and the value of the SNR_0 (the signal to noise ratio associated to the first multipath propagation path) can be imposed by the user.

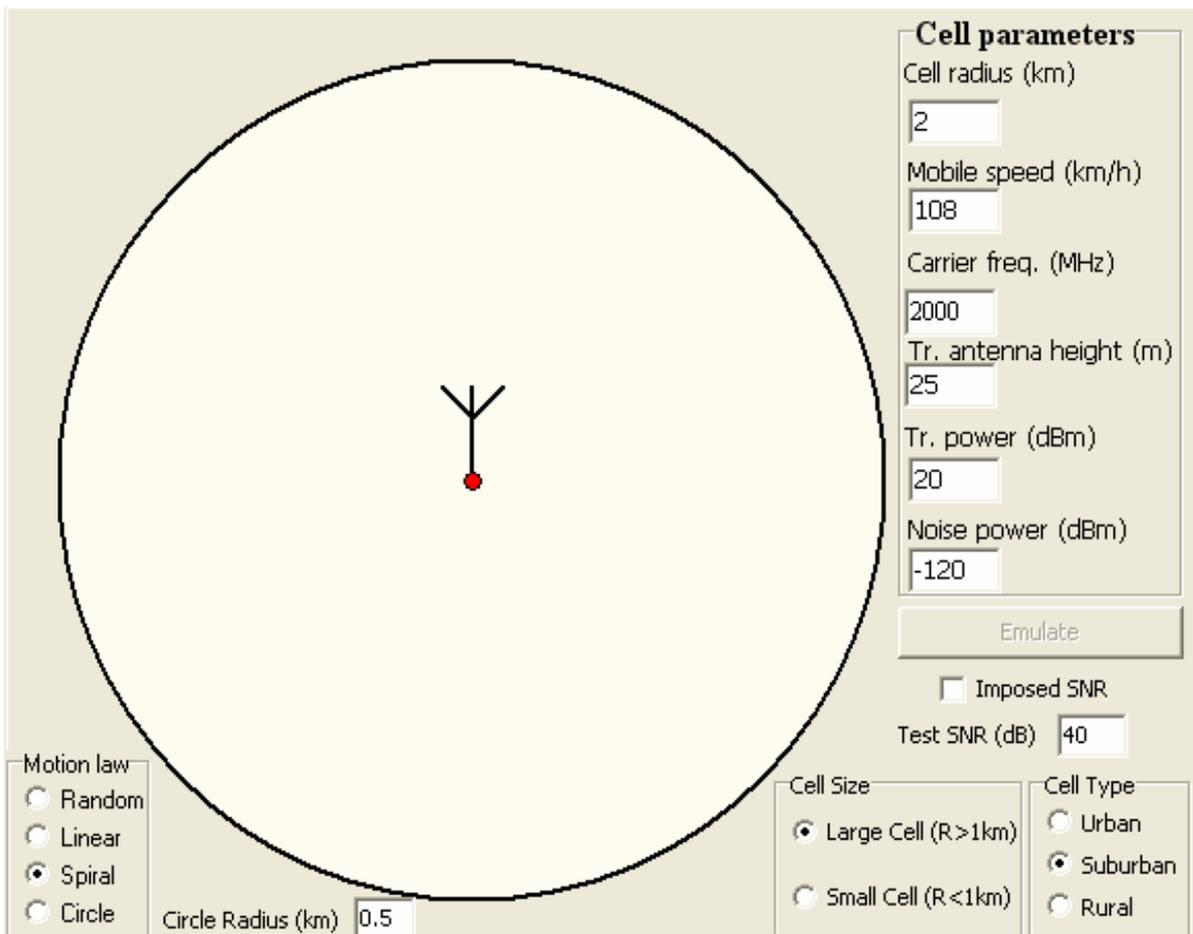


Fig. 3.2. User interface of the signal to noise ratio computation block.

The relations used to calculate the channel attenuation between the mobile and the base station transceiver are the following:

- o for cells with large dimensions (radius>1km) the COST-231 model is used; this model is an extension of the Hata model for large scale fading [Rap01]. Small empirical changes were added in the implementation of this model to correct the attenuation in points

located closely to the cell center, i.e. points located closely to the transmission antenna of the base station transceiver;

- for cells with small dimensions (radius < 1km) the Wideband PCS Microcell Model [Rap01] is used. Small changes were also added in the implementation of this model to correct the attenuation in points located closely to the cell center;

Calculation of the SNR_0 value in a point x is performed according to relation (1).

$$SNR_0(x) = \frac{P_s}{P_n} \cdot a(x) \quad (1)$$

where P_s is the transmitted signal power, P_n is the noise power, and $a(x)$ is the attenuation calculated in point x .

The „Emulate” button starts/stops the emulation process; when this button is in the Start position, an ideal channel with a bit rate limited only by the computation capacity of the computer which runs the program, is emulated.

Another important parameter of the emulation process is the radio channel coherence time, parameters which makes the link between the speed of the mobile and the variability of the mobile radio channel at a given carrier frequency. This parameter is also calculated in this block and is then transferred to the delay and error generation block. The calculation of the coherence time is performed according to relation (2), [Rap01].

$$t_c = \sqrt{\frac{9}{16 \cdot \pi \cdot f_m^2}} = \frac{0.423}{f_m}; \quad f_m = \frac{v \cdot f_c}{c} \quad (2)$$

In (2) v is the speed of the mobile, f_c is the radio channel central frequency, meaning the used carrier frequency, c is the speed of the light in vacuum, f_m is the maximum Doppler frequency, [Rap01].

3.1.2. The command and control block

This block selects the network cards of the computer which runs the emulator, network cards which receive respectively transmit the data packets, and it also takes the MAC addresses of network card which delivers the input data packets and of the one towards which the processed packets are transmitted. It also sets the capture-buffer length, adjusts the nominal transfer rate through the emulator, selects the files with delay and bit-error statistics of the emulated transmission, displays the number of transferred bytes through the emulator in an imposed time interval, displays the average transferred bit rate and establishes the time interval used to compute the average bit rate.

The present version *only emulates the downlink connection (base station – mobile user)*. The local network card(s) which ensures the connection with the computer (or computers) acting as mobile node(s) is selected from the list „Client NIC” (see fig. 3.3.) and the MAC address (addresses) of the network card(s) of the computer(s) acting as mobile node(s) is (are) established with the edit window called „Edit MAC Address List”; the window appears if a double click is applied on the list box called „Mobile Node MAC”, and the mentioned addresses can be visualized with this list box. The selection of the local network card which ensures the connection with the computer (or computers) acting as base station is made from the list „Server NIC” (see fig. 3.3.), and setting the MAC address (addresses) of the network card(s) of the computer(s) acting as base station, is (are) accomplished by the same edit window (the „Edit MAC Address List” window). The window appears if a double click is applied on the list box called „Base Station MAC”, and the mentioned addresses can be visualized with this list box – see fig. 3.3. *In the downlink direction, each packet having the source address included in the Base Station MAC list is captured; these packets are processed and delivered to the local network card(s) which connects the mobile node computer(s). Similarly, in the uplink direction,*

each packet having the source address included in the Mobile Node MAC list is captured; these packets are delivered to the local network card(s) which connects the base station computer(s).

To insert a MAC address in the list this address have to be typed in the edit space of the „Edit MAC Address List” window and then the „Add” button must be pushed – see fig. 3.3. The window could be closed, leading to a return in the main interface, by pushing the „Return” button or by clicking on the X sign from the right upper corner. To remove an address from the list, select that address with the „mouse” and then push the „Remove” button – see fig. 3.3.

The compensation of the possible supplementary delays inserted by the computer which runs the emulator requires the adjustment of the transfer rate. This operation uses a data generator with adjustable constant bit rate, which has to be used in an emulation process of channel with high signal to noise ratio, channel witch has no errors and for which the exact average bit rate is known (see paragraph 3.1.1. related to the possibility to select a certain signal to noise ratio for emulation).

The adjustment of the bit rate is accomplished with the „Scroll Bar” called „Bit rate adjustment” and the adjustment value is displayed in the „Adj_value” edit box. The adjustment value has no direct significance; it just indicates how great is the deviation that has to be corrected, relatively to the ideal situation when the computer does not insert supplementary delays. See also chapter 8. that deals with the adjustment of the bit rate transferred through the emulator.

The edit box „Emulation Statistics” allows the selection of the file in which the tables with statistics corresponding to the configuration of the emulated system are stored. The name and the path of the file containing the statistics have to be introduced in this edit box. If the path and/or name are incorrect, an error message is displayed and the emulation process is stopped. The path and name of the file can be reintroduced after pushing the “Start/Stop” button.

The „Average Bit Rate” edit-box displays the average bit rate, in kbps, computed over a time interval (expressed in seconds) specified in the „Averaging interval” edit box. The „Transm. bytes” edit box displays the number of transferred bytes, in Kbytes, starting from the beginning of the emulation or from the moment when the “Clear tr. bytes” button is pushed, when the counter associated to number of transmitted bytes is cleared – see fig. 3.3.

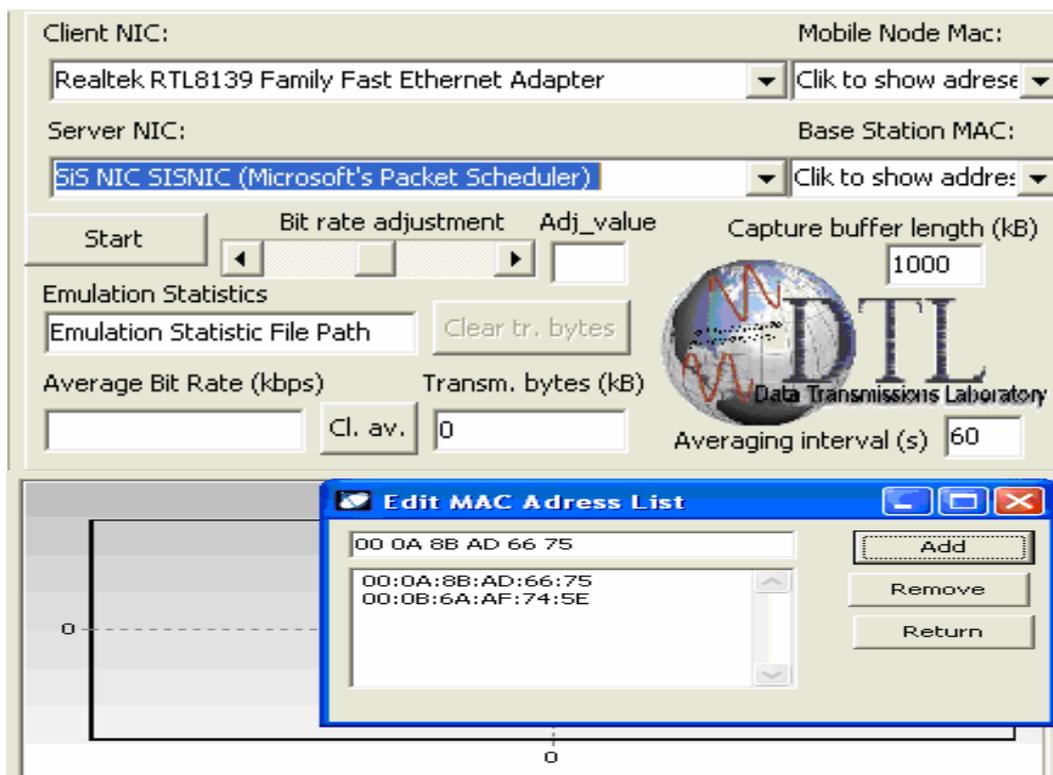


Fig. 3.3 User interface of the command and control block.

The parameters associated to the command and control block are read and the emulation process is started by pushing the “Start/Stop” button.

When this button is pushed the following actions are performed:

- the emulation parameters are read from the interface;
- the file containing the statistics required by the emulation is read and the data structures storing these statistics during the emulation process are built; these structures are built in such a way to allow a fast generation of the random variables associated to the instantaneous bit rate and to the error distribution – see paragraph 5.2.2. related to the generation of the statistics tables and paragraph 3.2.2. related to the generation of random variables distributed according a given pdf function);
- the communication channels to the network cards associated to the emulator, which receive/transmit data packets, are open;
- the packet-capture block is initialized;
- the threads that emulate the transmissions in the downlink and uplink directions are launched.

3.1.3. The emulation statistics display block

In parallel with emulation of the transmission (delay of the received packets and error insertion in these packets), some significant statistics related to the emulation process are displayed.

Three graphs included in the user interface are displayed, namely the evolution in time of the average bit rate, the time-evolution of the number of inserted errors and of the average delay of the packets passing through the emulator. These graphs allow the correlation between the position of the main user (the emulated user) inside the cell, which determines its instantaneous SNR, and the main parameters that define the transmission quality.

