PUBLIC

IST R&D. FP6-Priority 2.
SPECIFIC TARGETED RESEARCH PROJECT
Project Deliverable

SUIT Doc Number	SUIT_208
Project Number	IST-4-028042
Project Acronym+Title	SUIT- Scalable, Ultra-fast and Interoperable Interactive Television
Deliverable Nature	Report
Deliverable Number	D2.3
Contractual Delivery Date	30 Nov 2006
Actual Delivery Date	23 Dec 2006
Title of Deliverable	WIMAX Modelling
Contributing Workpackage	WP2
Project Starting Date; Duration	01/02/2006; 27 months
Dissemination Level	PU
Author(s)	C. H. Liew (UniS), S.Worrall (UniS), M. Goldshtein (Runcom), A. Navarro (IT), M. A. Mota (IT)

Abstract

This report features a discussion of the WIMAX system, which is a part of the SUIT proposed network architecture. The first section discusses the function of basic components in the WIMAX model. Secondly, the simulation results facilitate a study of the performance of WIMAX for different mobile speeds, channel models and channel coding schemes. Furthermore, the report outlines the developed WIMAX software for trace generation, the error pattern collection method and the format of the error pattern. The error pattern will be used in other WPs to study the video performance when transmitted over WIMAX or to develop a rate control strategy.

Keyword list: WIMAX, error pattern, physical link level performance, trace generation

PUBLIC

WIMAX Modelling

SUIT_208

23 Dec 2006

Table of Contents

1 Introduction

This deliverable’s objectives are to present the performance of the WIMAX platform and to provide documentation for the generated error pattern files. These error pattern files can be used to study cross layer optimization in WP3, which allows the selection of optimal video coding parameters for transmission. This is also important for the design of intelligent playout for dynamic video source rate adaptation according to the channel condition. The usage scenario of WIMAX can be seen in Figure 1 [1] where the system is either receiving SVC or MDC streams from the playout. The robustness of the video adaptation schemes for delivery to a WIMAX network can be tested using error pattern files before the final system deployment.

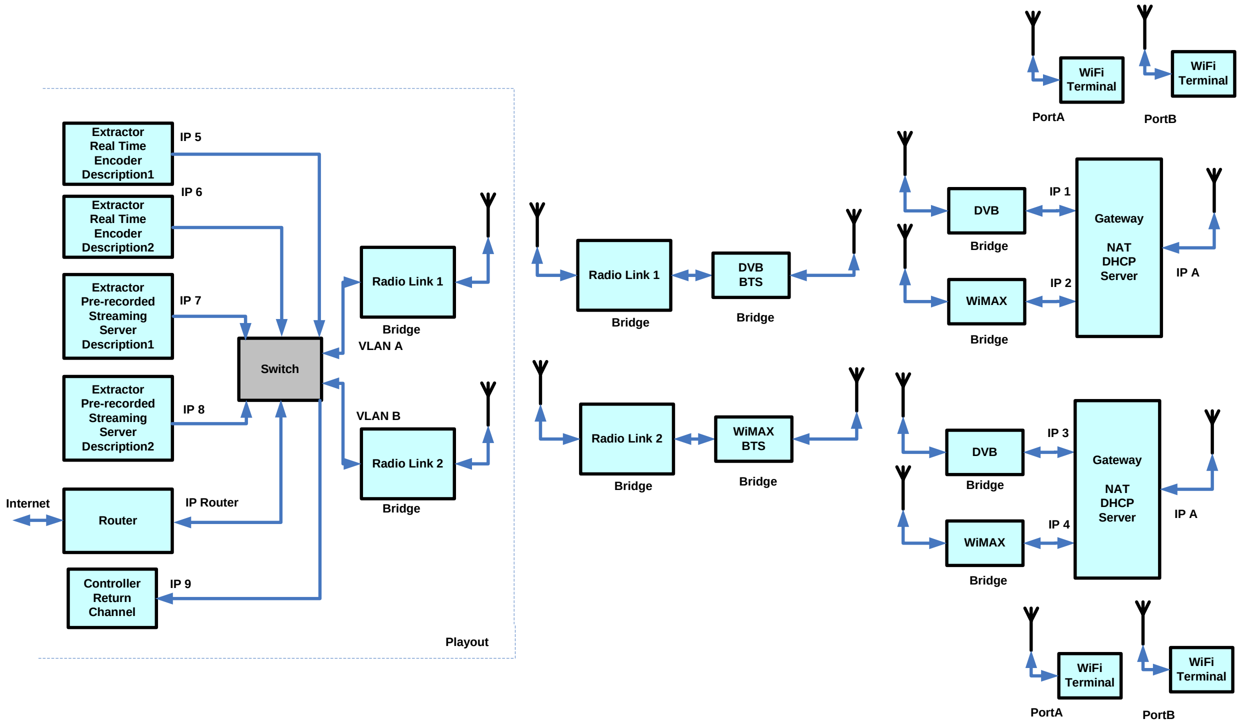

Figure 1 SUIT system for home network scenario

In Section 2, the WIMAX physical layer system is outlined. Section 3 presents the results of WIMAX physical layer performance with different setting and different channel models. Finally, Section 4 discusses the error pattern generation format and parameters. Basic assumptions for the error pattern generation are also presented. Furthermore, the functions of each module in the simulator software are given in the same section.

2 WIMAX System Description

The IEEE 802.16e-2005 standard [2] provides specification of an air interface for fixed, nomadic and mobile broadband wireless access systems with superior throughput performance. It enables non-line of sight reception, and can also cope with high mobility of the receiving station. The IEEE802.16e extension enables nomadic capabilities for laptops and other mobile devices, allowing users to benefit from metro area portability of an xDSL-like service.

The standard allows the physical layer to be scalable in bandwidth ranging from 1.25 MHz to 20 MHz with fixed subcarrier spacing while at the same time providing advanced features such as Adaptive Modulation and Coding (AMC), Advance Antenna Systems (AAS), coverage enhancing safety channel, Convolutional Turbo Coding (CTC) and Low Density Parity Check code (LDPC). This rich set of IEEE802.16e features allows the equipment manufacturer to pick and mix the features to provide superior throughput performance while at the same time allowing the system to be tailored to cater for restrictions in certain countries. The WIMAX forum is currently standardizing system profiles, which encompass subsets of IEEE 802.16e features considering country restriction, and at the same time allowing interoperability between equipment from different companies.

This section describes the implemented WIMAX baseband simulation model in SUIT project. The implemented features match the Runcom platform (whenever possible) that will be used for the trials, while considering the deployment scenario: broadcasting. The section first outlines the system specifications and gives an overview of every component in the baseband system model.

2.1 System Specifications and Baseband System Model

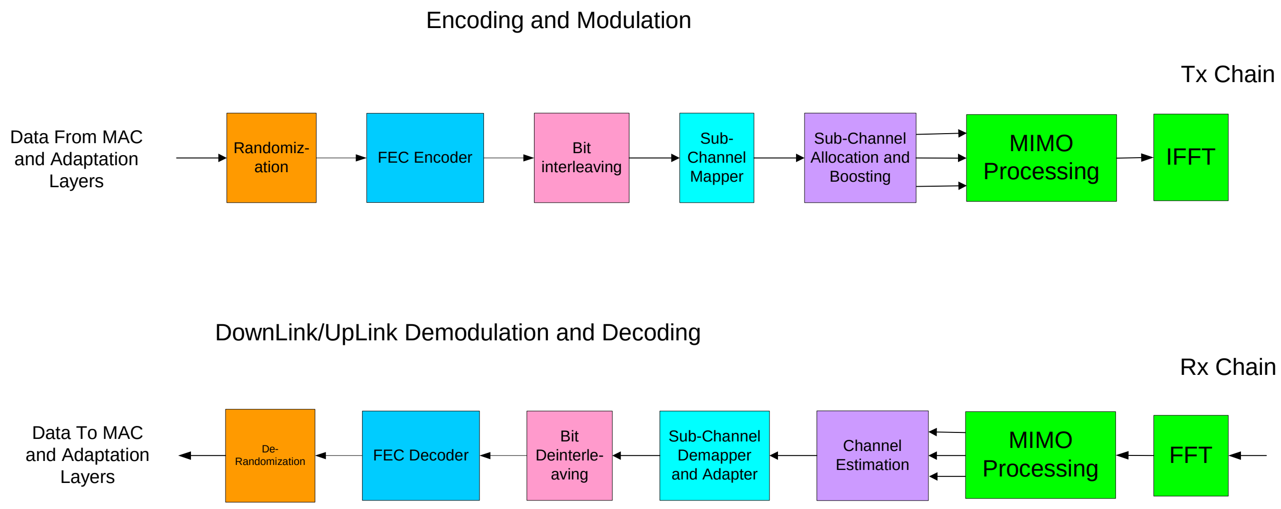

Figure 2 Physical layer of IEEE802.16e standard

Figure 2 shows the block diagram of the physical layer of the IEEE.802.16e standard. The block diagram specifies the processing of data streams. The components of the system are below:

- Randomizer.
- FEC encoder:Convolutional Turbo Code (CTC), Repetition Code etc.

- Bit interleaver.
- Data and pilot modulation.
- Subchannel allocation: FUSC, PUSC etc
- MIMO processing: Space Time Coding (STC), Spatial Multiplexing (SM) etc
- FFT/IFFT: 2048, 1024, 512, 256 points.

Channel coding procedures include randomization, FEC encoding, bit interleaving and repetition coding. When repetition codes are used, allocation for the transmission shall always include an even number of adjacent subchannels. The basic block shall pass the regular coding chain where the first sub channel shall set the randomization seed. The data shall follow the coding chain up to the QAM mapping. The data outputted from the QAM mapper shall be loaded onto the block of pre-allocated subchannels for transmission. The subchannel allocation follows one of the subcarrier permutation schemes, e.g. FUSC or PUSC. After that, multiple antenna signal processing is applied if available in the system and finally the data is passed to the OFDM transceiver (IFFT block) for transmission.

For the SUIT project purposes, a subset of the features of IEEE 802.16e is implemented considering the broadcasting scenario. The functionality and requirements from each block shall be specified in the following sections. The C++ software implementation of the baseband model uses the IT++ communication signal processing library [3].

2.1.1 Data Randomizer

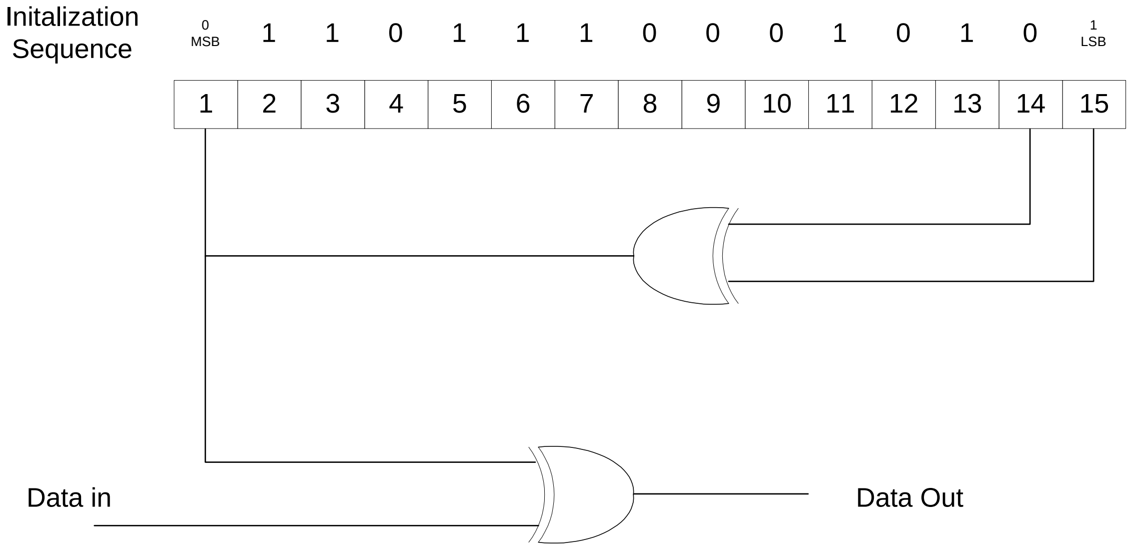

Figure 3 Data randomizer

Data randomization is performed on data transmitted on the downlink and uplink. The randomization is initialized on each FEC block (using the first subchannel offset and the OFDMA symbol offset on which the FEC block is mapped. Symbol offset, for both UL and DL, shall be counted from the start of the frame, where the DL preamble shall be count 0. If the amount of data to transmit does not exactly fit the amount of data allocated, padding of 0xFF ('1' only) shall be added to the end of the transmission block, up to the amount of data allocated. The PRBS generator is shown in Figure 7. Each data byte to be transmitted shall enter sequentially into the randomizer, MSB first. Preambles are not randomized. The seed value shall be used to calculate the randomization bits, which are combined in an XOR operation with the serialized bit stream of each FEC block. The randomizer sequence is applied only to information bits.