3.2. Main operations performed in the emulation process

The operations performed by the emulator are grouped in 3 separate threads [Thr] namely:

- the main thread (the main program) – performs the operations related to the SNR calculation, as explained in paragraph 3.1.1. dedicated to the signal to noise ratio computation block. This operation uses the system timer, with a timing of 1s, to generate the real time. The **actualization** of the mobile position and implicitly of the SNR value is performed at each second;
- the thread associated to the uplink connection – performs in the present version only the filtering and interception of the packets received from the computer (or computers) acting as mobile station(s). The captured packets are transferred at the network card which ensures the connection with the computer (or computers) acting as base station(s) without any delays and errors. The speed in the uplink is limited only by the processing time of the packet filtering and capturing operations;
- the thread associated to the downlink connection – performs the filtering and interception of the packets received from the computer (or computers) acting as base station(s), the delay of the received packets and the error insertion according to the appropriate statistics which characterize the emulated situation;

If the emulation process is stopped, the packets are only intercepted and transmitted to the network card ensuring the connection with the computer (or computers) acting as mobile station(s).

If the emulation process is enabled, the intercepted packets are read from the reception buffer according to the instantaneous bit rate, i.e. the numbers of bytes calculated according to the pdf of the instantaneous bit rate are read in each TTI interval and then errors are inserted in these bytes according to the distribution of the errors.

These thread computes the average bit rate on a specified time interval (see paragraph 3.1.2. dedicated to command and control block) and uses a timer with a 1s timing; the procedure associated to this timer displays this average bit rate and the evolution in time of this average value. The same procedure also computes the average delay per packet/second and displays the evolution in time of this value.

The timing required by the generation of the instantaneous bit rate, meaning the generation of the TTI intervals, uses the high precision timer of the system, timer which can be accessed by the following functions [Que]:

- „*QueryPerformanceFrequency(*LARGE_INTEGER)*” – gives the frequency of the high precision timer;
- „*QueryPerformanceCounter(*LARGE_INTEGER)*” – reads the counter associated to this timer;

3.2.1. Interception and filtering of the processed packets

The solution used for implementation of the present emulator is based on interception of MAC Ethernet packets, approach that ensures independence of the higher-layer protocols. The packet-interception is accomplished by using the functions of the WinPcap [Wpc] library for the Windows operating system. The intercepted packets are stored in buffers managed by the WinPcap functions, their lengths being controlled from the user interface of the command and control block. This library also includes special functions for packet-filtering at the kernel level, facility used in the present implementation the decrease the processing time.

The functions of the WinPcap library that are used in the present implementation are the following (for supplementary details see [Wpc]):

*int pcap_findalldevs (pcap_if_t **alldevsp, char *errbuf);*

- generates a list of network devices which can be open with the “*pcap_open_live()*” function; for each device it associates a “*pcap_if_t*” descriptor, which is a structure that contains information related to the network device that can access it; *errbuf*, is a string which contains a possible error message; the function returns 0 for successful operation and -1 otherwise;

*pcap_t pcap_open_live (const char *device, int snaplen, int promisc, int to_ms, char *ebuf);*

- generates a „packet capture descriptor” *pcap_t* ; this is a structure which contains data necessary to access a network device and to perform a packet capture; *device* – is the name of the accessed network device; *snaplen* – specifies the maximum number of captured bytes; *promisc* – specifies the functioning mode of the network device during the capture; *to_ms* – specifies the „read timeout” in ms;

*int pcap_sendpacket (pcap_t *p, u_char *buf, int size);*

- allows to send a “raw” packet (a full packet containing all the protocol headers, loaded in the packet by the sending application); the pointer *p* of type *pcap_t* describes the network interface and the capture process; *buf* – is the buffer which stores the bytes to be loaded in the packet (the entire structure of the packet excepting the CRC, which is computed by the network interface); *size* – length of the packet which will be transmitted;

*int pcap_next_ex (pcap_t *p, struct pcap_pkthdr **pkt_header, const u_char **pkt_data)*

- the function is used to capture the next available packet; *pkt_header* – is a pointer to a *pcap_pkthdr* type structure, which contains the header of the captured packet, *pkt_data* – is a pointer to a string containing the data bytes of the captured packet;

*void pcap_close (pcap_t *p);*

- frees the memory and resources related to the packet capture process;

*int pcap_compile (pcap_t *p, struct bpf_program *fp, char *str, int optimize, bpf_u_int32 netmask)*

- ensures the compilation of a packet filter, converting a high level filtering expression into a program which can be interpreted by the kernel-level filtering procedures; for more details, see “Filtering expression syntax” [Wpc];

*int pcap_setfilter (pcap_t *p, struct bpf_program *fp)*

- associates a filter to a capture process – details can be found in [Wpc];

3.2.2. Delay of the intercepted packets and error insertion in these packets

The emulation of the packet-delay is accomplished by the generation of the instantaneous bit rate; the intercepted data are transferred from the buffer of the input network-card to the buffer of the output network-card in the rhythm of this instantaneous bit rate. In this way, the packets delays are generated automatically. This is practically accomplished by generating the number of bits/TTI at the beginning of each channel coherence time interval; additional details can be found in paragraph 5.2.2. that deals with the generation of the tables with statistics in the simulation process. The error-patterns inserted in the intercepted (and filtered) packets are described by two statistics; one of them contains the number of errors/TTI and the other one, the effective error-distribution (localization) inside that error packet.

The values taken by the number of bits/TTI and by the number of errors/TTI are jointly processed by means of a discreet bi-dimensional random variable described by a finite set of “no. errors/TTI – no. bits/TTI” pairs, which is obtained after processing the initial data provided by the simulation. A cumulative probability function (cdf), is associated to the above mentioned pairs, see fig. 3.4.; for details see paragraph 5.2.2.

This distribution is represented (stored) using 3 tables, two of them containing the values of no. errors/TTI and respectively no. bits/TTI, i.e. the previously mentioned pairs; the third table contains the values of the cumulative distribution function (cdf) associate to these pairs.

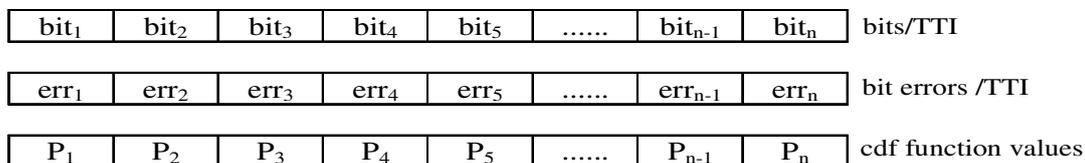


Fig. 3.4. Representation of the bi-dimensional random variable no. errors/TTI – no. bits/TTI.

To generate a pair of random variable according to an imposed probability function the following steps have to be performed:

- a uniformly distributed variable within the interval [0 , 1] is generated; this could be accomplished by using a random number generator, e.g. the generator offered by the programming environment. Then, the generated values are divided by the maxim value that was be generated;
- the P_a and P_{a+1} values of the cdf function which frame the currently generated value of the uniform random variable are established; the minimum and maximum possible values of the cdf function, i.e. 0 and 1, may not be stored in the tables;
- the values err_a and bit_a, meaning no. errors/TTI and no. bits/TTI corresponding to value P_a of the cdf function which lower-bounds the currently generated value of the uniform random variable, are extracted from the appropriate tables; more details can be found in [Bot02] [Bot00]. The number of bits/TTI is transformed in number of bytes/TTI for an easier generation of the instantaneous transfer rate;

The error-patterns generation has to consider not only the number or errors/TTI, but also the distribution pattern of these errors (the error-localization). For this distribution pattern, a second statistics which describes the distance between consecutive bit errors is used, more details being available in paragraph 5.2.2. This statistics is represented by two tables, one containing the

quantized values of the distance between consecutive errors and the other one containing the values of the associated cumulative density function – see fig. 3.5.

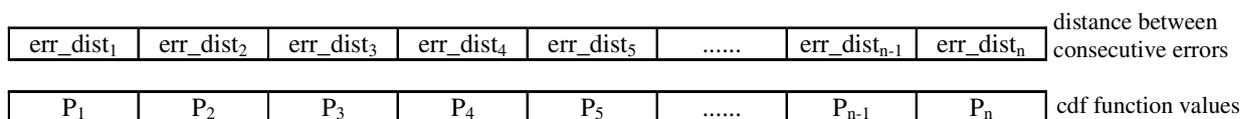


Fig. 3.5. Representation of the random variable which gives the distance between consecutive errors.

The values of this random variable are generated according to the previously presented algorithm. A number of distances, equaling the number of errors/TTI, are generated at first, then the algorithm computes the error-positions and finally, the values of the bits located in these positions are changed. The search of the error positions starts at the beginning of the group of bits corresponding to the current TTI interval and a modulo (no. bits/TTI) arithmetic is used; if an error-position index is greater than the length of the group of bits corresponding to the current TTI interval, then the search for this position continues circularly from the beginning of the group. Such a situation could occur due to the fact that the number of errors and the positions of these errors, i.e. the distribution pattern of these errors, are characterized by different non-correlated distributions.

4. The parameters of the LTE simulator

Two categories of users (mobile stations) are considered in the simulations that provide the statistics which are employed by the emulator:

- one main user, for which the simulation is actually performed;
- a group of secondary users, which makes up the loading of the cell carrier;

The delay (bit rate) and error-probability statistics are built up only for the main user, by performing in the simulator all the operations (processing) included in the transmission chain.

For the secondary users, the simulator performs only the processing required by the generation of the characteristic of the radio channel and those involved in the allocation of the radio resources, which affects the allocation of the radio resources for the main user who is actually emulated; this approach allows the simulation of the effect of the cell-carrier loading.

The description of the processing performed for the two types of users requires a previous definition of the data structures employed in the transmission process, the specification of the modulation and coding techniques employed, the definition of parameters that characterize the cell and the specification of the parameters that individually define each user.

4.1. Parameters that characterize the cell in the simulation process

The parameters that characterize the cell are defined in a distributed manner between the simulator and the emulator, as shown in chapter 3. which describes the operating principle of the emulator.

This way of defining the parameters is employed to reduce as much as possible the number of simulations, and implicitly the number of statistics-files, required to perform the emulation; this approach ensures also a reasonable amount of processing performed by the emulator. To accomplish this goal, the simulator makes use only of the parameters that are independent of the cell-geometry and of the base-station characteristics; this approach allows for the employment of the statistics delivered by a simulation in more emulations.

In the simulation process, the cell is practically characterized by:

- the number of active users;
- the multipath propagation profile of the channel;
- the radio-channel carrier frequency (cell-carrier);
- the parameters of OFDM modulation;
- the probability density function of the fast-fading inserted by the users' motion;

The limits of SNR domain of the signal received by the main user and the limit-values of the attenuation of the signal received by the secondary users also describe, in an indirect manner, the cell employed in the simulation process. All the above parameters can be set in the user-interface, except for the probability density function of the fast fading, which is implemented as a Rayleigh fading.

The multipath propagation profile is specified by the number of the secondary propagation paths and by the delays and attenuations inserted by these paths, referred to the first arrived path (the main path).

The main path is considered to be the first arrived one, with a minimum delay, and, since it is a reference path, its relative delay and attenuation are not explicitly specified, being equaled to zero. The absolute delay of the reference path is of little importance for the performances of the physical level; it could only affect the timing aspects of the higher-level protocols. As for the absolute attenuation of the first path, it is inserted by the large-scale fading, and is computed and considered in the emulation process. *If required, the absolute delay inserted by the main path might be inserted in a subsequent version of the emulator.*

The multipath propagation profile is combined with the fast fading generated by the user's motion, each individual propagation path being affected separately by a Rayleigh-distributed

fading. Additional information regarding the generation of the mobile multipath radio channel are presented in paragraph 5.1.

| | | | | |
|--|---------|------------|------------|---------|
| <input checked="" type="checkbox"/> Mobile Channel | | Delay (ns) | Atten (dB) | |
| Central frequency [Hz] | 50e8 | Path 1 | 10 | 5.22 |
| OFDM symbol rate [Hz] | 39062.5 | Path 2 | 30 | 6.98 |
| Guard interval [ratio] | 0.125 | Path 3 | 360 | 5.2204 |
| No. of secondary paths | 17 | Path 4 | 370 | 7.4404 |
| No. of payload subcarriers | 416 | Path 5 | 385 | 9.2004 |
| | | Path 6 | 250 | 4.7184 |
| | | Path 7 | 260 | 6.9384 |
| | | Path 8 | 280 | 8.6984 |
| | | Path 9 | 1040 | 8.1896 |
| | | Path 10 | 1045 | 10.4094 |
| | | Path 11 | 1065 | 12.1696 |
| | | Path 12 | 2730 | 12.0516 |
| | | Path 13 | 2740 | 14.2716 |
| | | Path 14 | 2760 | 16.0316 |
| | | Path 15 | 4600 | 15.5013 |
| | | Path 16 | 4610 | 17.7213 |
| | | Path 17 | 4625 | 19.4813 |

Fig. 4.1. Specification of OFDM and multipath propagation profile in the user interface of the simulator.