2.1.2 Convolutional Turbo Coding (CTC)

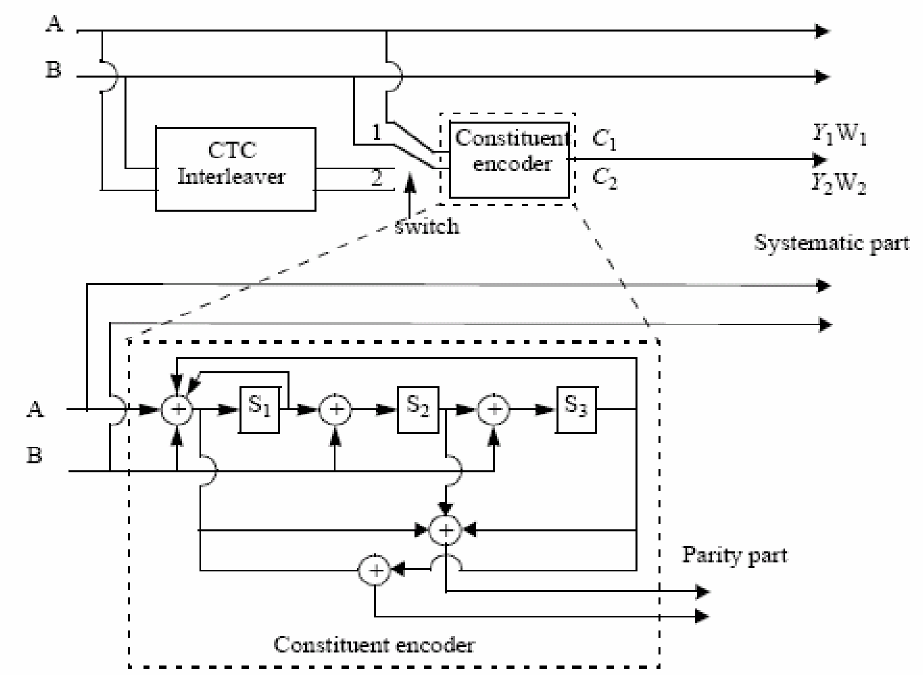

Figure 4 Convolutional Turbo Coding (CTC)

Figure 4 shows the Convolutional Turbo Coding (CTC) encoder. Incoming bits are fed alternately into A and B. During the first encoding operation, A and B are connected to position 1 of the constituent encoder to generate parity C_1. In the second step, the interleaved A and B bits are connected to position 2 of constituent encoder to generate parity C_2. A, B, C_1 and C_2 together are the mother codeword that can be punctured to obtain different code rates for transmission. The constituent encoder uses duo-binary Circular Recursive Systematic Convolutional (CRSC) code where the encoding operation is performed on a pair of bits, hence the notion duo-binary. The code is circular in such a way that the ending state matches the starting state. The polynomials defining constituent encoder connections are described in hexadecimal and binary symbol notations as follows:

- For the feedback branch: 0xB, equivalently $1+D+D^3$
- For the Y parity bit: 0xD, equivalently $1+D^2+D^3$
- For the W parity bit: 0x9, equivalently $1+D^3$

Table 1 Encoding slots concatenation for different rates in CTC

Modulation and rate	No. bits per data slot	Max. no. of concatenated slots
QPSK 1/2	48	10
QPSK 3/4	72	6
16QAM 1/2	96	5
16QAM 3/4	144	3
64QAM 1/2	144	3
64QAM 2/3	192	2
64QAM 3/4	216	2
64QAM 5/6	240	2

The number of input bits depends on the number of slots allocated by the MAC layer to the user for transmission. The slot allocation mechanism and the definition of slots are discussed in Section 2.1.5. Table 1 shows the number of bits per data slot. Concatenation of a number of data slots shall be performed in order to make larger blocks for coding whenever possible, with the limitation of not exceeding the largest block defined in Table 1 for a given applied modulation and coding rate. A larger coding block improves the coding performance, but introduces higher decoding and computational complexity. For decoding the above duobinary CTC codes, MAX-LOG-MAP [4] have been adopted.

2.1.3 Interleaver

The interleaving operation ensures that adjacent coded bits are mapped onto non-adjacent OFDM subcarriers, and at the same time mapping coded bits alternately onto less and more significant bits of the modulation constellation. The intention is to increase the robustness and avoid long runs of low reliability bits. The operation is divided into two permutation steps. The first permutation equation is:

$$i = (N_{CBPS}/16)(k \bmod 16) + \lfloor k/16 \rfloor \qquad k = 0,\ldots,N_{CBPS}-1 \tag{1}$$

where i is the index after the first permutation of index k. k is the index of coded bits before first permutation. N_{CBPS} is the encoded block size. The second permutation mapping index i to j is:

$$j = s \times \lfloor i/s \rfloor + (i + N_{CBPS} - \lfloor 16 \times i / N_{CBPS} \rfloor) \bmod s \qquad i = 0,\ldots,N_{CBPS}-1 \tag{2}$$

where $s = \max(N_{BPSC}/2,1)$ with N_{BPSC} being the number of bits per modulation symbol e.g. 2 bits for QPSK. Note that the interleaver shall not be used for CTC.

2.1.4 Modulation

2.1.4.1 Permutation Definition

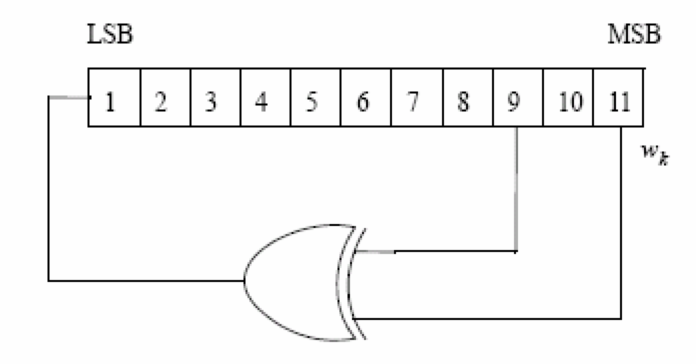

Figure 5 PRBS for pilot modulation

Figure 5 shows the PRBS used to produce a random sequence, w_k, that will be used for subcarrier randomization (after QAM mapping) and pilot modulation in the following sections. The polynomial for the PRBS generator shall be $X^{11} + X^{9} + 1$. The initialization vector for the memory shall follow the steps in [2].

2.1.4.2 Data Modulation

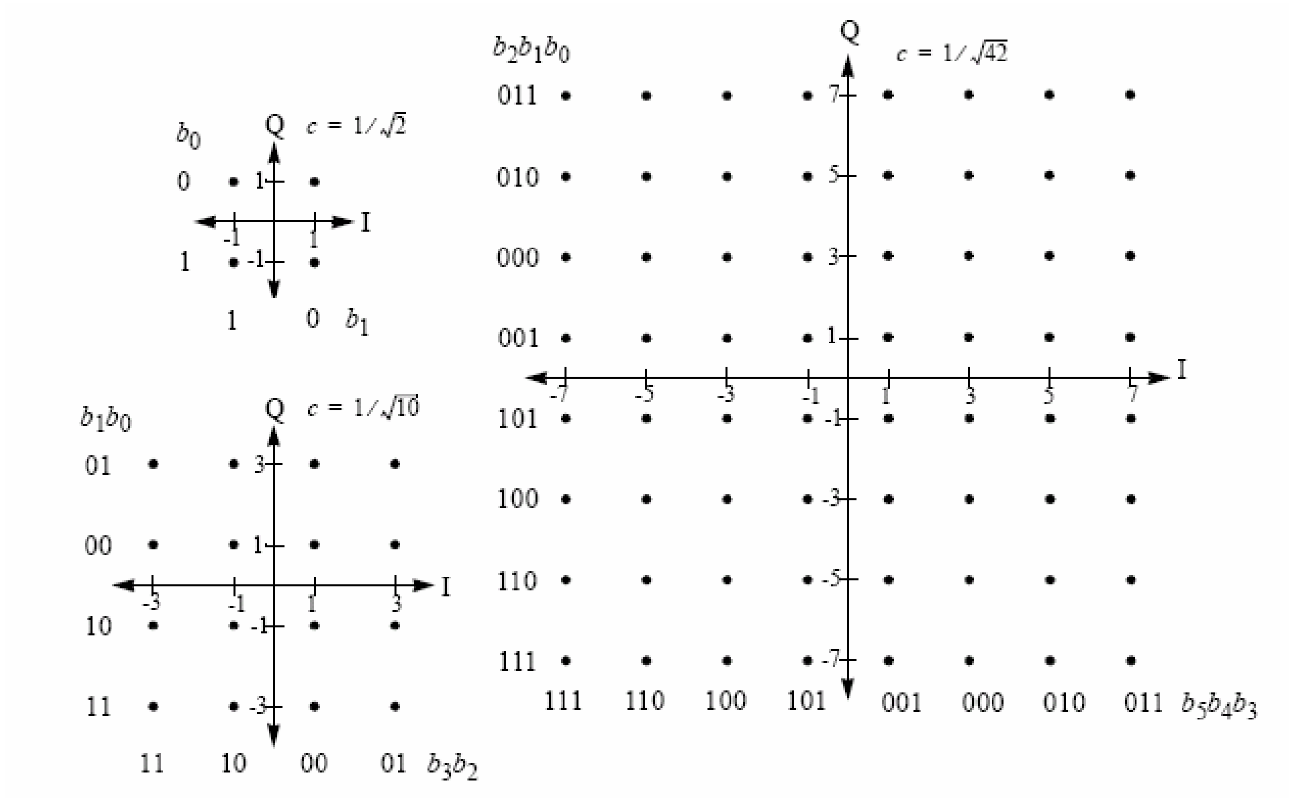

Figure 6 QPSK, 16QAM and 64QAM constellations point

Table 2 Modulation-dependent normalization factor

Modulation	K_{MOD}
QPSK	$1/\sqrt{2}$
16-QAM	$1/\sqrt{10}$
64-QAM	$1/\sqrt{42}$

OFDM subcarriers are modulated using QPSK, 16-QAM and 64-QAM constellations. The encoded and interleaved serial input data are divided into groups of N_{BPSC}, i.e. 2, 4 or 6, bits which will then be converted into a complex number, $I + Q \times j$, representing either a QPSK, 16-QAM, or a 64-QAM constellation point. The mappings for QPSK, 16-QAM and 64-QAM are shown in Figure 6. Finally, the resulting complex value is normalized by multiplying it by the normalization factor, K_{MOD}, specified in Table 2. The constellation-mapped data shall be subsequently modulated onto the allocated data subcarriers and each subcarrier is multiplied by the factor $2 \times (1/2 - w_k)$ according to the subcarrier index k. w_k is derived using the method described in the previous section about permutation definition.