The OFDM modulation is specified by the following parameters: the carrier (cell carrier)-frequency, the frequency-separation between sub-carriers, the number of sub-carriers used for transmission and the guard-interval of the OFDM symbols. Figure 4.1. shows the user interface of the simulator used to set the specifications for the parameters of the OFDM modulation and of the multipath propagation profile. The particular multipath profile presented is the one of WP5 channel model [Win05_1], with 18 propagation paths, for a carrier frequency of 5GHz.

4.2. Parameters that characterize the users in the simulation process

Each user mobile station is defined by a set of four parameters, namely:

- **the user’s moving speed**, employed in the generation of the small-scale fading. Actually it is employed for the computation of the coherence time T_c , (2); additional details can be found in paragraph 5.1. The coherence time is also used for the simulation of the displacement of the secondary users by changing the attenuation of the signal received by such a user in the rhythm given by its corresponding coherence time. Since the coherence time is lower-bounded by the parameters of the OFDMA scheme (TTI), the maximum user speed should be limited accordingly, so that the coherence time should not be too small.
 - ❖ *Note: the current version of the emulator does not check if the condition $T_c > TTI$ is fulfilled.*
- **the absolute attenuation of the main propagation path**, which is specified for the secondary users to allow the simulation of their position, relative to the position of the main user. The attenuation of the first path of the main user is considered 0 dB, the absolute attenuation of his main path being computed within the emulation process, see chapter 3. , paragraph 3.1.1. dedicated to the computation of SNR_0 within the emulator. Note that the relative attenuations of the secondary users might have negative values,

indicating that their position is closer to the center of the cell (base-station) than the one of the main user; additional details are to be found in chapter 5. , which describes the operating principle of the simulator.

- **the class and priority of the service offered to the user** – an unlimited number of services may be defined, the services with smaller indexes having higher priorities. The Winner project [Win05_2] defined 21 classes of services and 24 generic applications. Those definitions are a basis for the definition of a set of possible services. The priority and bit rate assigned to each service are employed in the process that allocates the radio resources to each user, as shown in paragraph 5.2.1. that describes the scheduler block;
- **the average bit rate associated to the service provided to the user.**

Figure 4.2. displays the simulator interface employed for the setting of the user parameters; the example shown considers 5 users on the cell carrier.

| | | | | | |
|----------------------|-----|-------------|------------|----------|----------------|
| No. of users in cell | 5 | | | | |
| | | Speed(km/h) | Atten.(dB) | Priority | Bit_rate(kbps) |
| User1 | 120 | 0 | 0 | 150 | |
| User2 | 120 | 3 | 1 | 250 | |
| User3 | 120 | 3 | 2 | 500 | |
| User4 | 120 | 3 | 3 | 700 | |
| User5 | 120 | 3 | 2 | 400 | |

Fig. 4.2. Setting the users' parameters in the simulator interface.

4.3. Parameters that define the modulation and FEC coding techniques and the OFDM access method in the simulator

Because the modulation and FEC encoding methods employed on the OFDM sub-carriers and the data structures employed in transmission are strongly correlated, their parameters are defined in the same window.

4.3.1. Parameters of the adaptive modulations employed on OFDM sub-carriers

Each OFDM sub-carrier is modulated with a square QAM or a 2-PSK constellation defined by the number of bits/symbol. The accepted numbers of bits/symbol are 1, 2, 4, and 6 and optionally 8, i.e. the signal constellations employed are respectively 2PSK, QPSK, 16QAM, 64QAM and optionally 256QAM.

The above mentioned modulations are employed on groups of sub-carriers (OFDM chunks – see chapter 2.). The adaptive employment of these modulations require the setting of some SNR domains, within which each modulation should be employed according to a given criterion (bit-error rate or symbol-error rate), these SNR domains being separated by SNR thresholds. The simulator interface allows the setting of the SNR thresholds, but the numbers of bits/symbol can not be modified, since only the modulations mentioned above can be employed. Actually, the interface allows the modification of their values, e.g. to odd values, but the modification has no effect, since the numbers of bits/symbol are imposed to be either even or 1. The interface window that allows the setting of the SNR domains for each modulation is presented in the left-hand side of figure 4.3. The 256-QAM constellation can be validated in the corresponding check-box.

4.3.2. Parameters of the error-correcting codes employed

The simulator employs LDPC FEC codes, see [Gal03] [Mck99], and more specific it employs regular $L(2,q)$ -LDPC codes, as defined in [Ele02] and [ITU]. These codes are characterized by three parameters K , J and P , P being a prime number, K , $J \geq 3$ and $J < K \leq P$. The codeword length equals $K \cdot P$ and the number of check bits is given by the product $J \cdot P$.

The length of an LDPC codeword can be modified by shortening the code, i.e. the removal of a number of information bits, which leads to a decrease of the code’s coding rate. The codes employed are shortened, so that the lengths of the codewords would match the lengths of the transport frames, L_{frame} . The coding rate of the shortened codes, R_c , is expressed by:

$$R_c = \frac{L_{frame} - J \cdot P}{L_{frame}} \quad (3)$$

| | Threshold (dB) | Bits/symbol |
|-----------|----------------|-------------|
| config. 1 | -100 | 1 |
| config. 2 | 10.2 | 2 |
| config. 3 | 17.4 | 4 |
| config. 4 | 23.6 | 6 |

| | Frame/code index | Frame_length | K | J | P |
|--|------------------|--------------|----|----|----|
| | 1/1 | 81 | 11 | 1 | 11 |
| | 1/2 | 81 | 11 | 2 | 11 |
| | 1/3 | 81 | 11 | 3 | 11 |
| | 1/4 | 81 | 11 | 4 | 11 |
| | 1/5 | 81 | 11 | 5 | 11 |
| | 2/1 | 162 | 13 | 1 | 13 |
| | 2/2 | 162 | 13 | 3 | 13 |
| | 2/3 | 162 | 13 | 5 | 13 |
| | 2/4 | 162 | 13 | 7 | 13 |
| | 2/5 | 162 | 13 | 9 | 13 |
| | 3/1 | 324 | 19 | 1 | 19 |
| | 3/2 | 324 | 19 | 3 | 19 |
| | 3/3 | 324 | 19 | 6 | 19 |
| | 3/4 | 324 | 19 | 8 | 19 |
| | 3/5 | 324 | 19 | 11 | 19 |
| | 4/1 | 486 | 23 | 1 | 23 |

Fig. 4.3. Setting the parameters of the adaptive modulations, of the error-correcting codes and of the data structure in the simulator interface.

Five codes are defined by default for each length of the transport frame, but more codes can be defined in simulator.

The codes employed for various lengths of the transport frame are selected so that they should ensure about the same coding rate.

The parameters „Initial_rate_index” and „No_retransm” see figure 4.3., are related to the H-ARQ procedure, [Fau06] [See06] [Val06].

The first parameter indicates the code that is employed in the first transmission of a transport frame, while the second shows the maximum number of retransmission attempts of a transport frame, if bit-errors occur. For each retransmission the FEC code is changed, by using the next code on the list, code that has smaller coding rate. The maximum number of retransmission should be smaller with 1 than the number of codes defined for each transport frame length. The number of transport frame lengths (“No_transp_frames”), the number of FEC codes employed for each length of a transport frame (“No_codes/frame”), and the parameters of the FEC employed can be set in the simulator interface, see fig. 4.3., by filling in the appropriate boxes. Several default groups of codes, defined for several transport frame lengths are also presented in fig. 4.3.

- ❖ *Note: the selection of the set of LDPC codes employed for each L_{frame} (and for each modulation) requires a careful evaluation of the throughputs provided by the resulting configurations (code + modulation).*

4.3.3. Parameters of the data structures employed in transmission

Two data structures are employed in the transmission process:

- the chunk (or bin) – a group of adjacent sub-carriers during a number of OFDM-symbol periods;
- the transport frame – composed of more chunks, spread in the frequency domain, in the same chunk interval

Inside each chunk, the same modulation (and possible codes) is employed on all sub-carriers and during all OFDM-symbol periods of that chunk. The chunks that compose a transport frame may employ different configurations (modulation). Additional information regarding these issues can be found in chapter 2. dealing with OFDMA multi-user access.

The parameters „Sbcarrs/chunk”, „OFDM Symb./chunk” set the chunk dimensions and parameter „Service symb./chunk” sets the number of sub-carriers (or scattered QAM symbols) within a chunk, which are employed for service purposes (synchronization, equalization, channel-measurement and are not payload symbols).

The transport frames are defined only by their length (in bits), the exact structure (in chunks) being set dynamically by the resource allocation procedure (algorithm).

Figure 4.3. presents an example of setting the data structures employed in transmission by the simulator.

4.4 The main parameters of the simulation

The main parameters that define a simulation process are the following:

- the number of transport frames transmitted by each user (the main user and the secondary ones) during the simulation interval. The delay (or more specific the instantaneous bit-rate) and bit-error statistics of the main user are obtained using the bits transmitted by this user within the specified number of transport frames. Note that the numbers of bits transmitted by various users are not equal; they are depending of characteristics of the radio channel „seen” by each of them and of parameters of the service that is provided to each of them;
- the maximum number of iterations performed by the MP („Message Passing”) decoding algorithm of the LDPC code, [Gal63], [Mck99]. The coding-decoding processes are performed only for the main user; therefore this parameter is significant only for this user;
- the range of the SNR values of the first arrived propagation path, SNR_0 , across which the simulation is performed, parameter that is specified only for the main user. This range is specified by a minimum and maximum SNR_0 values and by a sweeping step. A distinct simulation transmitting the specified number of transport frames is performed for each SNR_0 value inside this domain; distinct statistics tables are generated for each simulation, see beginning of chapter 5. and paragraph 5.2.2. for additional information regarding the simulation’s principles and the way the statistics tables are generated;
- the range of the channel attenuation (more specific of the first arrived path) for the secondary users. The range is defined by the minimum and maximum values of attenuation and by a sweeping step. The sign of the sweeping step changes independently for each user, when the current value of the attenuation reaches one of the limit values imposed. The attenuation is modified with a different rhythm for each user, rhythm that is correlated to the user’s speed and to the user’s coherence time. During each coherence interval this attenuation is modified with a sweeping step (one unit); see additional details in chapter 5. that deals with principles of the simulation process.

Figure 4.4 shows the window that allows the setting of these parameters and their default values, as well as the main functional buttons of the simulation program.

The „Read Settings” button reads the set values of all above mentioned parameters and initializes the tables/matrices required by various processing.

The „Start” button calls the program sequences that actually perform the simulation; see the beginning of chapter 5. for the structure of the simulation program and paragraph 5.2. for information regarding the processing performed throughout the simulation. To obtain reliable error statistics, each simulation should be performed on a great number of transport frames; the minimum number of frames should be at least 10000.

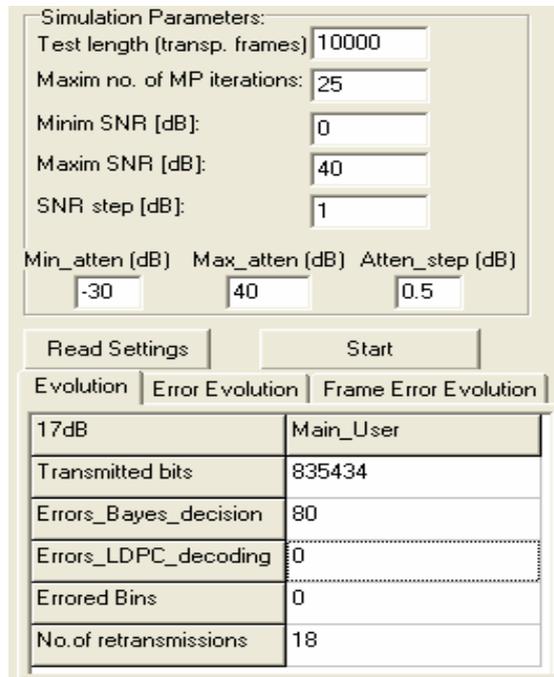


Fig. 4.4. Setting the main simulation parameters and display of intermediary simulation results in the user interface

4.5. The output data delivered by the simulator

The simulator delivers the files that contain the statistics of the delay and bit error-rate. The simulation results are delivered in two files:

- the first one named „stat_file.txt” is a text file employed for the analysis of the generated statistics;
- the second one, named „stat_file.sta”, is a binary file (file stream) which contains the cumulative distribution functions (cdf) associated to the two mentioned random variables. These functions are employed by the emulator to insert the delays and errors in the emulation process, see paragraph 3.2.2. describing the delay and error insertion in the intercepted packets.

The ways the data are structured in these files are presented in paragraph 3.2.2., as well as in paragraph 5.2.2., which explains the generation of the statistics tables. Each simulation generates files with imposed name and extension; previously generated files are overwritten.

❖ *Note: the user should rename the files generated by previous simulations, to avoid the loss of the information they contain.*

The simulator also displays some intermediary data during the simulation process, which are employed for the control of this process. It displays dynamically for the main user the number of transmitted bits, the number of bit-errors (with and without the employment of the error-correcting codes), the number of retransmissions and the number or incorrectly decoded transport frames, as shown in figure 4.4.