2.1.4.3 Pilot Modulation

In the downlink, the pilot shall be transmitted with a boosting of 2.5 dB over the average non-boosted power of each data tone. The pilot subcarriers shall be modulated with sequence w_k defined in previous section using the following equation:

$$\begin{aligned} \mathrm{Re}\{c_k\} &= \frac{8}{3}\left(\frac{1}{2} - w_k\right) \\ \mathrm{Im}\{c_k\} &= 0 \end{aligned} \qquad (3)$$

In the downlink, for PUSC, FUSC, AMC and optional FUSC permutations, all pilots (of the segment, in case of PUSC) shall be modulated, whether or not all the sub channels are allocated in the DL-MAP. For AMC permutation in AAS zone, the BS is not required to modulate the pilots that belong to bins that are not allocated in the DL-MAP, or are allocated as gaps (UIUC=13).

2.1.5 Subchannel Allocation

There are many types of subchannel allocation mechanism, depending on whether the transmission is uplink or downlink. For the downlink, the two main subchannel allocation methods are:

- Partial Usage of SubChannels (PUSC) where part of the subchannels is allocated to the transmitter.
- Full Usage of SubChannels (FUSC) where all of the subchannels are allocated to the transmitter.

FUSC employs full channel diversity by distributing data subcarriers to subchannels using a permutation mechanism designed to minimized interference between cells. It is somewhat similar to the idea of classical frequency hopping technique. As for PUSC, subchannels are divided and assigned to 3 segments that can be allocated to sectors of the same cell. Similar to FUSC, a permutation mechanism is applied for allocating subcarriers to subchannels to harvest interference averaging and fast fading averaging effect. The reader may refer to [2] for details of the permutation mechanism.

Figure 7 shows the PUSC data frame consisting of L subchannels across the time interval. Data can be transmitted over the subchannels depicted in the figure. In PUSC mode, a data slot is composed of one subchannel and two OFDMA symbols. The data region is the allocated area for user data transmission. The mapping of encoded data blocks onto the subchannels is depicted in Figure 7. The mapping shall follow the order in Figure 7 (vertical direction first then proceed to the next two OFDMA time symbols).

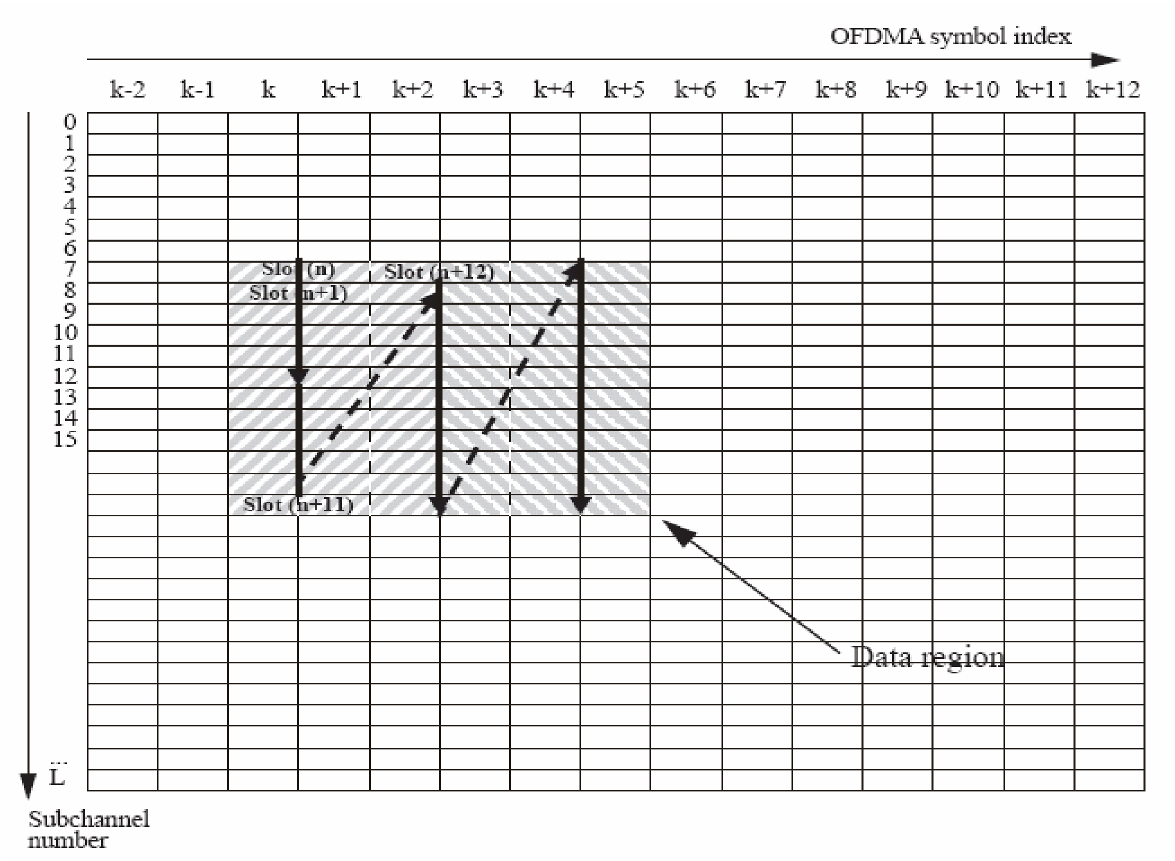

Figure 7 Example for mapping of encoded data blocks to subchannels in downlink PUSC mode

2.1.6 IFFT / FFT

The IFFT block transforms the data from frequency to time domain using inverse fast fourier transforms at the transmitter while the FFT performs the reverse operation at the receiver. A highlight of the IEEE 802.16e standard is the idea of Scalable OFDMA (SOFDMA) where the FFT size can be adjusted while fixing the subcarrier frequency spacing at a particular value. This is advantageous for supporting a wide range of bandwidths to flexibly address the need for various spectrum allocations ranging from 1.25 Mhz to 20MHz. The relevant FFT parameters for FUSC and PUSC schemes are shown in Table 3 and Table 4 respectively.

Table 3 FUSC FFT parameters

System Parameter	Description			
FFT size	128	512	1024	2048
No. of guard subcarriers	22	86	173	345
No. of used subcarriers	106	426	851	1703
No. of data subcarriers	96	384	768	1536
No. of pilot subcarriers	9	42	83	166

Table 4 PUSC FFT parameters

System Parameter	**Description**			
FFT size	128	512	1024	2048
No. of guard subcarriers	43	91	183	367
No. of used subcarriers	85	421	841	1681
No. of data subcarriers	72	360	720	1440
No. of pilot subcarriers	12	60	120	240

3 Physical Layer Simulation Results and Analysis

This section describes the physical layer performance of the WIMAX platform. The simulation parameters used are the WIMAX platform parameters, which can be found in Table 5

Table 5 System parameters of WIMAX platform

System Parameter	Description
Duplexing	TDD
Multiple Access	OFDMA
Subcarrier permutation	PUSC
Carrier Frequency	2.3 GHz
Channel Bandwidth	8.75 MHz
FFT	1024
Subcarrier Spacing	9.765625 KHz
Symbol Duration, T_S	102.4 µs
Cyclic Prefix, T_G	12.8 µs
OFDM Duration	115.2 µs
TDD Frame Length	5 ms
No. Symbols in a Frame	42
DL / UL ratio	27 / 15
TTG / RTG	121.2 / 40.4 µs

Other simulation parameters and assumptions are below:

- ITU Vehicular A and Vehicular B channel models [5].
- The simulations assume the following phase noise model: A spectral mask of -85dBc/Hz flat up to 10 KHz and then reducing 20dB/dec up to –120dBc/Hz.
- Perfect channel knowledge is assumed.

Unless otherwise specified, the parameters above are assumed for the simulation results presented in the following sections.

3.1 Performance of WIMAX at different mobile speeds

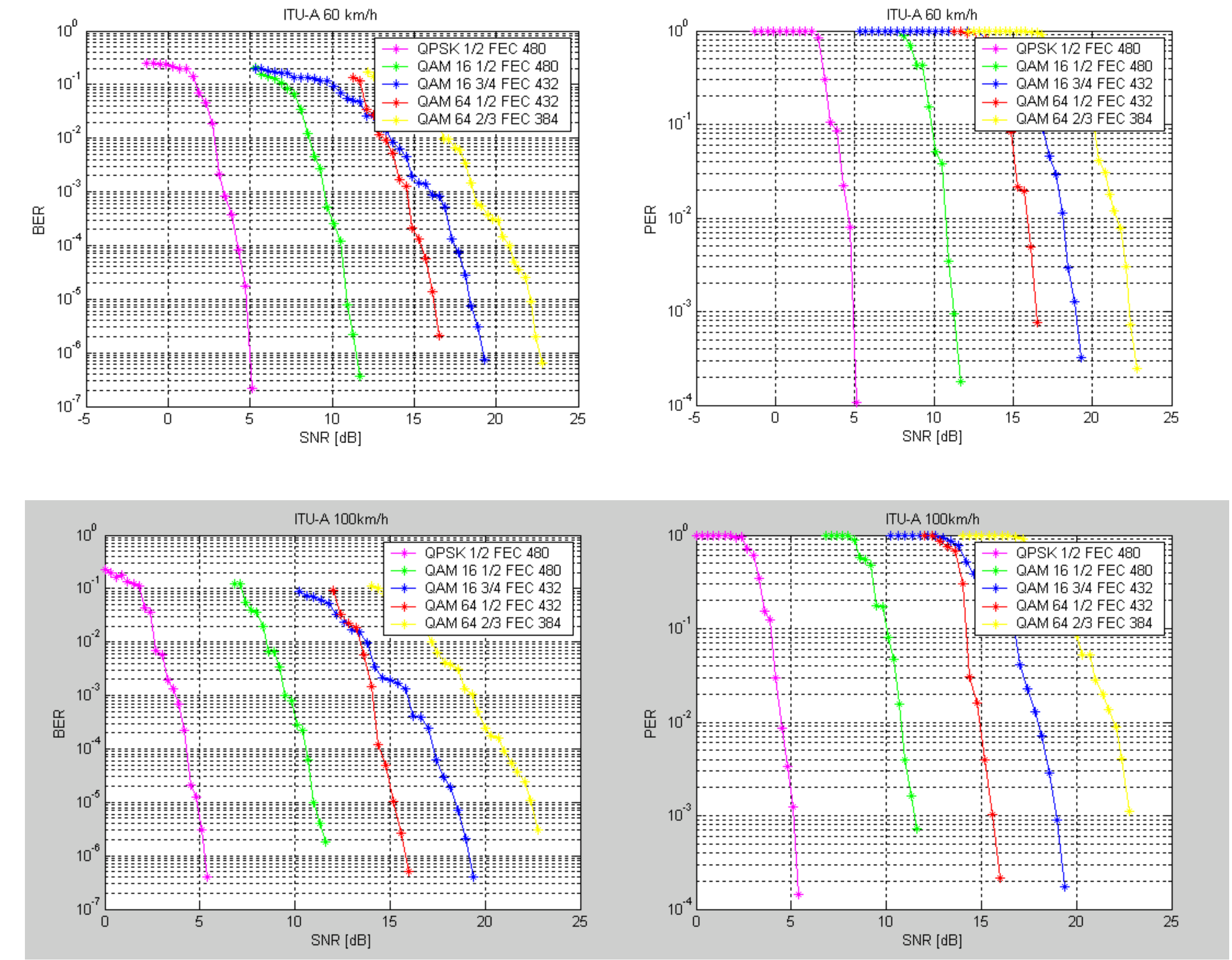

Figure 8 BER and PER performance of PUSC scheme at mobile speeds 60kmph and 100kmph

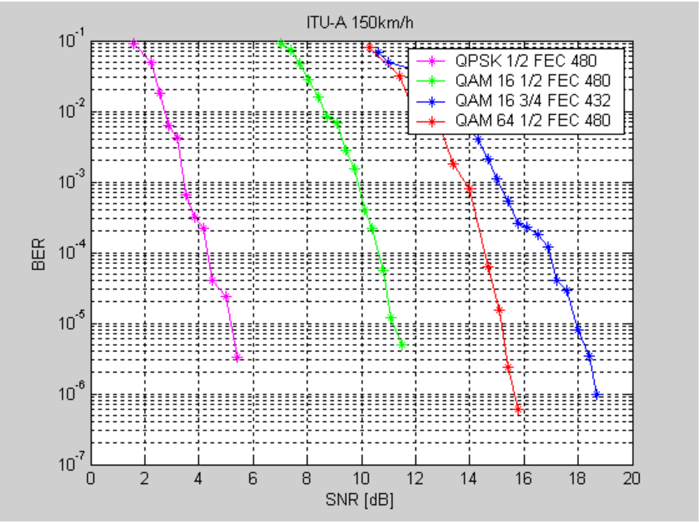

Figure 9 BER performance of PUSC scheme at mobile speed 150 kmph

Figure 8 shows the performance of PUSC schemes for mobile speeds of 60 kmph and 100 kmph respectively in the ITU Vehicular A channel. The FEC blocks size are indicated in the graph legends, in the range of 384 to 480bits, depending on the modulation and coding setting. It can be seen that WIMAX is fairly robust to doppler spread and the performance loss is minimal. Figure 9 further shows the BER performance at a mobile speed of 150 kmph. Comparing Figure 8 and Figure 9, the performance loss is still small, even when the QAM64 1/2 are used.

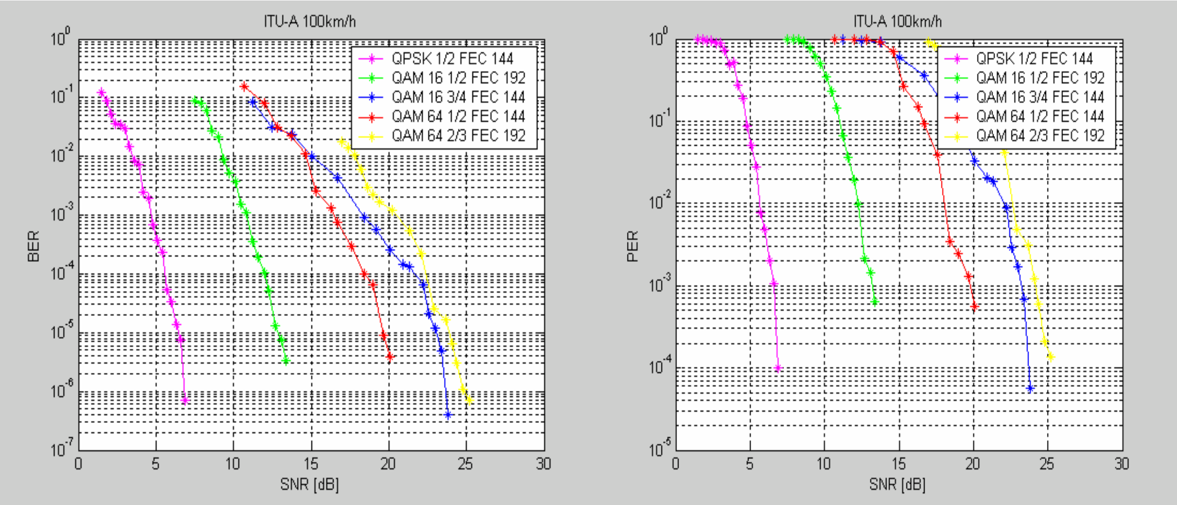

Figure 10 BER and PER performance of PUSC scheme at 100kmph

Figure 10 further shows the result for ITU-Veh A BER and PER at 100km/h using FEC blocks of 144 and 192 bits. When compared to Figure 8, there a small performance lost. This is due to small block size being used and for the class of CTC codes, larger block size will have a better BER / PER performance. Nevertheless, the performance is still robust to Doppler spread even at 100 km/h.

3.2 Performance of WIMAX with different channel models

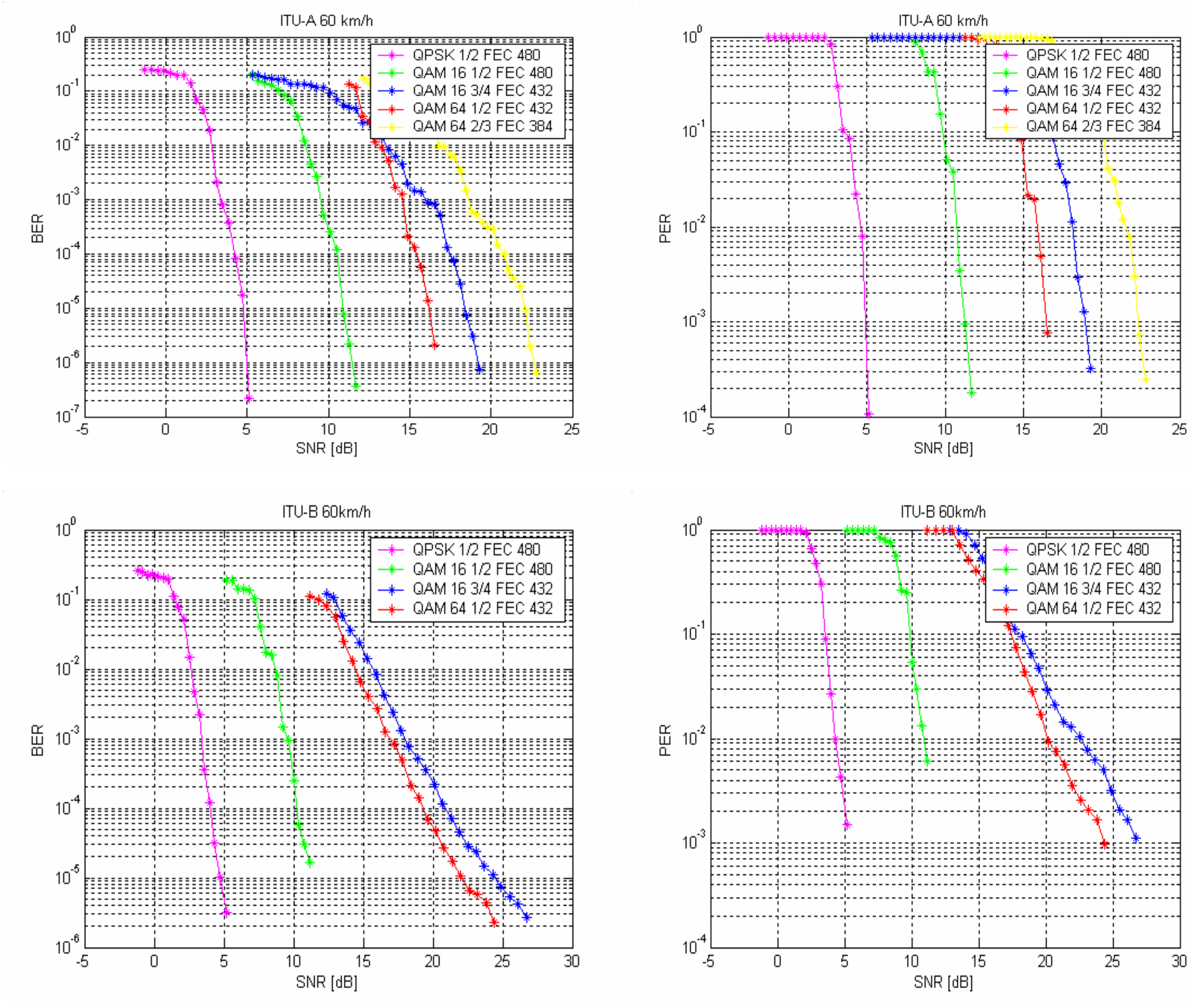

Figure 11 BER and PER performance with ITU Veh-A and Veh-B channel

Figure 11 shows the BER and PER performance of ITU Veh-A and Veh-B channel at 60km/h using FEC blocks of length 384 - 480bits as defined in the graphs, depending on the modulation and coding setting. It can be seen that performance for ITU Veh-B beyond 16QAM-3/4 model has large performance degradation when compared to ITU Veh-A channel. This is due the larger channel echo of ITU-Veh B model than the guard interval of the defined system parameters in Table 5. While the defined system parameters are targeting smaller cell size and low antenna, the figure shows that the system is still usable (and robust for mode below 16QAM-1/2) at larger cell size of normal GSM/3G system. Note that this problem can be solved easily by using a larger guard interval at the cost of reduced bit rate if a larger cell size is desired.

3.3 Performance of WIMAX using different channel coding schemes

This section presents the simulation results comparing different channel coding schemes including Convolutional Coding (CC), duo-binary Convolutional Turbo Coding (CTC), and Low Density Parity Check (LDPC). The simulation results are for a FFT size of 2048 using the FUSC scheme, either in AWGN or ITU Vehicular B channels. The channel bandwidth and mobile speed are 10MHz and 120 kmph respectively. All the simulation results presented in this section use a carrier frequency of 2.35GHz.

The other assumptions of the simulation are:

- Perfect channel estimation
- Perfect power control is assumed.
- The decoders used for CC, CTC and LDPC are Viterbi, Max-Log-Map and Flooding respectively.
- For graphs of 2304 bits, CC performance is presented with 576 bits
- For graphs of 2304 bits, CTC code is presented with 1728/1152 bits due to code size mismatch.
- Simulations were run up to PER of 10^{-2} with at least 40 errors, or BER of 2×10^{-4}, with at least 100 bit errors.

For LDPC, the assumptions are below:

- Structure: block matrix with permutation sub-block
- Encoding: parity-check matrix is lower triangular – completely linear encoding.

LDPC is flexible and any coding rate between ½ and ¾ is possible for coding block size of 2304, 1728, 1152, 576. Other lengths and rates can be obtained via shortening and puncturing of any of the basic codes.

$$H_b = \left[\begin{array}{cccccccccccccccc}
P^5 & 0 & P^{20} & 0 & P^2 & 0 & 0 & P^8 & P^{20} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & P^4 & 0 & 0 & P^{21} & 0 & 0 & P^{20} & P^{13} & P^{13} & 0 & 0 & 0 & 0 & 0 & 0 \\
P^{11} & 0 & 0 & P^7 & 0 & P^{21} & 0 & 0 & 0 & P^9 & P^9 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & P^9 & 0 & 0 & 0 & P^{14} & P^9 & 0 & 0 & P^7 & P^4 & 0 & 0 & 0 & 0 \\
0 & P^{13} & 0 & 0 & 0 & P^2 & 0 & 0 & 0 & P^{12} & 0 & P^{10} & P^{12} & 0 & 0 & 0 \\
P^5 & 0 & P^9 & 0 & 0 & 0 & P^{10} & 0 & 0 & 0 & 0 & 0 & P^4 & P^3 & 0 & 0 \\
0 & P^1 & 0 & 0 & P^{10} & 0 & 0 & 0 & P^{18} & 0 & 0 & 0 & 0 & P^3 & P^{13} & 0 \\
P^5 & 0 & 0 & P^{15} & 0 & P^{18} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & P^7 & P^{st}
\end{array}\right]\begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \end{array}$$

For rate ½ use base matrix as it is. As for rate 0.6875 merge rows (1,5), (2,6), (3,7) and for rate ¾ merge rows (1,5), (2,6), (3,7), (4,8).