5. The principle of the simulation process

The design of the simulator has as a main goal the generation of instantaneous bit rate (delay) and bit error probability statistics which can be reused in several emulation processes, decreasing in this way the simulation-time and the amount of memory required to store the mentioned statistics. An important aspect is the removal of the parameters related to the cell geometry and to the base station transceiver from the simulation process; this approach allows the use of the same statistics to emulate the transmissions in cells with different characteristics. To achieve this goal, the simulation process uses as the main parameter the signal to noise ratio, of the path with the lowest delay of a multipath propagation scenario, SNR_0 . The simulations are performed across a range of signal to noise ratios, with an acceptably small separation step, each simulation generating a set of statistic tables. During the emulation process the SNR_0 is calculated and the appropriate statistics tables are chosen – see first part chapter 3. and paragraph 3.1.1., related to the operating principle of the emulator and to the calculation of the SNR_0 value. The multipath propagation model and the cell load, meaning the number of users and the offered services, are imposed in the simulation process. Practically, the simulations must be performed for a limited, but representative, set of multipath propagation models and cell loads; a limited set means a number of several tens or hundreds scenarios. The dimensions of the statistic tables are relatively small, tens of kB, allowing the storage of a large number of such statistics on usual memory circuits.

The simulation process considers two types of users, namely the main user, for which the tables with instantaneous bit rate and error probability statistics are generated, and a set of secondary users, which give the load of the cell carrier and which occupy radio resources.

The simulation for a number of transport frames great enough to provide highly reliable statistics are performed for each value of SNR_0 of the main user. The number of the transport frames has to be at least of several thousands; a larger number of transport frames ensure more reliable statistics, but also increases substantially the simulation time.

This approach is equivalent to the movement of the main user on a circle with the center located in the base station and the radius imposed by the value of SNR_0 and the used propagation model, model which depends on the cell type. The total signal to noise ratio used in the simulation process is computed based on the SNR_0 value, on the multipath propagation profile, on the distribution of the fast fading due to the mobile motion, taking also into account the channel coherence time.

The motions of the secondary users are simulated by the modification of an attenuation parameter associated to each secondary user; this parameter indicates the distance to the base station transceiver. It changes for each secondary user in the cadence of the channel coherence time specific to that user, i.e. a function of the user's speed. This approach simulates a real case, where the main user moves on a circular trajectory around the base station and the other users move randomly inside the cell. In the emulation process the main user motion trajectory is changed, by moving him from one circle to another. The basic idea of the simulation process is presented in fig. 5.1.

The attenuation parameter of the secondary users have to be located in a large domain (-30, + 30dB), to allow a realistic simulation of the motion between the center and the edge of the cell. This domain must include both positive and negative values, to allow the simulation of the relative positions of the secondary users relative to the main user (a positive attenuation means that the secondary user is located farther to the center of the cell than the main user, while a negative attenuation means that the secondary user is located closer to the center of the cell than the main user). The calculation method of the total signal to noise ratio is presented in paragraph 5.1. This signal to noise ratio includes the effect of the multipath propagation and of the motion generated fast fading and is used for radio resource allocation, both for the main user and the secondary users.

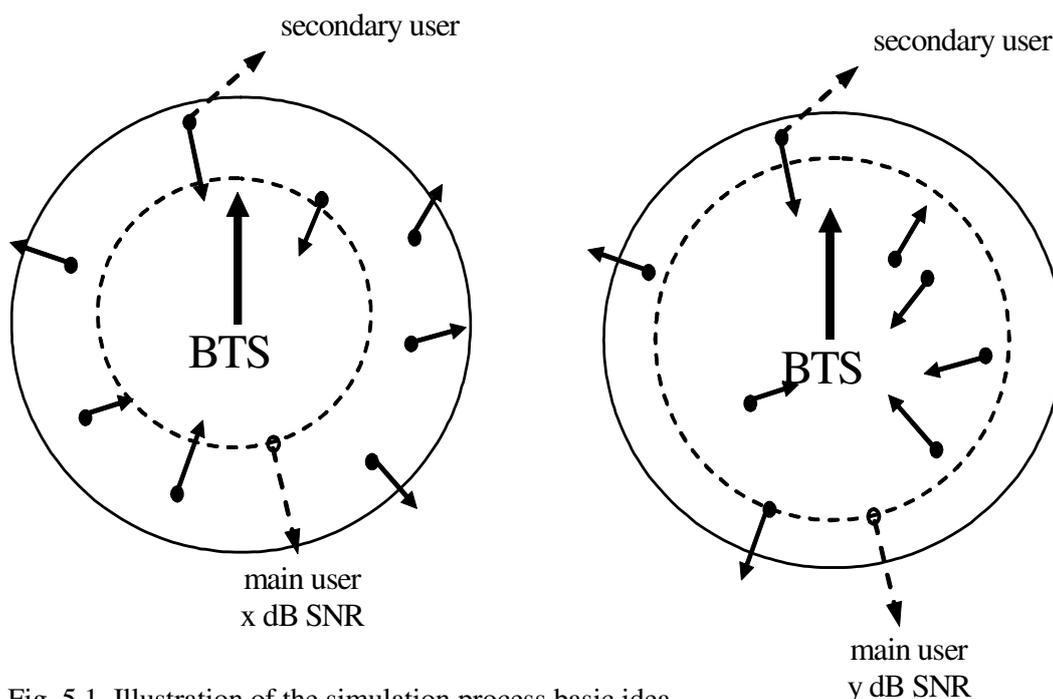


Fig. 5.1. Illustration of the simulation process basic idea.

5.1. The channel model employed in the simulation process

The simulation process considers both the effects of the large scale fading, of the multipath propagation and of the small scale fading. The large scale fading is simulated by the signal to noise ratio associated to the first propagation path (with the lowest delay); the multipath propagation and the small scale fast fading are considered together, each separate propagation path being affected by a Rayleigh type fading. The channel model used is presented in fig. 5.2., and it is identical with the model proposed in [Rap01].

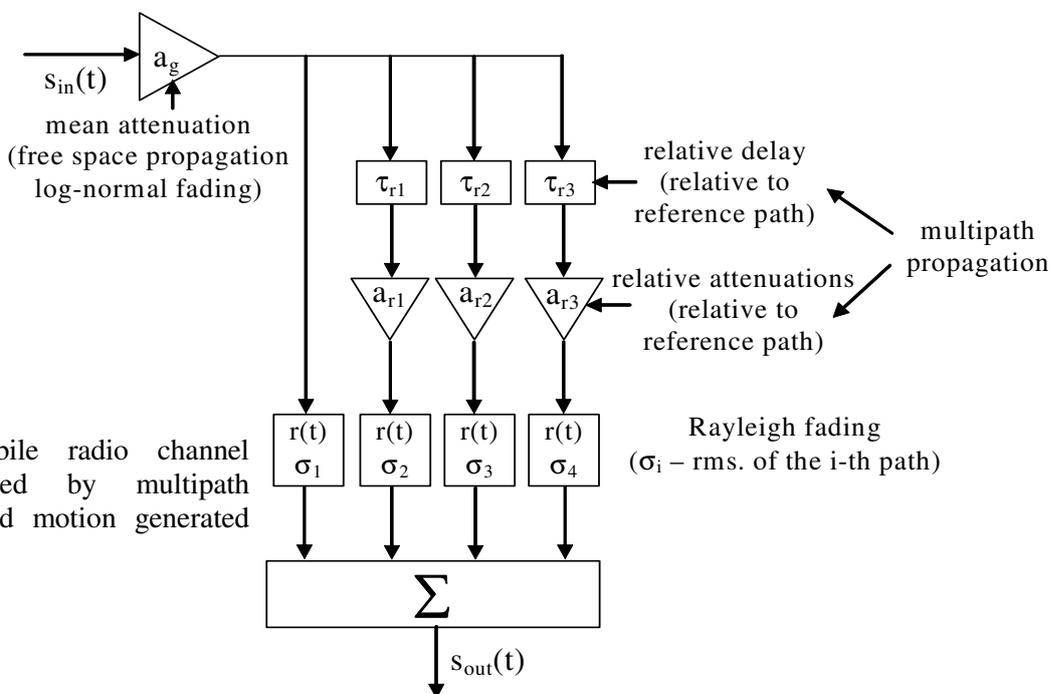


Fig. 5.2 Mobile radio channel model, affected by multipath propagation and motion generated fast fading.

The Rayleigh type probability density function is given in (4). The relation gives the probability density of the Rayleigh faded signal envelope, r , and σ^2 represents the power of the signal affected by Rayleigh fading.

$$p(r) = \begin{cases} \frac{r}{\sigma^2} \cdot \exp\left(-\frac{r^2}{2 \cdot \sigma^2}\right) & ; 0 \leq r \leq \infty \\ 0 & ; r < 0 \end{cases} ; \frac{\bar{r}^2}{2} = \sigma^2 \quad (4)$$

To calculate the global instantaneous signal to noise ration at a given moment the following procedure is us:

- the amplitude of the first propagation path (the main path) is considered to equal 1 and (a_i, τ_i) are the attenuations (expressed as ratios) and the relative delays of the other multiple propagation paths;
- the amplitudes of the secondary propagation paths will be a_i , and the phases of these paths, relative to the one of the first path are given by:

$$\phi_i = 2\pi \cdot f \cdot \tau_i \quad (5)$$

where f is the signal frequency; (5) considers only one spectral component, the channel model being applied separately to each spectral component;

- ❖ Note: for a single carrier narrow-band transmission, a flat fading in the signal frequency band should be considered since the amplitude variation of the carrier is quasi-identical to the variations of the other spectral components of the modulated signal. For a multi-carrier transmission (OFDM), the global fading which affects the signal with multipath propagation and implicitly the instantaneous value of the SNR, have to be computed for each sub-carrier or for each chunk, if the sub-carriers are grouped.
- The global fading which affects the signal with multipath propagation, i.e. the amplitude variation of this signal is computed with the following algorithm:

$$\begin{aligned} r_{g_cos} &= f_a(1/2) + \sum_{i=1}^N f_a(a_i^2/2) \cdot a_i \cdot \cos(\phi_i) \\ r_{g_sin} &= \sum_{i=1}^N f_a(a_i^2/2) \cdot a_i \cdot \sin(\phi_i) \\ r_g &= \sqrt{r_{g_cos}^2 + r_{g_sin}^2} \end{aligned} \quad (6)$$

where r_g is the amplitude of the signal affected by the multipath propagation and fast fading, and $f_a(p)$ is the value of the Rayleigh fading corresponding to a signal with power level p ;

- The total instantaneous SNR of the main user, SNR_p , and those of the secondary users, SNR_{s_i} , are given by the following relations:

$$\begin{aligned} SNR_p &= SNR_{p_0} + 20 \cdot \log_{10}(r_{g_0}) \\ SNR_{s_i} &= SNR_{p_0} + 20 \cdot \log_{10}(r_{gs_i} \cdot a_{s_i}) \end{aligned} \quad (7)$$

where r_{g_0} is the signal level of the main user, SNR_{p_0} is the signal to noise ratio of the first propagation paths of the main user, r_{gs_i} is the signal level of the secondary user i , a_{s_i} is the attenuation associated to secondary user i (see chapter 5.), SNR_p is the instantaneous total signal to noise ratio of the main user, and SNR_{s_i} is the instantaneous total signal to noise ratio of the secondary user i ; the total SNR considers all the multiple propagation paths;

The total signal to noise ratios are calculated for each user and for each OFDM sub-carrier, finally generating a matrix G with a number of rows equal to the number of users, a number of columns equal to the number of sub-carriers; this matrix stores the instantaneous signal to noise ratios seen by each user in the entire bandwidth, which are applied to the radio resource allocation (scheduler) block.

- ❖ Note: considering that one chunk employs the same modulation, the number of bits/QAM symbol allocation process uses an average signal to noise ratio on the chunk sub-carriers.

The appendix presents some standardized multipath propagation channel profiles, for various frequency bands.

5.2. The structure of the simulation program and the main processing performed

The simulation program is composed of two major components: an initialization and a processing component; the latter performs the simulation processing involved by the transmission processes of the main and secondary users.