Note that the graphs presented here are provided by the consortium partner, Runcom, during an early study of LDPC [6] within the IEEE 802.16 work group. Results are generated by different companies using the same “Belief Propagation” algorithm. It may not strictly adhere to LDPC in the standard, but it does gives a performance indication on this kind of code when compared to CC and CTC. Similar observation can be found in [7].

3.3.1 BER and PER performance for QPSK1/2 on DL for short and long blocks with AWGN

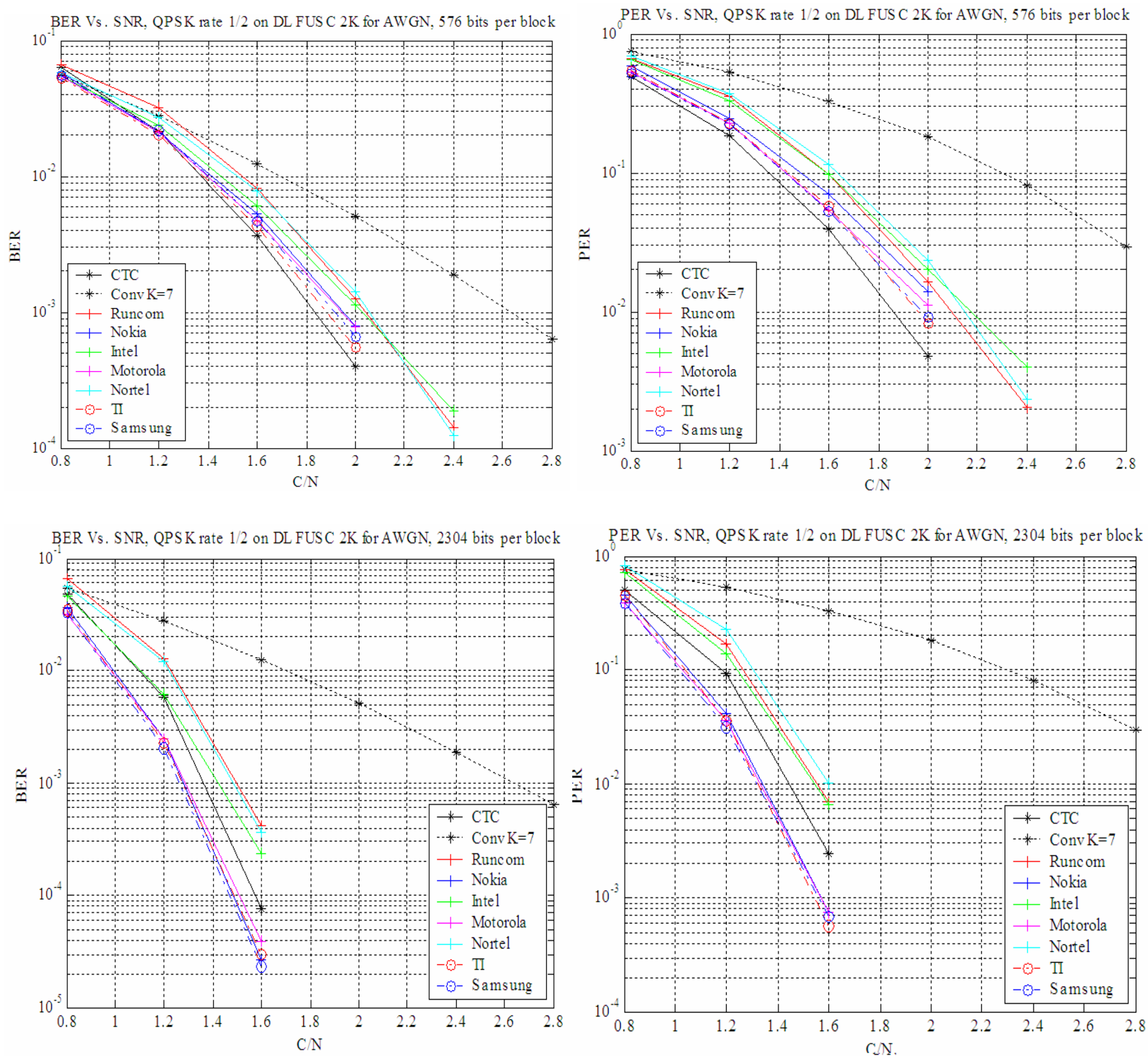

Figure 13 BER and PER performance for QPSK-1/2 in AWGN channel

Figure 13 shows the result of BER and PER performance in AWGN channel. As depicted in Figure 13, it can be seen that the gain of LDPC over CTC is less than 1dB, and is only present when a large coding block size is used. When a short coding block is considered, LDPC does not have an advantage over the CTC FEC. LDPC has more than 1 dB gain over CC. This observation is found to be consistent with simulations conducted for other coding rates and channel models in the following sections. Note that the results are generated by different companies using the same “belief propagation” algorithm. The generated results are fairly consistent, with 0.5 – 1dB difference amongst curves from different companies. This is because the parity check matrix used by each company were different.

3.3.2 BER and PER performance for QPSK1/2 on DL for short and long blocks with ITU Vehicular B

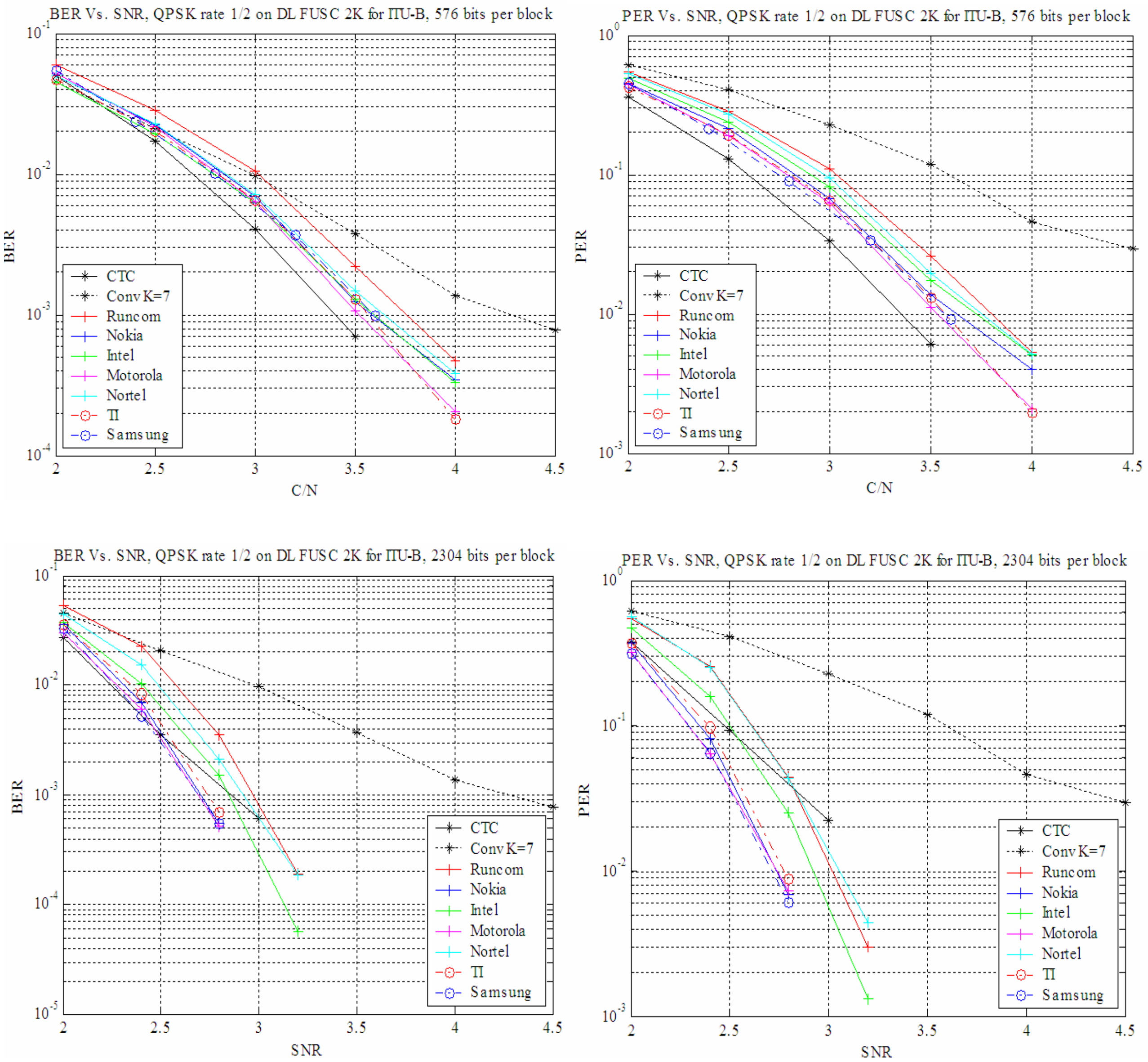

Figure 14 BER and PER performance for QPSK-1/2 in ITU Vehicular B channel

Figure 14 shows the result of BER and PER performance with modulation and coding rate QPSK 1/2 in ITU Vehicular B channel. Again, LDPC does not have an advantage over CTC for short coding blocks, and less than 1dB gain is observed when a large coding block is used.

3.3.3 BER and PER performance for 16QAM 1/2 on DL for short and long blocks with ITU Vehicular B

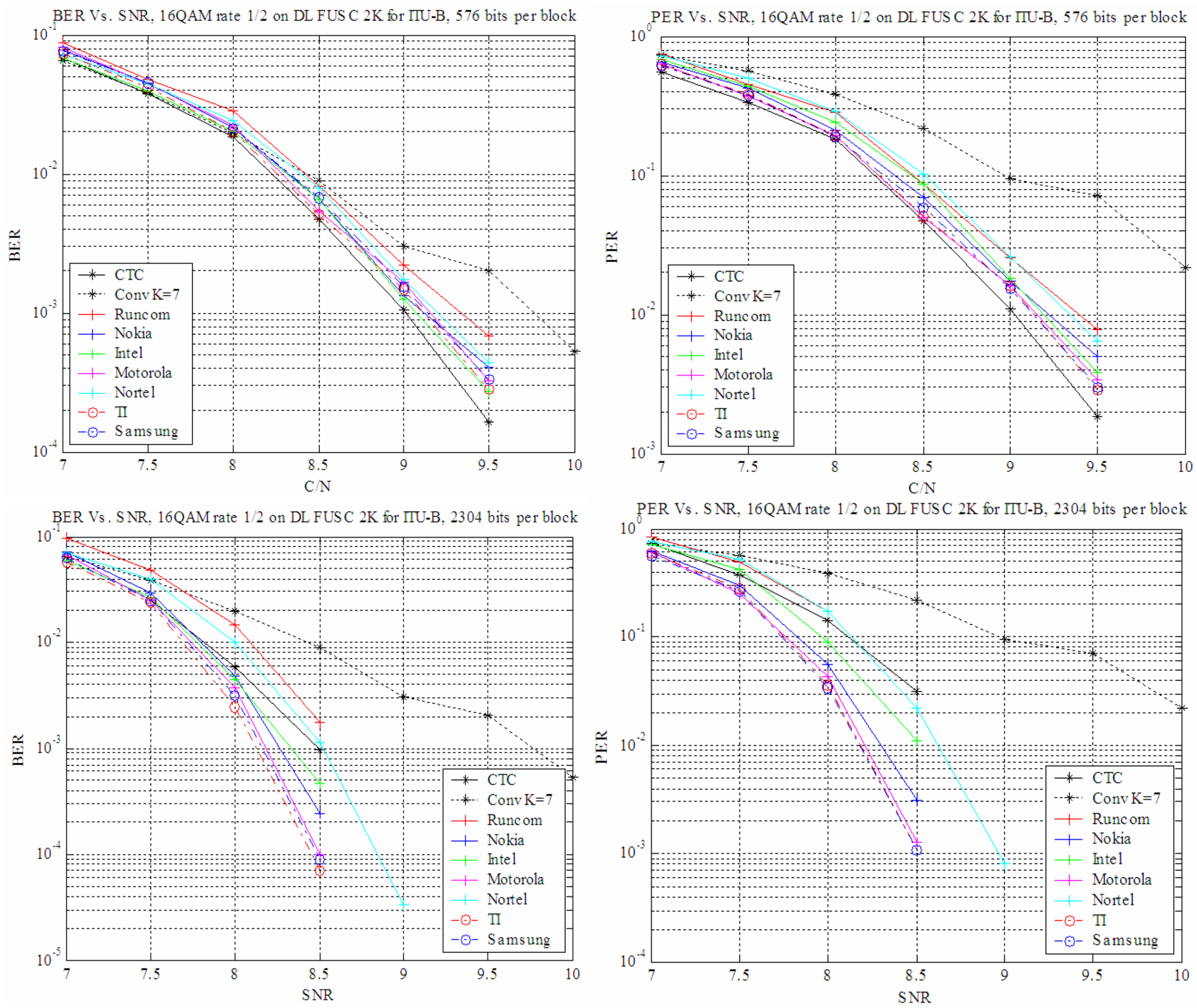

Figure 15 BER and PER performance for 16QAM-1/2 in ITU Vehicular B channel

Figure 15 shows the result of BER and PER performance with modulation and coding rate 16QAM 1/2 in ITU Vehicular B channel. It shows that the gain of LDPC over CTC is not large.