The initialization component performs the following tasks:

- reads the parameters specified in the user interface;
- builds up the mapping and demapping tables necessary for QAM modulation and demodulation;
- builds up the table required by the adaptive changing of the modulations employed;
- generates the control matrices of the error correcting codes and the auxiliary matrices employed in the encoding process;
- computes the channel coherence time associated to different users;
- performs the initialization of different processing blocks;

The processing component performs, for each value of SNR_0 , the operations briefly described below:

- simulation of the secondary users motion, by changing the values associated to the attenuation parameter;
- generation of the radio channel affected by the multipath propagation and fast fading both for the main user and the secondary users;
- allocation of the radio resources for all users, based on the characteristics of the associated radio channels and on the characteristics of the provided services;
- generation of the test data for the main user;
- encoding of the main user's data and generation of the appropriate transport frames;
- modulation of coded data on the OFDM sub-carrier and generation of the OFDMA access data structures;
- generation of the noise signal, according to the value of instantaneous SNR characteristic to each chunk associated to the main user transmission, and addition of the noise to the transmitted signal;
- demodulation of data in each chunk associated to the main user transmission, and reconstruction of the transport frames;
- decoding of the demodulated data;
- detection of post-decoding transmission error and implementation of the HARQ retransmission process; it modifies the used coding rate (actually decreases it) and resumes the transmission using the new coding rate;

- ❖ Note: Since the simulator has only to generate some statistics, the simulation of the retransmission process does not require the effective retransmission of the original

data, but only the modification of the coding rate and the inclusion of the retransmission index (number) in the computation of the final statistics.

- generation of the instantaneous bit rate and bit error probability statistics (distributions) based on the data obtained after the demodulation/decoding and HARQ processes;

Figure 5.3. presents schematically the processing chain performed by the simulator, briefly described in the previous points.

5.2.1. Allocation of radio resources. The scheduler block

The radio resource allocation to a certain user is realized using 4 parameters, namely:

- the value of the instantaneous signal to noise ratio, which is imposed by the fading (large scale and small scale) and the background noise/interference;
- priority of the service offered to the considered user;
- average bit rate associated to the service offered to the user;
- coding rate imposed by the HARQ process, and, consequently, generates the increased radio resources allocated for the transmission of the imposed instantaneous bit rate.

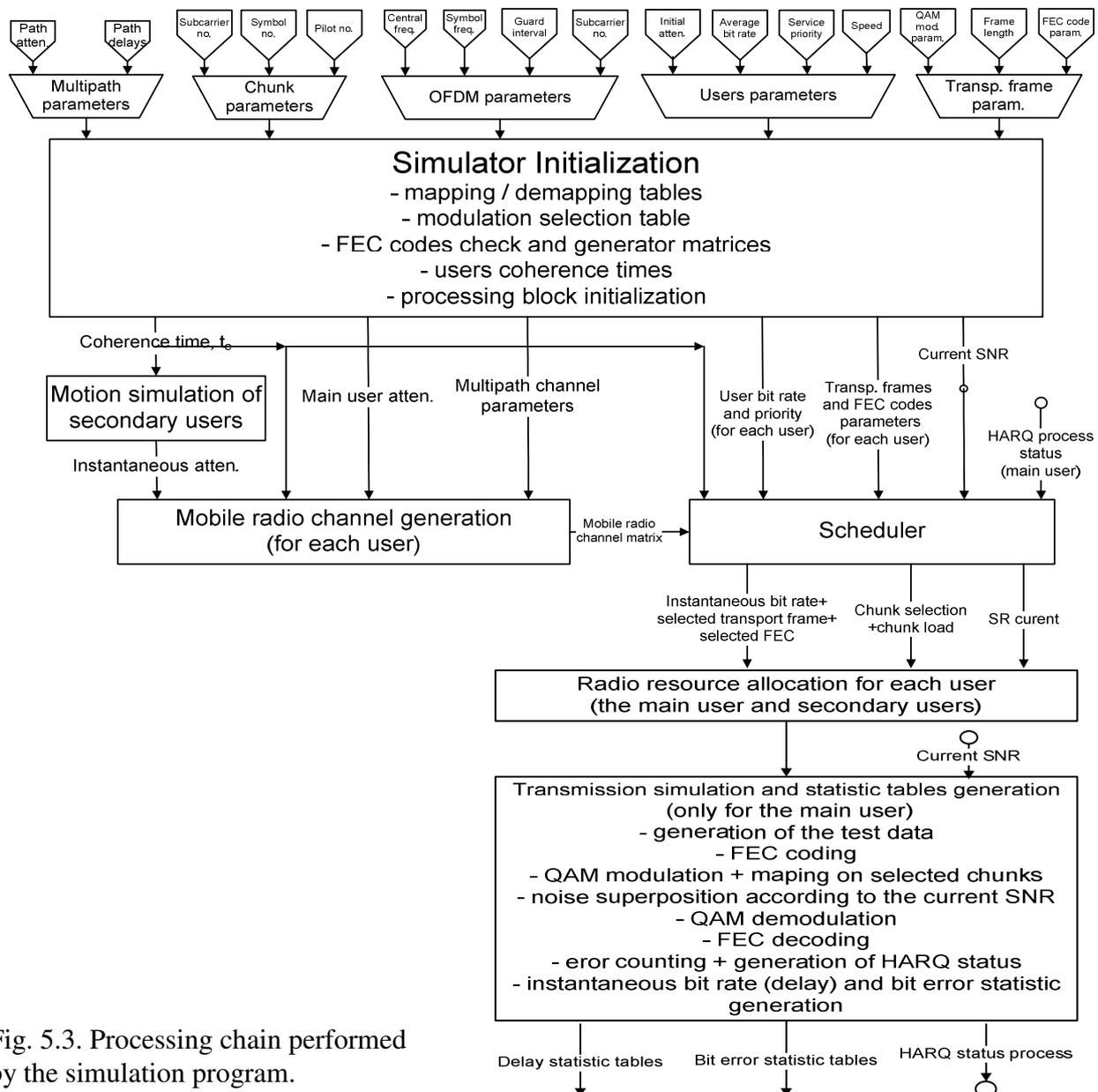


Fig. 5.3. Processing chain performed by the simulation program.

The radio resource allocation (the scheduling process) can not be performed separately for

each user, but it has to be made in parallel for each user, by taking into account their characteristic parameters, since all users compete for the same limited radio resources. It is a complex process which establishes broadly the global transmission performances of all users allocated to a cell carrier.

As for the priorities of the services associated to different users, the simulator allows the definition of a large number of services with different priorities in its interface (see paragraph 4.2. dedicated to description of the user's parameters). Some significant examples of service classes, together with the imposed requirements (bit rate, delay tolerance, maximum bit error rate, etc.) are defined in [Win05_2]. The current version of the simulator – emulator employs a simpler approach, namely:

- priority 0 (the highest) – telephony – low bit rate;
 - priority 1 – low bit rate streaming applications, e.g. medium quality voice and video applications;
 - priority 2 – high bit rate streaming applications, e.g. high quality voice and video applications;
 - priority 3 – file download – low and high bit rate – best effort type applications;
- ❖ Note: these categories of services constitute only examples of usual applications, the possible type of applications not being limited by the simulator.

There are a lot of implementation solutions of the scheduler block [Lar04] [Yag04] [Win05_1] [Ste03], implementations which depend mainly on the characteristics of processing required by transmission, on the medium access technique, on the provided service classes and on the strategies used for service provisioning for different users. The scheduling algorithm implemented in the simulator allocates resources according to the priorities of the provided services and tries to ensure fairness between the users which have services with the same priority, even if they have different transmission bit rates.

The operating principle of the implemented scheduler is briefly described bellow:

- each user application is characterized, within each TTI interval, by an average bit rate, r_M , and by an instantaneous bit rate, r_i . The computed average bit rate r_M must be equal to the average bit rate, d_M , which characterizes the application, if there are sufficient radio resources available;
- the average bit rate is recomputed in each TTI interval using the instantaneous bit rate of (N-1) previous TTI intervals and of the current TTI interval;
- the desired instantaneous bit rate of the current interval is computed such that the computed average bit rate r_M to be equal with the imposed average bit rate d_M , according to (8).

$$r_N = d_M \cdot N - \sum_{i=1}^{N-1} r_{N-i} \quad (8)$$

where r_N is the instantaneous bit rate to be provided in the current interval N, to ensure the imposed average bit rate, d_M ; r_{N-i} denote the real instantaneous bit rate allocated in the previous N-i TTI intervals;

- the computed (desired) instantaneous bit rate is replaced with the closest available bit rate, which can be obtained with the available transport frames and the current coding rate (the coding rate is modified by the HARQ process – see details in paragraph 5.2. related to the operations performed by the simulation program);
- the transport frame obtained is mapped on the OFDMA chunks. If the mapping is not possible due to the lack of radio resources (available chunks), a smaller transport frame is used and the instantaneous bit rate, r_N , is recomputed (reduced), providing in this way the real instantaneous bit rate in the current TTI interval N;
- the allocation of the chunks to a certain user is performed according to the Best

Frequency Position (BFP) principle, [Ste03], [Var05]. According to this principle, the scheduler uses the SNR levels predicted by each user for each chunk of the cell carrier (the predicted values are transmitted to the base station on the uplink channels) and allocates to the user the chunk or chunks which ensure the maximum possible values of the SNR. This approach ensures an instantaneous bit rate as close as possible to the bit rate imposed by the application and priority of the user – see the next paragraph. In this way, the chunk allocation algorithm considers both the radio channel quality of the considered user in the current TTI and the specific parameters of the application (required bit rate, priority).

The operation required by the computation of the instantaneous bit rate/TTI and the OFDMA chunk allocation are performed sequentially for each user, a given user disposing at a certain moment only of the resources available after the resource allocation operations of the previous users. So, an ordering of the users is required, which is performed in the following way:

- the users are grouped on service classes and inside each class they are placed in the descending order of the ratio between the desired instantaneous bit rate (see step three of the scheduler functioning principle) and the average bit rate of the application, d_M ; note that the ratio between this bit rate and the imposed average bit rate, d_M is more important than the absolute value of the required bit rate. This procedure ensures a reasonable allocation of the resources (fairness) between users having services with the same priority, but with different bit rate requirements (for ex. an audio streaming and a video streaming);
- the resource allocation is performed in the previously mentioned order. Those users who have higher priority and require higher instantaneous bit rate relative to the application average bit rate, are allocated first;
- if it is not possible to allocate enough resources to some users, they will be allocated only the available resources, or it is possible not to be allocated at all. This will lead to a decrease of their computed average bit rate, and in the next TTI these users will “climb” in the allocation table and will get more radio resources (chunks with higher SNR or/and a greater number of chunks);
- this algorithm requires the use of some buffers with appropriate dimensions in the base station, which should undertake the bit rate fluctuations and possible burstiness of the input data streams and ensure a continuous output stream with an average value equal to the average bit rate characteristic to the considered application;

5.2.2. Generation of the statistic tables. Table formats

The types of the statistics associated to the emulated transmission, the format of tables with statistics and the generation algorithm of random variables distributed according to the imposed statistics constitute essential aspects of the emulation process. The number of statistics used and the table formats significantly influence the external memory capacity required by the emulator. The emulator uses two types of distributions, namely the distribution of data packets' (IP packets or Ethernet MAC packets transmitted through the emulator) delays, and the distribution of errors which affect these packets.

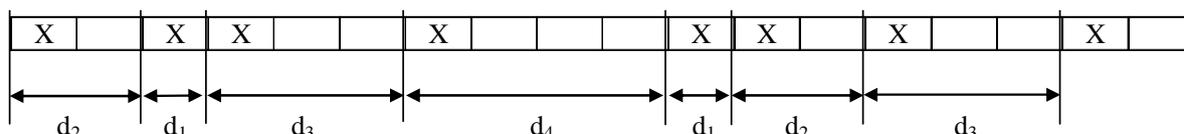
Regarding the delays of the data packets, the direct generation of these delays is not an optimal solution, because it would require an absolute or relative reference, relative to which the delays are measured. The use of the previous packet as a reference is not an acceptable solution because consecutive data packets applied to the emulator could be separated by random time intervals, which is characteristic to packet transmissions.

An alternative better solution is to generate the instantaneous bit rate corresponding to the considered transmission, i.e. the generation of the instantaneous number of bits or bytes which are transferred in an imposed time period, corresponding to one TTI. This is the solution adopted in the present version of the emulator.

As for the error distribution, for a realistic emulation of the transmission process it is not sufficient to generate the number of bit errors out of a number of transmitted bits, i.e. the bit error probability, but it is necessary to generate the errors distribution pattern. The generation of these patterns is equivalent with the generation of the packet errors, operation which requires 4 parameters: the length of the error packet, the distance between two consecutive packets, the number of errors in a packet and the distribution of errors inside the packet.

A more simple generation of the error packets can be accomplished by generating the distance between two consecutive errors, solution which allows the generation of both packet errors and independent errors. The simulated transmission chain delivers a sequence of bits affected by errors as depicted in fig. 5.4.

If the distances between all groups of consecutive errors are measured and these distances are counted, the distribution, or probability density function (pdf), of the distance between two consecutive errors could be obtained.



X – bit errors; d_x – distance (in bits) between two consecutive errors

Fig. 5.4. Sequence of bit errors and the distance between consecutive errors.