3.3.4 BER and PER performance for 64QAM 1/2 on DL for short and long blocks with ITU Vehicular B

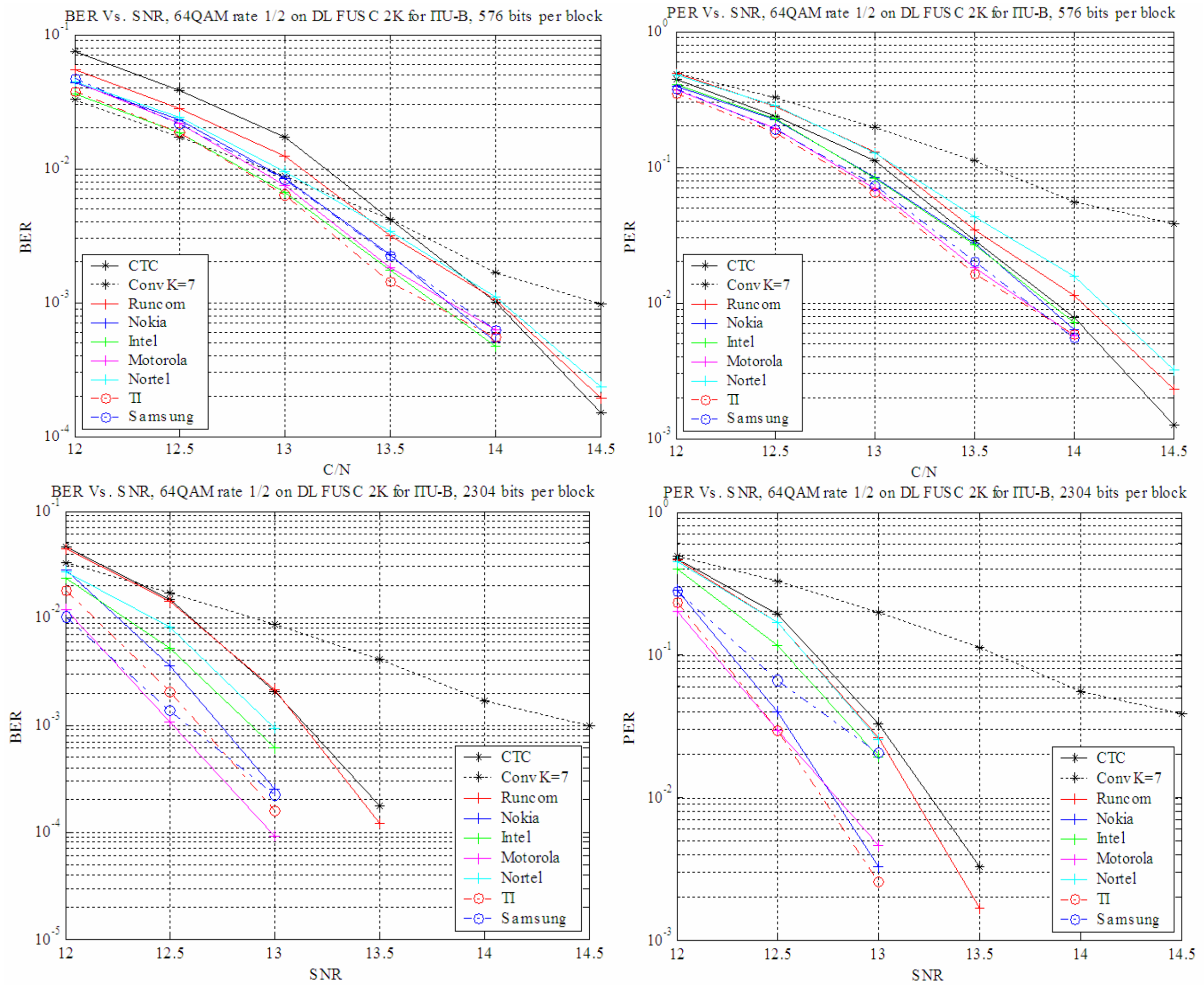

Figure 16 BER and PER performance for 64QAM-1/2 in ITU Vehicular B channel

Figure 16 shows the result of BER and PER performance with modulation and coding rate 64QAM 1/2 in ITU Vehicular B channel. From the simulation results in the last few sections, for modulation and coding rates QPSK-1/2, 16QAM-1/2 and 64QAM-1/2, it can be seen that the results are fairly consistent depicting a marginal gain with LDPC compared to CTC.

3.3.5 Conclusions

Extentsive simulations have been conducted to compare the performance of CC, CTC and LDPC. Various coding block sizes and channel models are considered. The simulation results clearly shows that LDPC gain over CTC FEC is less than 1 dB and is present only when long blocks are transmitted. For short blocks there is no advantage for LDPC compared to CTC FEC. LDPC has more than 1 dB gain compared to CC. In an OFDMA system where the allocation of slots (block size) to the user for transmission can be small, LDPC does not really have a significant advantage over CTC in this sense. It should be noted that CTC is in favoured within the WIMAX forum [8]. Although the results are generated by different companies using the same “belief propagation” algorithm, the resuts showed small differences (0.5-1dB). This is because the parity check matrix used by each company were different, but the results are fairly consistent.

4 Error Pattern Files Generation

This section discusses the data flow over the WIMAX protocol layers. Following the data flow discussion, the simulator software of the protocol layer – coded in C++ (using the IT++ signal processing library) – is outlined, and the functionality of each module is outlined. Finally, the parameters used for trace generation and the trace format are described.

4.1 Data Flow over Protocols Layer

This section describes how IP Packets transit through the WIMAX system. We discuss this in two topics. The first topic is packet flow over the MAC layer, where packets can be fragmented into different MAC frames, packed together into a single MAC frame, or left untouched and transmitted in a single MAC frame. At the physical layer, multiple MAC frames for a single user can be concatenated into a single data burst for transmission over the allocated data region in the physical layer frame. Both topics are discussed in the following sections.

4.1.1 Data Flow over the MAC Layer

In the MAC layer, IP packets are encapsulated into MAC frames with headers and CRCs added. There are three types of encapsulation method, namely generic packet encapsulation, packet encapsulation with fragmentation and packet encapsulation with packing. Each encapsulation method is described in detail in the following sections.

4.1.1.1 Generic Packet Encapsulation

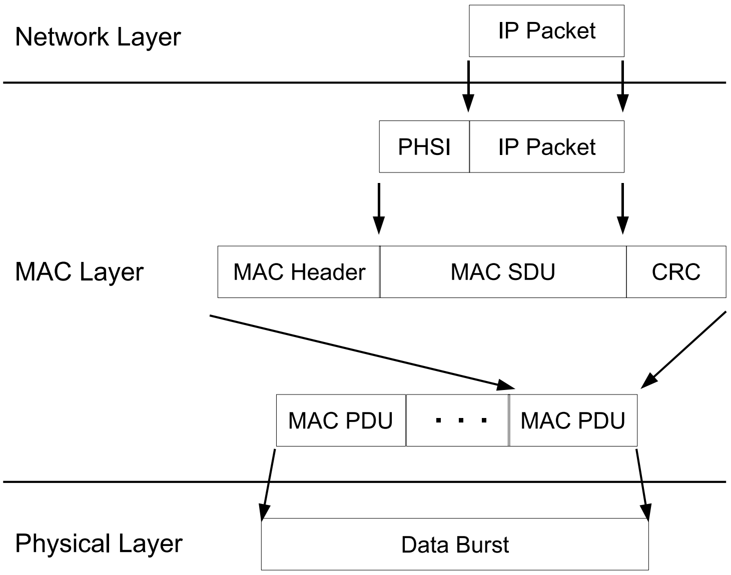

Figure 17 Data flow over MAC layer for generic case

Figure 17 shows how the packet is encapsulated in the MAC layer, before it is passed to the physical layer. The Packet Header Suppression Index (PHSI) of 1 byte indicates whether header compression is in used. The MAC header contains information about the payload, while the Cyclic Redundancy Check (CRC) provides parity bits for error detection. The MAC header and CRC are 6 bytes and 4 bytes respectively. Note that the maximum size MAC Service Data Unit (SDU) in the diagram is 2041 bytes.

4.1.1.2 Packet Encapsulation with Fragmentation

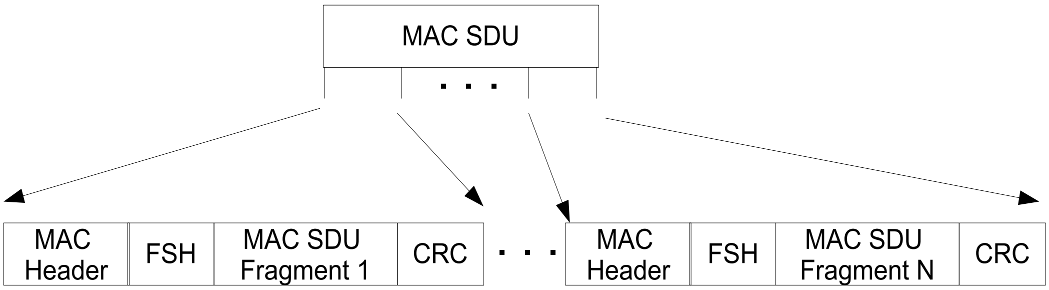

Figure 18 Packet encapsulation with fragmentation

Fragmentation is the process whereby the MAC SDU is split into smaller units. This process allows efficient use of available bandwidth relative to the QoS requirements of a connection. The encapsulation process is similar to Figure 17 except that there is an extra 1 byte header called Fragmentation SubHeaders (FSH). The encapsulated packet is shown in Figure 18.

4.1.1.3 Packet Encapsulation with Packing

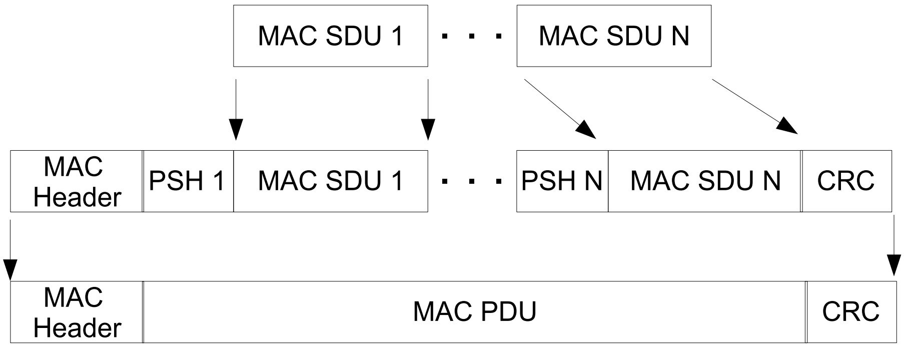

Figure 19 Packet encapsulation with packing

Packing is the process whereby several MAC SDUs are packed into a single MAC frame. A Packing SubHeader (PSH) of 2 bytes is added for each incoming MAC SDU. Figure 19 shows the corresponding packet encapsulation with packing enabled. Similar to the fragmentation function, packing allows the efficient use of available bandwidth. Note that the above diagram assumes variable length MAC SDU, which is in fact the case for incoming packets carrying video data.

4.1.2 Data Flow over the Physical Layer

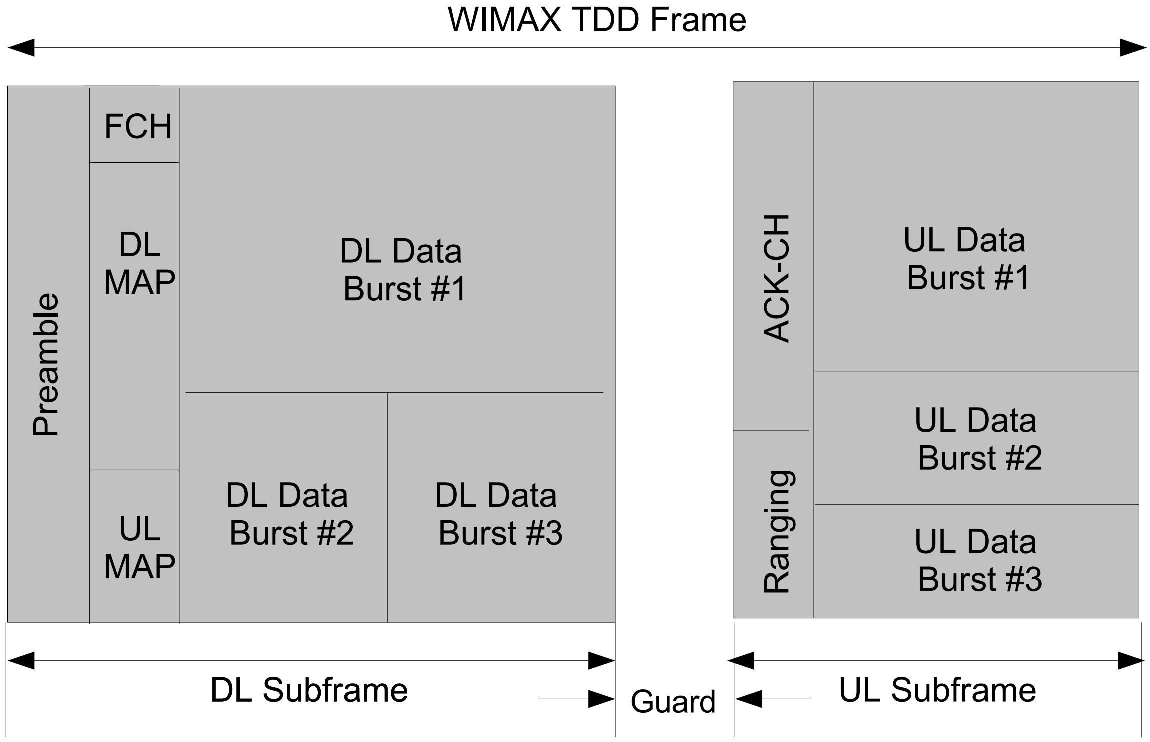

Figure 20 WIMAX Physical Layer TDD Frame

Figure 20 shows the physical layer Time Division Duplex (TDD) frame of WIMAX system. It contains DownLink (DL) Subframes and UpLink (UL) frames with various regions for performing protocol functions and data transmissions. Basically the DL Data Burst #x region is the allocated data region where a user can transmit data. A data burst, as shown in Figure 17, generated by users, is placed at the respective DL Data Burst, depending on the MAC scheduler allocation decision. The allocation decision is transmitted in DL-MAP section of DL Subframe.

4.2 Simulator Software Description

The simulation software developed in the project for trace generation is described below. The function of each module is outlined in Table 6.

Table 6 Simulation software files description

Files	Descriptions
scrambler.h scrambler.cc	Scramble and descramble the data before channel coding
wimaxctc.h wimaxctc.cc repetition.h repetition.cc	Perform FEC operation using CTC and repetition coding.
interleaver.h interleaver.cc	Perform interleaving and de-interleaving
modem.h modem.cc	Perform QPSK, 16QAM and 64QAM mapping
fusc.h fusc.cc pusc.h pusc.cc	Perform subcarriers permutation using FUSC and PUSC. Then map incoming bits into data regions allocated for the users.
wimaxfdm.h wimaxofdm.cc transceiver.h transceiver.cc	Perform OFDM operation which includes zero padding, IFFT / FFT, and cyclic prefix insertion / removal
chanmodel.h chanmodel.cc	Initialize TDL channel model of IT++ and set the parameters according channel profile e.g. ITU Veh-A.

fileio.h fileio.cc	Calculate bit error and slot error and save the error pattern onto the hardisk.
helper.h helper.cc	Consists of a set of useful functions for other modules
simparam.h simparam.cc	Defines all the simulator parameters
wimax_pusc.cc wimax_pusc_trace.cc	Main simulator file integrating individual modules above for estimating BER and generating error pattern files.

An example parameter file for the simulator for PUSC permutation is below:

```
% Parameter file for 802.16e simulation
%
%-------------------------------------------------------------------------
% OFDM Parameter
% Nfft(nil)  = number of fft point
% Ncp(nil)   = number of sample for cyclic prefix
% Nused(nil) = number of subcarriers used, pilot + dc + data
% Ntimesymb(nil)   = number of ofdm symbol in 1 frame, must be odd number
%                    for fusc & even number for pusc
% Nallocslot(nil)  = number of allocated slot in time dimension
% Nallocchan(nil)  = number of allocated subchannel in frequency dimension
% carrierfreq(Hz)  = center carrier frequency
% samplingfreq(Hz) = sampling rate of the system
%
% Note: Total allocated slots is Nallocslot x Nallocchan
% Note: Please check 802.16-2005 for number of allocatable subchannels
% Note: In FUSC and PUSC, the number of allocatable slots are Ntimesymb
% and Ntimesymb/2
%

Nfft         = 1024;
Ncp          = 128;
Nused        = 841;
Ntimesymb    = 42;
Nallocslot   = 21;
Nallocchan   = 30;
carrierfreq  = 2.35e9;
samplingfreq = 10.0e6;

%-------------------------------------------------------------------------
% Channel Coding Parameter,
% MCS_1  =  QPSK 1/2  repetition x 6  MCS_2  =  QPSK 1/2  repetition x 4
% MCS_3  =  QPSK 1/2  repetition x 2  MCS_4  =  QPSK 1/2
% MCS_5  =  QPSK 3/4                  MCS_6  = 16QAM 1/2
% MCS_7  = 16QAM 3/4                  MCS_8  = 64QAM 1/2
% MCS_9  = 64QAM 2/3                  MCS_10 = 64QAM 3/4
%
% MCS_1  = 0,  MCS_2  = 1,  MCS_3  = 2,  MCS_4  = 3,  MCS_5   = 4,
% MCS_6  = 5,  MCS_7  = 6,  MCS_8  = 7,  MCS_9  = 8,  MCS_10  = 9
%

MCSMode = 3;


%-------------------------------------------------------------------------
% Channel parameters
% snr(dB)     = signal to noise ratio
% speed(km/h) = speed of terminal
%
```

```
% chanprofile = selects the channel profile i.e. ITU Ped/Veh,
% using the value below
% ITU_Vehicular_A  = 0, ITU_Vehicular_B  = 1,
% ITU_Pedestrian_A = 2, ITU_Pedestrian_B = 3
%
% Note: Please see channel.h in IT++ library for other channel profile
%

snr         = 30;
speed       = 30;
chanprofile = 0;


%-------------------------------------------------------------------------
% Misc
% rndseed(nil) = set the seed of simulator
% IDCell(nil)  = value between [0,31], used for calculation of
%                permutation, fusc and pusc
% prbs_state   = bvec with 11 bits, to initialized the PRBS state, used
%                for subcarrier randomization
% num_of_frame = number of frames to run in simulation
%

rndseed       = 13579;
IDCell        = 3;
prbs_state    = "1 0 1 0 1 0 1 0 1 0 1";
num_of_frames = 5;
```

Most of the parameters of the system have been described in the comment of the parameter file or they are self-explanatory as most notations use the standard notation whenever possible. Note that Ntimesymb specifies the number of OFDMA symbols in a TDD frame. For the parameters above, a 5ms TDD frame corresponds to 42 time symbols. Nallocchan and Nallocslot both define a 2D region for data transmission with total allocated slots be Nallocchan x Nallocslot. Maximum subchannels that can be set for Nallocchan is 30 for 1024 FFT size where as for Nallocslot, it is Ntimesymb/2 or 42/2 = 21 since one data slot is one subchannel and two OFDMA time symbol in PUSC case.

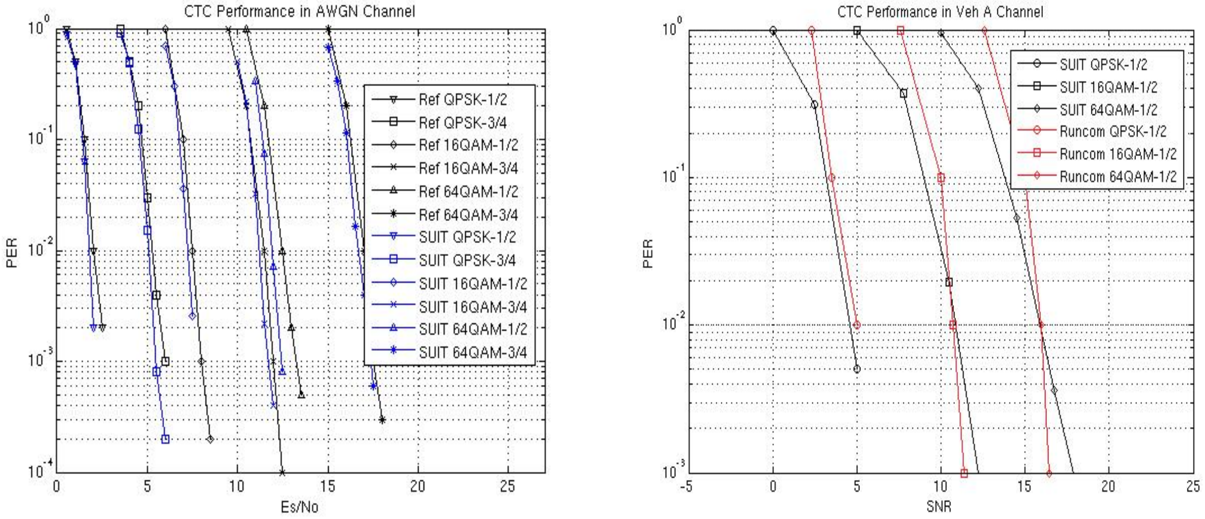

Figure 21 PER performance comparison of SUIT simulator to literature for validation

The SUIT simulator is compared to the literature for validation in Figure 21. In the AWGN case, the reference curves (Ref) are obtained from [9] for comparison. For the Veh A case, the Runcom reference curves are taken from Figure 8, in section 3. Note that the SUIT simulator is developed separately so that the source code can be made available to all consortium partners, while the results in Section 3 are provided by consortium partner, Runcom, using a proprietary simulator. It can be seen that the SUIT simulator is sufficiently accurate when compared to existing simulators.