This pdf can be represented using two tables, one containing the values of distances between two consecutive errors and the other one containing the number of apparitions or the apparition probability of these distances - see fig. 5.5. The apparition probability of these distances is calculated according to:

$$p_i = \frac{\text{count}_i}{\sum_{k=1}^n \text{count}_k} \quad (9)$$

| | | | | | | | | |
|------------------------------------|------------------------------------|------------------------------------|------------------------------------|------------------------------------|------------------------------------|-------|--|------------------------------------|
| d ₁ | d ₂ | d ₃ | d ₄ | d ₅ | d ₆ | | d _{n-1} | d _n |
| count ₁ /p ₁ | count ₂ /p ₂ | count ₃ /p ₃ | count ₄ /p ₄ | count ₅ /p ₅ | count ₆ /p ₆ | | count _{n-1} /p _{n-1} | count _n /p _n |

Fig. 5.5. Representation of the distance distribution between two consecutive errors.

The number of distinct distances between two consecutive errors can be relatively large due to the variability of the radio channel inserted by the motion of the mobile and by the combination of the signals on different propagations paths, fact which could generate a temporary increase of the instantaneous SNR. To decrease the dimensions of the tables which store the distances between consecutive errors, their values are quantized using a moderate quantization step (e.g. 7) and the counters/probabilities associated to the distances falling in the same quantization interval are grouped; this is equivalent to the computation of the occurrence probability of the distance intervals and generates the probability density function of the quantized distances, see fig 5.6.

| | | | | | | | | |
|--|----------------|--|----------------|--|----------------|-------|------------------|--|
| d ₁ | d ₂ | d ₃ | d ₄ | d ₅ | d ₆ | | d _{n-1} | d _n |
| p ₁ | p ₂ | p ₃ | p ₄ | p ₅ | p ₆ | | p _{n-1} | p _n |
| ← quantization step → | | ← | | ← | | ← | | ← |
| mean(d ₁ ÷ d ₂) | | mean(d ₃ ÷ d ₄) | | mean(d ₅ ÷ d ₆) | | | | mean(d _{n-1} ÷ d _n) |
| p ₁ +p ₂ | | p ₃ +p ₄ | | p ₅ +p ₆ | | | | p _{n-1} +p _n |

Fig. 5.6. Generation and representation of the distribution of quantized distances between consecutive errors.

Finally, two tables are stored for the representation of the error distribution, namely: the quantized values of the distances between two consecutive errors and the cumulated probability density function (cdf function), computed based on the pdf function. The cdf function can be used more easily for the generation of random variables distributed according a given pdf function.

The generation of the error-sequences also requires the distribution of the number of errors on a TTI time interval. This distribution should be considered together with the number of transmitted bits in such an interval. At the output of the simulated transmission chain, besides the random variable describing the distances between consecutive errors, a bi-dimensional random variable which jointly describes the distribution of the numbers of bit errors/TTI and the distribution of the numbers of bits/TTI is also provided. These distributions can be described by the matrix presented in fig. 5.7. The indexing in the matrix is given by number of bit errors/TTI (on rows) and by the number of bits/TTI (on columns), and the positions inside the matrix give the number of occurrences of a certain combination (pairs of counters of numbers of errors/TTI and bits/TTI).

| | | | | | | |
|----------------|---|-----------|-----------|-----------|-----------|-----------|
| | | bits/TTI | | | | |
| | | i | j | k | l | m |
| no. errors/TTI | a | | count j-a | | count l-a | |
| | b | count i-b | | | | |
| | c | | | | | count m-c |
| | d | count i-d | | count k-d | | |
| | e | | count j-e | count k-e | count l-e | count m-e |
| | f | | | | | |
| | g | count i-g | | | count l-g | |
| | h | | count j-h | | | |

Fig. 5.7. Matrix structure which describes the bi-dimensional random variable no. errors/TTI and no. bits/TTI.

The matrix structure presented in figure 5.7 is not worth to be used as such to generate the two combined random variable, due to two reasons: a lot of positions of the matrix are empty (some pairs no. errors/TTI, bits/TTI have null occurrence probabilities); the generation of bi-dimensional random variables is more complicated and requires more processing time.

For an easier generation of the above mentioned random variables, the matrix of fig. 5.7. is transformed into 3 tables, after the suppression of the zero-value positions. One table contains the no. of bit errors/TTI, the second table contains the no. bits/TTI, and the third table contains the numbers of the considered pairs. For a certain position in the matrix, the same position is used in each table to store the three mentioned values, as shown in figure 5.8., which illustrates the transformation of the matrix presented in fig. 5.7.

| | | | | | | | | | | | | | |
|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|
| i | i | i | j | j | j | k | k | l | l | l | m | m | bits/TTI |
| b | d | g | a | e | h | d | e | a | e | g | c | e | bit errors /TTI |
| cnt _{i-b} | cnt _{i-d} | cnt _{i-g} | cnt _{i-a} | cnt _{i-e} | cnt _{i-h} | cnt _{k-d} | cnt _{k-e} | cnt _{l-a} | cnt _{l-e} | cnt _{l-g} | cnt _{m-c} | cnt _{m-e} | |

Fig. 5.8. Representation of the distribution of the bi-dimensional random variable no. errors/TTI - no. bits/TTI using a set of three tables.

To further decrease the numbers of positions in the three tables, a distance d_i between two consecutive positions is defined according to relation (9).

$$d_{i-i} = |bit_er_i - bit_er_{i-1}| + |bit_i - bit_{i-1}| \tag{9}$$

where i is a given position in the tables, bit_er_i represents the number of bit errors on position i , and bit_i represents the number of bits on position i .

The consecutive positions which have the minimum value of the defined distance $d_{t,j}$ are combined in the following way: the counter (or probability) of the new group is the sum of the two initial counters, and the numbers of bit errors/TTI (bit_er_{ni}) and bits/TTI (bit_{ni}) of the new group are given by the weighted averages calculated according (10).

$$bit_er_{ni} = \frac{bit_er_i \cdot contor_i + bit_er_{i-1} \cdot contor_{i-1}}{contor_i + contor_{i-1}} \quad (10)$$

$$bit_{ni} = \frac{bit_i \cdot contor_i + bit_{i-1} \cdot contor_{i-1}}{contor_i + contor_{i-1}}$$

See fig. 5.9 for an example of the processing applied to data stored in the 3 tables presented in fig. 5.8.

| | | | | | | | | | | | | | |
|---|--------------------|--------------------|--------------------|---|--------------------|--------------------|--------------------|--------------------|--------------------|---|--------------------|--------------------|-----------------|
| i | i | i | j | j | j | k | k | l | l | l | m | m | bits/TTI |
| b | d | g | a | e | h | d | e | a | e | g | c | e | bit errors /TTI |
| cnt _{i-b} | cnt _{i-d} | cnt _{i-g} | cnt _{j-a} | cnt _{j-e} | cnt _{j-h} | cnt _{k-d} | cnt _{k-e} | cnt _{l-a} | cnt _{l-e} | cnt _{l-g} | cnt _{m-c} | cnt _{m-e} | counter |
| $\longleftrightarrow d_{t,m}$ | | | | $\longleftrightarrow d_{t,m}$ | | | | | | $\longleftrightarrow d_{t,m}$ | | | |
| $\frac{(i \cdot cnt_{i-b} + i \cdot cnt_{i-d})}{(cnt_{i-b} + cnt_{i-d})}$ | | i | j | $\frac{(j \cdot cnt_{j-e} + j \cdot cnt_{j-h})}{(cnt_{j-e} + cnt_{j-h})}$ | | k | k | l | l | $\frac{(l \cdot cnt_{l-g} + m \cdot cnt_{m-c})}{(cnt_{l-g} + cnt_{m-c})}$ | | m | bits/TTI |
| $\frac{(b \cdot cnt_{i-b} + d \cdot cnt_{i-d})}{(cnt_{i-b} + cnt_{i-d})}$ | | g | a | $\frac{(e \cdot cnt_{j-e} + h \cdot cnt_{j-h})}{(cnt_{j-e} + cnt_{j-h})}$ | | d | e | a | e | $\frac{(g \cdot cnt_{l-g} + c \cdot cnt_{m-c})}{(cnt_{l-g} + cnt_{m-c})}$ | | e | bit errors /TTI |
| cnt _{i-b} +cnt _{i-d} | | cnt _{i-g} | cnt _{j-a} | cnt _{j-e} +cnt _{j-h} | | cnt _{k-d} | cnt _{k-e} | cnt _{l-a} | cnt _{l-e} | cnt _{l-g} +cnt _{m-c} | | cnt _{m-e} | counter |

Fig. 5.9. Processing of tables which describe the bi-dimensional random variable.

The operations described above are repeated until the lengths of the tables become smaller than an imposed limit; e.g. the maximum length of the tables can be set to 128 or 256. Finally, the positions in the counter table are replaced with the occurrence probabilities of the pairs of random variables stored in the two other tables, i.e. the pdf functions of these pairs, and after this operation the pdf function is replaced by the cdf function.

The simulation program generates as output data two files:

- a text file (stat_file.txt), which stores the data of the initial tables, i.e. the distances between consecutive errors, and the counters of these distances (see fig. 5.5.), and the data of the matrix with the numbers of errors/TTI, numbers of bits/TTI and the counters of these groups of data (see fig 5.7.);
- a file stream (stat_file.sta), which contains the representation of the final cdf functions associated to the distances between errors and to the pairs (no. errors/TTI; no. bits/TTI), see figures 5.6. and 5.9.

6. Compatibility with 3GPP standards. Possible developments

6.1. Compatibility with E-UTRA standards

To evaluate the compatibility of the present version 1 (drop 1) of the emulator and simulator with the 3GPP E-UTRA standards the following aspects are considered:

1. the physical layer for the downlink transmission – frame structure, transmission scheme, link adaptation;
2. data transfer at layer 2 (MAC and RRC) – frame structures, buffer management, ARQ and HARQ operations, mapping of the logical channels on the transport channels;
3. QoS concepts, service bearer architecture and scheduling/rate control operations (Dynamic Resource Allocation – DRA or Packet Scheduling - PS);

The compatibility analysis regarding the uplink connection (physical layer and layer 2), management and transmission on signaling channels and mobility management can be performed only after the implementation or at least the design of the blocks in discussion in the emulator and in the simulator. Paragraph 6.2. presents several considerations regarding the insertion of the blocks mentioned above in the implemented emulator-simulator and the processing required.

1. *Compatibility at the physical layer - downlink.*

The [3GPP07] standard proposes for the downlink physical layer (chapter 5 [3GPP07]) the following features:

- a 10ms radio frame composed of 20 slots with 0.5ms duration (an alternative variant for the TDD duplex mode contains 12 slots). 12 consecutive OFDM sub-carriers during one slot period compose a resource block. This resource block corresponds practically to a chunk (or bin) and has duration of one TTI interval. Several resource blocks can be allocated to a service (user). The employment of pilot sub-carriers inside the resource block is not specified (at least until the elaboration of the present material). The OFDM sub-carrier separation can be 15 kHz or 7.5 kHz and the used cyclic prefix varies from one resource block to other (see [3GPP07 chapter 5]). Some of the first and last OFDM symbols of the radio frame are used to transmit some information regarding the resource allocation, ARQ/HARQ processes, ACK/NACK signaling, uplink scheduling and reference signals for cell identification;
- turbo codes are proposed as coding techniques, one code being used for the data of all resource blocks associated to a layer 2-connection. The QPSK, 16QAM or 64QAM modulations are considered, the same modulation being used in all resource blocks associated to a layer 2-connection. The modulation technique and the parameters of the coding techniques can be modified adaptively from one slot to another. Static or semi-static (combination of static and adaptive) use of modulation and coding techniques is also allowed;
- the standard also includes the possibility to use multi-antenna techniques with 2 or 4 transmit antennas and the use of power control;

The simulator associated to the implemented emulator has the following features:

- it allows the use of chunks with variable dimensions, the modification of OFDM modulation parameters, but the cyclic prefix is the same for all the OFDM symbols. It does not offer the possibility to use some groups of OFDM symbols for the transmission of signaling information;
- it uses a single LDPC code on a group of chunks associated to a layer 2-connection and the allowed modulation techniques are 2PSK, QPSK, 16QAM and 64 QAM and optionally 256QAM. The modulation techniques can vary adaptively from one chunk to other, thus providing a higher flexibility in modifying the instantaneous bit rate;

- it does not include the possibility of using multi-antenna techniques;

Concluding, the main differences between the implemented simulator and the [3GPP07] specifications (at the physical layer) are the use of LDPC codes, instead of turbo codes and the lack of multi-antenna techniques and power control in the implemented simulator. There should be noted that the turbo codes and the LDPC codes have similar performances, especially when used adaptively, and that the decoding of LDPC codes requires smaller computation resources than the decoding of turbo codes.

2. *Compatibility at layer 2 - downlink*

The Layer 2 of the [3GPP07] is composed of the following sub-layers (chapter 6 [3GPP07]): PDCP (Packet data Convergence Protocol), sub-layer which implements this protocol, RLC (Radio Link Control), and MAC (Medium Access Control).