4.3 Pre- Generated Trace Format

The developed WIMAX simulator has been used to generate error pattern files. The format of the trace matches the TDD frame format of Runcom's WIMAX platform since it will be used for the trial. The platform uses the system parameters shown in Table 5. The DL Subframe format of Runcom's WIMAX platform is illustrated in Figure 22. The value on the vertical axis is the subchannel number, while the value on horizontal axis is the time symbol number. As PUSC have been used in the platform, one data slot is equal to one subchannel and two time symbols. Thus, the DL Subframe is a matrix of 30x13 data slots, excluding the preamble symbol.

In order to reduce data storage requirements, the error pattern is saved in the form of a data slot error pattern instead of a bit error pattern. The data slot error pattern is obtained by comparing all the data bits within an original data slot to the transmitted data slot. If there is any bit error within the data slot, it is declared as an error. Note that we have not specifically assumed any IP packet size or data burst size within a physical layer frame. Also, no MAC layer packet encapsulation procedures have been performed. This decision was made to allow flexibility on the choice of packet size and the data throughput during video transmission simulation. Data slots can be aggregated to different MAC frame sizes for different packet sizes, and different burst sizes for different data throughput. This also allows the study of efficient packetization scheme for video transmission. The user of the error pattern trace will need to perform the procedures outlined in Section 4.1.1. Maximum FEC code block size has been assumed for the trace generation. The maximum code block size is MCS mode dependent, as defined in the IEEE 802.16e-2005 standard.

Table 7 Parameters used for trace generation

Parameter	Values
Length of Trace (s)	15
Permutation	PUSC
Channel Coding	CTC
Terminal Speed (km/h)	60, 120
Test Environment	ITU Vehicular A
MCS Mode	QPSK-1/2, QPSK-3/4, 16QAM-1/2, 16QAM-3/4, 64QAM-1/2
SNR Range	0 to 30 dB which will have 5 to 7 data points for each MCS mode

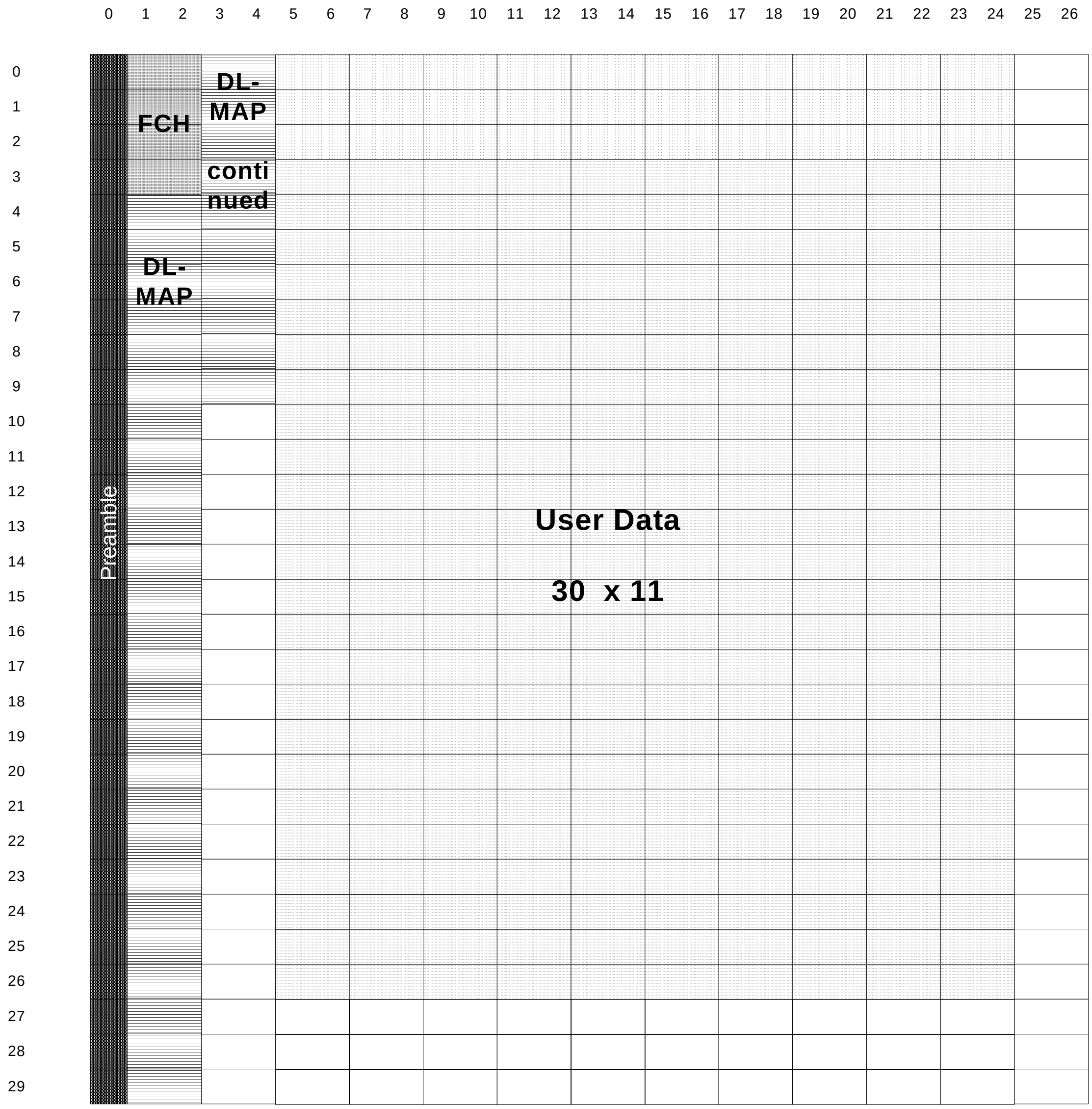

Figure 22 Downlink subframe of Runcom WIMAX system

For error pattern generation, the system parameters in Table 5 and the simulation parameters in Table 7 are used. Nevertheless error patterns for other scenarios may be generated easily with the SUIT WIMAX simulator made available to the consortium. For each MCS mode, 5 – 7 data points, each representing different SNR/BER level, are generated. In the generated error pattern files, symbol “1” refers to a data slot error while symbol “0” means that there is no error. The traces are named using the following context, “trace_mcsX_Y” where mcsX refers to the MCS mode X while Y refers to the SNR value. Companion files “ber.X” that are submitted together with the traces contain the corresponding BER/PER values for individual trace generated with MCS mode X and SNR Y. Note that the error pattern is saved in C++ “long integer” format that is 4 bytes (32 bits) long. Hence, every long integer value represents a slot error pattern across the 30 subchannels in one time symbol by using a bit XOR operation. In the trace directory, software files fileio_c.h and fileio_c.cc are provided for extracting error pattern from the error trace file.

5 Conclusions

This deliverable describes the WIMAX simulation models developed within the project. The first section of this deliverable provides an overview of the WIMAX baseband model and its specification. The implemented functions are subset of IEEE 802.16e standard since not all the features are relevant or necessary for the purpose of study in this project. In the second section, the physical link level performance of WIMAX is presented. It can be seen that WIMAX is fairly robust to high vehicular speeds (doppler spread) from the simulations carried out for 150 kmph. There is some performance degradation for existing system parameters when the popular GSM/3G, large cell size, ITU Veh B channel is considered for 16QAM-3/4 mode and above. This is because the defined parameters are targeting smaller cell sizes (smaller delay spread) than the ITU Veh B channel. Nevertheless, the system is still robust for 16QAM-1/2 mode and below. A comparison of different coding schemes, Convolutional Coding (CC), Convolutional Turbo Coding (CTC) and Low Density Parity Check (LDPC) are made. It has been shown that the gain of LDPC over CTC is marginal (less than 1dB), and only when the block size is large. In OFDMA system where the allocated resource unit can be small, there is no huge advantage in LDPC scheme in such scenario. The third section provides an overview of the process of encapsulation of IP packets in the MAC/PHY protocol layer. It also describes the implemented trace generation software, and the format of the error pattern. The error pattern follows the format of the Runcom platform which will be used for the trial. The generated trace considers QPSK-1/2, QPSK-3/4, 16QAM-1/2, 16QAM-3/4 and 64QAM-1/2 for the ITU Vehicular A channel. Nevertheless, other parameters can be considered easily with the simulator made available to the SUIT partners. Note that no specified IP packet size or data burst size is assumed within a MAC/PHY frame, and for MAC/PHY encapsulation, procedures will still need to be performed. This is to allow flexibility on choice of packet size, the data throughput, and also to permit studies on efficient packetization schemes. The WIMAX error pattern will be used to study optimal video parameters in WP3.

6 Acronyms

BER	Bit Error Rate
CC	Convolutional Coding
CTC	Convolutional Turbo Coding
CRC	Cyclic Redundancy Check
dB	decibels
FEC	Forward Error Correction
FFT	Fast Fourier Transform
FUSC	Fully Used SubChannelization
IEEE	Institute of Electrical and Electronics Engineers
IFFT	Inverse Fast Fourier Transform
IP	Internet Protocol
LDPC	Low Density Parity Check
MAC	Medium Access Control
MCS	Modulation and Coding
MDC	Multiple Description Coding
OFDM	Orthogonal Frequency Division Multiplexing
OFDMA	Orthogonal Frequency Division Multiple Access
PRBS	Pseudo Random Binary Sequence
PUSC	Partially Used SubChannelization
QAM	Quadrature Amplitude Modulation
QoS	Quality of Service
RTG	Receive/Transmit Transition Gap
SNR	Signal to Noise Ratio
SVC	Scalable Video Coding
SUIT	Scalable, Ultra-fast and Interoperable Interactive Television
TTG	Transmit/Receive Transition Gap
WIMAX	Worldwide Interoperability for Microwave Access
WP	Workpackage

7 References

[1] "All IP Support Requirements," SUIT Deliverable 1.3, document number SUIT_125, 30 September 2006.

[2] "Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems. Amendments 2: Physical and Medium Access Control Layers for Combined Fixed and Mobile Operation in Licensed Bands and Corrigendum 1.," IEEE Std 802.16e-2005, Dec 2005.

[3] "IT++ Signal Processing Library, version 3.10.2," http://itpp.sourceforge.net, May 2006.

[4] "Turbo Coding and Satellite and Wireless Communications," M.Reza Soleymani, Yingzi Gao, and U.Vilaipornsawai, Eds. Kluwer Academic Publishers 2002.

[5] "Fixed and Mobile Channel Models Identifications," WP2.1 SUIT Project Deliverable, July 2006.

[6] IEEE C802.16e-05/006r2, “LDPC Coding for OFDMA PHY”, IEEE 802.16 Broadband Wireless Access Working Group <http://ieee802.org/16>, http://wirelessman.org/tge/index.html, 2005-1-25

[7] B.Baumgartner, M.Reinhardt, G.Richter, and M.Bossert, "Performance of Forward Error Correction for IEEE 802.16e," 10th International OFDM Workshop, Hamburg, Germany, August 2005.

[8] C.Eklund, R.B.Marks, S.Ponnuswamy, K.L.Stanwood, and N.J.M.V.Waes, "WirelessMAN: Inside the IEEE 802.16 Standard for Wireless Metropolitan Networks," IEEE Standards Wireless Networks Series, May 2006.

[9] E.Westman, "Calibration and Evaluation of the Exponential Effective SINR Mapping (EESM) in 802.16," Master's Degree Project Report, Royal Institute of Technology (Kungliga Tekniska Högskolan), Stockholm, Sweden, 2006-09-07.