- PDCP deals with header compression and security;
- RLC performs a segmentation of the data using a specific frame structure, implements an ARQ process for error control and loads the data streams received from the PDCP sub-layer in the defined logical channels;
- the MAC sub-layer performs the scheduling of different data streams received from the RLC sub-layer, multiplexes these streams and loads them in the transport channels (frames) composed of resource blocks. The MAC sub-layer implements also a HARQ process for error control;

In its present version, the LTE simulator/emulator implements only the MAC sub-layer without data multiplexing, because one user transmits a single data stream. The MAC sub-layer receives directly the data stream and loads it in the transport channels (frames).

The error control is made only by the HARQ process included in this sub-layer, because it is implemented only the MAC sub-layer;

- ❖ Note: the transport channels are composed of transport frames.

3. *Compatibility at bearer service level, QoS and scheduling/rate control*

The [3GPP07] specifications define a stratified SAE Bearer Service architecture (chapter 13 [3GPP07]) to ensure end-to-end QoS for the transmitted data streams. The data frames are transmitted according a given QoS profile associated to each SAE Bearer.

The implemented simulator allows the specification of different data streams (data streams associate to the main user and to the secondary users), and each such data stream can be considered as a separate bearer characterized by a given priority and average bit rate (see paragraph 4.2. related to the definition of the user parameters). Some conditions related to the error probability can be additionally imposed.

The scheduling/rate control specified by [3GPP07] (chapter 11 [3GPP07]) targets the allocation for each TTI, or groups of TTI, of the resource blocks and modulation and coding techniques. It defines four resource allocations strategies for different data streams (bearers), strategies which include fix, dynamic (adaptive) and combined (fix + dynamic) allocations. The scheduler allocates resources based on the traffic amount, QoS requirement associated to each data stream and on the radio propagation conditions associated to each mobile user. The allocation strategy of the bit rate in the downlink is based on increase of the MBR (“Maximum Bit Rate”) for an imposed GBR (“Guaranteed Bit Rate”), or on the maximization of the AMBR (“Aggregate Maximum Bit Rate”) for a group of Non-GBR data streams.

In the current version of the simulator, the basic principles of the scheduling process are identical with those specified by [3GPP07]. The resource allocation can be done only dynamically for each TTI interval. As for the bit rate allocation strategy, it is developed to ensure a given average bit rate and not to increase of the maximum bit rate associated to a guaranteed

bit rate, because it emulates only one data stream/user and therefore does not allow the evaluation of AMBR.

The brief comparison presented above shows that there is a good compatibility between this version of LTE downlink simulation/emulation and the [3GPP07] standard specifications.

6.2. Possible developments of the present version

In the present version of the LTE emulator, the physical layer emulation/simulation is performed only for the downlink direction, the uplink direction being considered ideal.

A first possible development is **the emulation/simulation of the transmission in the uplink direction**. The changes which have to be operated at the emulator level are the following:

- Reading from the files with statistics of the delay and bit error distributions for the uplink direction and the generation of tables required for the generation of the appropriate random variables;
- Delay of packets transmitted in uplink and error insertion in these packets;
- ❖ Note: the operations which have to be executed for the emulation of the uplink direction are identical with those executed for the downlink direction, only the distributions used to generate delays and error packets would be different.

More important modifications have to be made in the simulation program, because the parameters of the uplink physical layer are different to those that define the downlink physical layer. The physical layer of the uplink connection uses other types of transmission and access schemes, of modulation and FEC coding and other frame structures.

Another possible development is **the simulation/emulation of the signaling traffic**. This implies modifications both at the level of emulator and simulator.

At the emulator level it requires the separation between data and signaling packets in both transmission directions. These packets must be separately processed, fact which requires different appropriate statistics. The emulation of a separate bidirectional channel implies also more computations and therefore, more powerful computers and a better optimization of the implementation would be needed.

At the simulator level it would involve the implementation of the processing required by the transmission of data over the signaling channels and the generation of the file(s) with the appropriate statistics.

Another important development is **the emulation of a multicell configuration**, in which the main user can pass from one cell to other. This emulation would require modifications only in the emulator. When the main user reaches the edge of the cell or the SNR decreases below a given threshold, a handover procedure should be called to decide if the emulated users should connect to the base station of the new cell (considering the load of that cell or some statistics which describe the handover procedure). If the handover would be possible, this procedure should choose a new set of delay/error statistics corresponding to the new radio propagation conditions and cell load. The parameters which describe the new cell in the emulator (see paragraph 3.1.1.) should also be modified. The block that manages the handover operation should also generate/emulate the signaling traffic required by this operation.

Other possible developments at the simulator level are the following:

- Simulation of different multi-antenna techniques (MIMO techniques) on both transmission directions;
- Generation of the test data (test bit streams) used in the simulated transmission according to some appropriate distributions related to the application (service) associated to the main user, whose transmission is emulated. The present implementation considers that both ends of the radio transmission system are equipped with buffers with appropriate dimensions, which undertake all the variations of the input data stream. For a more realistic simulation/emulation, it would be necessary to consider in the simulation

process a finite length input buffer which stores the data according to the distribution of the input bit rate and from which data are read according to the distribution of the simulated channel bit rate. An important problem which should be solved in this case is the correlation between the loading of the buffer in simulator and emulator;

- The use of other scheduling algorithms (radio resource allocation) or even the possibility to select the scheduling algorithm;

Another necessary development of the emulator/simulator is the design of some kind of data base, which should store the names and paths of files with the statistics associated to different scenarios and of an appropriate interface of the emulator which should allow the easy retrieval of the name and path of the required file from this data base. The actual version of the emulator requires typing the name and path of the file with the used statistics in an editing-box of the interface, operation which is tedious and prone to typos.

7. Installation of the emulation program. Computational resources required

The emulation program does not require installation; it must be only copied on the hard drive of the computer used as emulator. All the library files are included (statically linked to) in the emulation program, which can be run without requiring any other operations.

The program requires one of the Windows NT4 / 2000 / 2003 / XP operation systems. The emulator also requires the previous installation of the WinPcap 4.0 library, the packet capture being based on the functions of this library. The installation of this library should be checked before running the emulator. For this open in the “Start” menu the “All Programs” sub-menu and check if there is a “WinPcap” entry and if it is the correct version, if a WinPcap library is installed. A quick check of WinPcap library installation can be made at the moment when the emulator program is launched. The emulator automatically detects the network cards of the computer (on which the program is run) and displays the name of these network cards in the “Client NIC” and “Server NIC” edit boxes of the user interface. If no WinPcap library is installed the network cards can not be detected and nothing will be displayed in the mentioned edit boxes.

- ❖ Note: if newer versions of WinPcap library are installed it is necessary to check the compatibility between the functions of the new version (functions used in the emulator) and the functions of WinPcap version 4.0. For this, the user should run the emulation using a known test sequence, see chapter 8. related to the adjustment of the bit rate transferred through the emulator.

As for the computational resources, the emulation program, which has to work in real time, must be run on a computer with a minimum processor clock frequency of 1.8 - 2GHz. A minimum 64MB RAM should suffice to run the program, but to ensure an appropriate processing speed a high bus clock frequency and a greater RAM are required.

It is also recommended to not run other applications in parallel with the emulator and to set its priority to „High” in the Task Manager; a „Realtime” level of priority would be a better choice, but in this case it is possible to block the computer, so this option has to be used carefully.

The deactivation of the network protocols on the computer which runs the emulator is strongly recommended to avoid the generation of unwanted packets by different components of the operating system running in the background. The generation of such packets will increase the number of operations performed by the packet capture block increasing in this way the execution time.

The tests performed on an Acer laptop with Mobile AMD Sempron 3300+ processor (processor clock frequency 2GHz) allowed the emulation of a maximum 2.5Mbps bit rate with 1 error event per second.

8. Adjustment of the bit rate transferred through the emulator. Remarks

Delays inserted by the operations performed in the emulation process (especially by the operations related to the graphical display of the user’s position inside the cell and of the emulation statistics), and the gaps between consecutive packets affect (decrease) the value of the instantaneous bit rate transferred through the emulator.

The emulation program offers facilities for fine adjustment of the transfer bit rate – see paragraph 3.1.2. dealing with the command and control block. To perform this adjustment, the user should select a high signal to noise ratio (see paragraph 3.1.1. related to the signal to noise ratio computation block), e.g. 40dB, and apply a known and constant bit rate to the emulator.

The bit rate is adjusted, using the “Scroll Bar” (“Bit rate adjustment”) of the user interface provided for this operation, see paragraph 3.1.2, until the average value displayed in the user interface (“Average Bit Rate” edit box) is identical (or approximately identical) to the bit rate applied at the input. This operation requires a constant bit rate generator.

The implemented bit rate generator supplies MAC Ethernet packets with variable length in an imposed cadence to provide the desired constant bit rate. The timing of the packets is performed with the computer’s high precision timer „*QueryPerformanceCounter(*LARGE_INTEGER)*” (see paragraph 3.2. related to the main operations performed in the emulation process), and the transmission of the generated MAC packets is achieved with the functions of the WinPcap library. The interface of the program is presented in fig. 8.1.

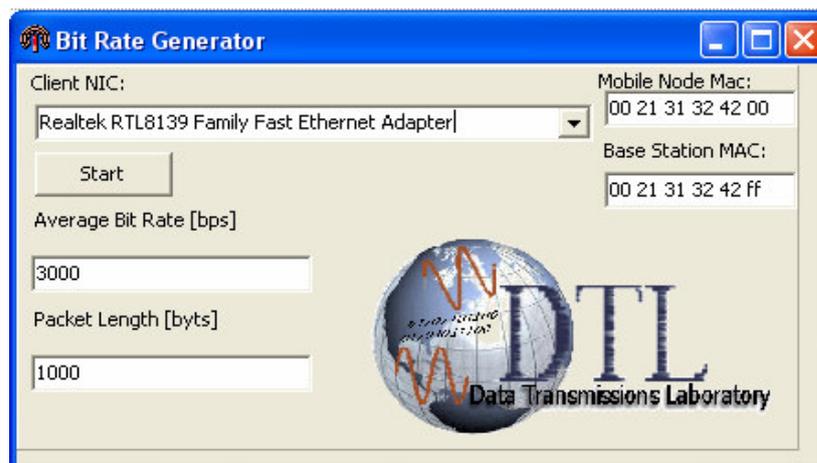


Fig. 8.1. User interface of the constant bit rate generator.

This interface allows the selection of the network card used for the connection with the emulator, the specification of the MAC address of this card (source MAC address) and of the MAC address of the client connected to the output of the emulator (should such a client be connected to the emulator), the value of the generated bit rate and the length (in bytes) of the generated MAC Ethernet packets. The “Start / Stop” button is used to start, and respectively, to stop the generated data stream.

Three important remarks regarding the operation of the emulator should be noted:

- If the error probability corresponding to the emulated scenario is too high, then the downlink thread, which implements the processing for the downlink transmission, could be blocked; in this case an average transferred bit rate equaling zero is displayed on the corresponding graph. Should such a situation occur, the emulation must be stopped and restarted using the “Start/Stop” button of the emulator interface;
- At high bit error probabilities, negative values might appear in the circular buffer used for the computation of the average bit rate; this fact determines the display of a negative average bit rate both in the appropriate edit box and in the diagram associated to the

evolution in time of the average bit rate. In this situation the “Cl. Av.” („Clear Average”) button must be pushed; it performs the re-initialization of the mentioned circular buffer and of the buffers used by the functions which display the time evolution of the average bit rate;

- For high bit-error probabilities, due to some very high values that could appear in the circular buffer used for average bit rate computation mentioned above, abnormal (very high) values for the average bit rate might be displayed. If such a case occurs, the user should also push the “Cl. Av.” button, as mentioned above.

9. Appendix: Multipath propagation models for different frequency bands

2.4GHz –WiFi channel model [Steg03]

| RMS Delay Spread | Path 1 Gain (0 dB) | Path 2 Gain (-4 dB) | Path 3 Gain (-8 dB) | Path 4 Gain (-12 dB) | Path 5 Gain (-16 dB) | Path 6 Gain (-20 dB) |
|------------------|--------------------|---------------------|---------------------|----------------------|----------------------|----------------------|
| 0 ns | 0.000 ns | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| 10 ns | 0.000 ns | 10.167 | 20.333 | 30.500 | 40.667 | 50.833 |
| 20 ns | 0.000 ns | 20.500 | 41.000 | 61.500 | 82.000 | 102.500 |
| 30 ns | 0.000 ns | 30.667 | 61.333 | 92.000 | 122.667 | 153.333 |
| 40 ns | 0.000 ns | 41.000 | 82.000 | 123.000 | 164.000 | 205.000 |
| 50 ns | 0.000 ns | 51.167 | 102.333 | 153.500 | 204.667 | 255.833 |

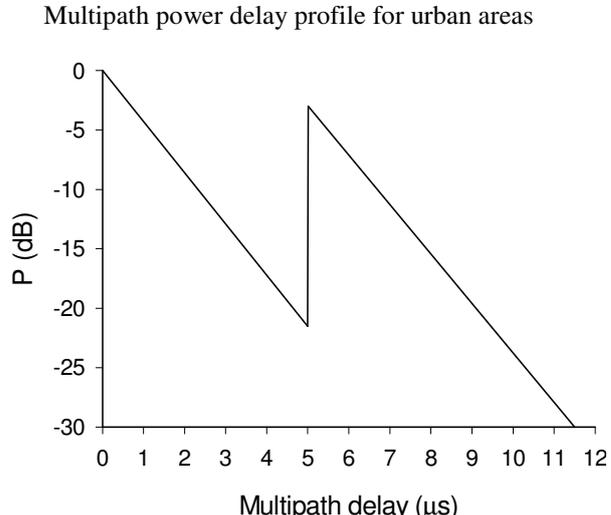
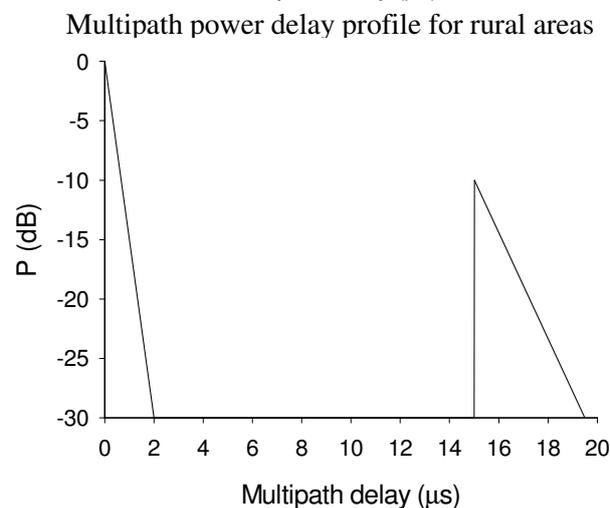
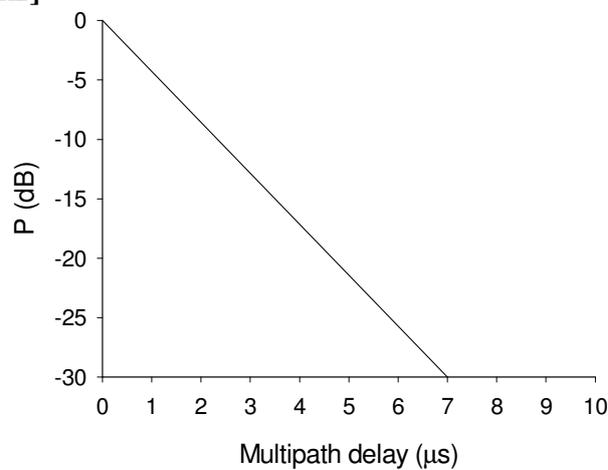
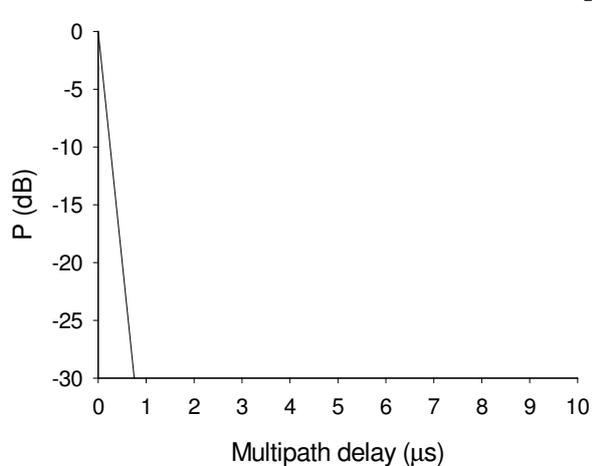
1.9GHz – simple channel model [Ste03]

| | | | | |
|------------------|---|-----|-----|-----|
| Path no. | 0 | 1 | 2 | 3 |
| Attenuation (dB) | 0 | 3 | 6 | 9 |
| Delay (ns) | 0 | 200 | 400 | 600 |

5GHz – simple channel model [Win05_4]

| | Urban Macro | | | Urban Micro | | |
|----------------------------|-------------|--------|----|-------------|--------|----|
| Relative Path Power / | 10 / 20 | -3.010 | 0 | 7 / 20 | -4.559 | 0 |
| Relative Path Power (dB) / | 6 / 20 | -5.229 | 7 | 5 / 20 | -6.021 | 5 |
| Delay (ns) | 4 / 20 | -6.990 | 27 | 4 / 20 | -6.990 | 11 |
| | | | | 4 / 20 | -6.990 | 28 |
| Delay-Spread (ns) | 10.2 | | | 10.0 | | |

800MHz – 1.8GHz – GSM channel model [IEE]



Multipath power delay profile for typical hilly terrain

Multipath power delay profile for bad case hilly terrain

TUCN – Data Transmission Laboratory

5GHz – WP5 channel model [Win05_3]

| Suburban Macro | | Urban Macro | | Urban Micro | |
|----------------|------------------|-------------|------------------|-------------|------------------|
| Power [dB] | Delay [μ s] | Power [dB] | Delay [μ s] | Power [dB] | Delay [μ s] |
| -3.0000 | 0.000 | -3.0000 | 0.000 | -4.5500 | 0.000 |
| -5.2200 | 0.010 | -5.2200 | 0.010 | -6.0000 | 0.010 |
| -6.9800 | 0.025 | -6.9800 | 0.030 | -6.9800 | 0.015 |
| -5.6682 | 0.140 | -5.2204 | 0.360 | -6.9800 | 0.030 |
| -7.8882 | 0.150 | -7.4404 | 0.370 | -5.8161 | 0.285 |
| -9.6482 | 0.165 | -9.2004 | 0.385 | -7.2661 | 0.290 |
| -9.2147 | 0.060 | -4.7184 | 0.250 | -8.2461 | 0.295 |
| -11.4347 | 0.070 | -6.9384 | 0.260 | -8.2461 | 0.310 |
| -13.1947 | 0.090 | -8.6984 | 0.280 | -7.2701 | 0.205 |
| -13.4132 | 0.400 | -8.1896 | 1.040 | -8.7201 | 0.200 |
| -15.6332 | 0.410 | -10.4096 | 1.045 | -9.7001 | 0.220 |
| -17.3932 | 0.430 | -12.1696 | 1.065 | -9.7001 | 0.230 |
| -19.4735 | 1.380 | -12.0516 | 2.730 | -8.8473 | 0.665 |
| -21.6935 | 1.390 | -14.2716 | 2.740 | -10.2973 | 0.670 |
| -23.4535 | 1.410 | -16.0316 | 2.760 | -11.2773 | 0.675 |
| -25.1898 | 2.830 | -15.5013 | 4.600 | -11.2773 | 0.685 |
| -27.4098 | 2.835 | -17.7213 | 4.610 | -10.5640 | 0.805 |
| -29.1698 | 2.855 | -19.4813 | 4.625 | -12.0140 | 0.810 |
| | | | | -12.9940 | 0.820 |
| | | | | -12.9940 | 0.835 |
| | | | | -12.9806 | 0.925 |
| | | | | -14.4306 | 0.935 |
| | | | | -15.4106 | 0.940 |
| | | | | -15.4106 | 0.960 |

10. References

- [And05] Hakan Andersson, "3G RAN Long Term Evolution", *Open Workshop on NGN*, Brussels, June 22, 2005.
- [Bar06] Ulrich Barth, "3GPP Long-Term Evolution / System Architecture Evolution Overview", Alcatel presentation, September 2006.
- [Bot00] Vasile Bota, Zsolt Polgar, "Telephone Vocal-Channel Simulator Implemented on a TMS320C31 DSK Board", *Proceedings of the Third European DSP Education and Research Conference*, September 20-22, 2000, ESIEE, Noisy Le Grand, Paris, France.
- [Bot02] Vasile Bota, Zsolt Polgar, "On the Approximate Digital Generation of Uniform and Gaussian Distributed Noise Signals", *Acta Tehnica Napocensis Electronics and Telecommunications*, ISSN 1221-6542, Volume 43, Number 2, 2002, pp.29-34.
- [Bot04] V.Bota, M. Varga, Zs. Polgar; "Performances of the LDPC-Coded Adaptive Modulation Schemes in Multi-Carrier Transmissions" *Proc. of COST 289 Seminar*, July 7-9, 2004, Budapest, Hungary.
- [Chi06] Francois Chin, "Key Features and Technologies in 3G Evolution", *Proc. of EUSEA 2006*, 20 June 2006.
- [Ele02] E. Eleftheriou, S. Olcer, "G.gen:G.dmt.bis:G.lite.bis: Efficient Encoding of LDPC Codes for ADSL", ITU-T, Temporary Document SC-064, 2002.
- [Fau06] Denis Fauconnier, "HSDPA and HSUPA: UMTS evolution toward higher-bit-rate data", Nortel Technical Journal, Issue 2, 2006.
- [IEE] , "WRAN Multipath Channel Considerations", IEEE 802.22-05/0048r0, May 2005.
- [ITU] ITU-T, "LDPC codes for G.dmt.bis and G.lite.bis," Temporary Document CF-060.
- [Gal63] R. G. Gallager, *Low-Density Parity Check Codes*, MIT Press, Cambridge, MA, 1963.
- [Hye06] Lee HyeonWoo, "3GPP LTE & 3GPP2 LTE Standardization, *Proc. of KRnet 2006*, June 27-28, 2006
- [Kat06] Osamu Kato, "Activities in Panasonic towards Next Generation Mobile Network Society", Panasonic Mobile Communications presentation, 29 Mar. 2006.
- [Lar01] R. Laroia, "Flash-OFDM™ – Mobile Wireless Internet Technology", *Proc. of IMA Workshop on Wireless Networks*, August 8-10, 2001.
- [Mck99] D.J.C. McKay, "Good error-correcting codes based on very sparse matrices," *IEEE Trans. on Information Theory*, vol. 45, March, 1999.
- [Pel05] Mari-Jaana Pelkonen, "Estimating end-to-end performance in 3G Long-Term Evolution compared to HSDPA", Thesis work seminar presentation, 2005.
- [Que] , QueryPerformanceFrequency Function & QueryPerformanceCounter Function, <http://msdn2.microsoft.com/en-us/library/ms644904.aspx>, <http://msdn2.microsoft.com/en-us/library/ms644905.aspx>
- [Rap01] Th. S. Rappaport, *Wireless Communications*, New Jersey:Prentice Hall PTR, 2001.
- [See06], Alexander Seeger, "High Speed Downlink Packet Access (HSDPA)", Siemens Mobile business presentation, 2006.
- [Sie06] Siemens, "3G Network Evolution towards an IP Centric Broadband Next Generation Mobile Network (NGMN)", *Technical Whitepaper*, May 2006.
- [Spi] , "Spirals", <http://www.mathematische-basteleien.de/spiral.htm>
- [Skl97] B. Sklar, "Rayleigh fading channels in mobile digital communication systems. Part I: Characterization," *IEEE Comm. Mag.*, July 1997.
- [Ste03] M. Sternad, T. Ottosson, A. Ahlen, A. Svensson, "Attaining both Coverage and High Spectral Efficiency with Adaptive OFDM Downlinks", *Proc. of VTC 2003*, Oct. 2003, Orlando, Florida.
- [Steg03] Christopher Steger, Predrag Radosavljevic, J. Patrick Frantz, "Performance of IEEE 802.11b Wireless LAN in an Emulated Mobile Channel", *Proc. of VTC03*, 22-25 April 2003
- [Thr] , "Thread (computer science)", http://en.wikipedia.org/wiki/Thread_%28computer_science%29
- [Val06] Cyril Valadon, "HSDPA: Pushing the Limits of 3G for Enhanced Data Rates", TTPCOM presentation, 2006
- [Var05], M.Varga, V.Bota, Zs.Polgar, „User-Bin Allocation Methods for Adaptive-OFDM Downlinks of Mobile Transmissions", Proceeding of COST 289 Workshop, July, 2005, Antalya, Turkey
- [Yag04] H. Yaghoobi, "Scalable OFDMA Physical Layer in IEEE 802.16 Wireless MAN", *Intel Technology Journal*, Vol. 8, Issue 3, 2004.
- [Win05_1] IST-2003-507581 WINNER, "Final report on identified RI key technologies, system concept, and their assessment", *Report D2.10 v1.0*.
- [Win05_2] IST-2003-507581 WINNER, "Final report on identified RI key technologies, system concept, and their assessment", *Reports D1.1-D1.3 v1.0*.
- [Win05_3] IST-2003-507581 WINNER, "Final report on identified RI key technologies, system concept, and their assessment", *Report D2.3 v1.0*.
- [Win05_4] IST-2003-507581 WINNER, "Final report on identified RI key technologies, system concept, and their assessment", *Report D3.2 v1.0*.
- [Wpc] , "WinPcap: The Windows Packet Capture Library", <http://www.winpcap.org/>
- [3GPP07] 3rd generation Partnership Project, "Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Overall description, Stage2, Release 8", 3GPP TS 36.300 v8.0.0 (2007-03)