Performance of Forward Error Correction for IEEE 802.16e

B. Baumgartner†, M. Reinhardt‡, G. Richter†, M. Bossert†

† University of Ulm
Dept. of Telecommunications and Applied Information Theory
Albert-Einstein-Allee 43
D-89081 Ulm, Germany
{bernd.baumgartner, gerd.richter, martin.bossert}@uni-ulm.de

‡ Siemens AG
Lise-Meitner-Straße 13
D-89081 Ulm, Germany
Markus-Reinhardt@siemens.com

***Abstract*— In this paper the broadband wireless access system provided by the IEEE 802.16 wireless MAN air interface with its amendment to mobile users (IEEE 802.16e) is addressed. We exclusively consider data transmission in the uplink based on scalable OFDMA. We provide performance results for the most important forward error correcting (FEC) schemes intended for IEEE 802.16e, namely convolutional codes (CC), convolutional turbo codes (CTC) and low-density parity-check (LDPC) codes.**

***Index Terms*— low-density parity-check codes, convolutional turbo codes, convolutional codes, IEEE 802.16e**

I. Introduction

THE IEEE 802.16 telecommunications standard [1] envisions broadband wireless access technology as a means of providing wireless ”last mile” broadband access in a metropolitan area network (MAN). The performance and services should be comparable or better than traditional DSL, cable or T1/E1 leased line services. Especially in areas beyond the reach of DSL and cable, IEEE 802.16 could offer a cost-effective broadband access solution. The term WiMax (worldwide interoperability for microwave access) has become synonymous with IEEE 802.16, promoting and certifying compatibility and interoperability of broadband wireless products. In its original release 802.16 focused on line-of-sight (LOS) applications in the licensed 10 to 66 GHz frequency range based on single carrier (SC) transmission (WirelessMAN-SC). In a first amendment non-line-of-sight (NLOS) applications in licensed and unlicensed bands in the 2 to 11 GHz frequency range were covered (WirelessMAN-SCa). To meet the requirements of a low cost solution in a multipath environment, orthogonal frequency division multiplexing (OFDM) was chosen as physical layer transmission technique (WirelessMAN-OFDM). To deliver optimum broadband wireless access performance, the concept of scalable OFDMA (orthogonal frequency division multiple access) was adopted. The architecture is based on a scalable subchannel bandwidth using a variable sized FFT according to the channel bandwidth. Within task group E (IEEE 802.16e, [2]) there is an ongoing evolution of IEEE 802.16 addressing mobile applications thus enabling broadband access directly to portable devices like smartphones, PDAs, notebooks and laptop computers.

Our investigations are focused on the uplink of the WirelessMAN-OFDMA physical layer of IEEE 802.16 together with its amendments for mobile applications addressed in IEEE 802.16e. For the purpose of forward error correction (FEC) within the IEEE 802.16 WirelessMAN-OFDMA standard, there is a mandatory convolutional code (CC) and an optional block turbo code (BTC) and convolutional turbo code (CTC). In the amendment for mobility (IEEE 802.16e) a low-density parity-check code (LDPC) was added.

In this paper we provide performance results for three out of the four coding schemes, namely CC, CTC and LDPC codes, BTC codes are not considered here. The paper is organized as follows: In section II, we will provide a brief overview of the basic WirelessMAN-OFDMA physical layer for the uplink. A detailed description of the FEC schemes is presented in Section III. Finally, we provide simulation results for the different FEC schemes in Section IV.

II. System Overview

We consider uplink transmission using the WirelessMAN-OFDMA physical layer specified in the IEEE 802.16 standard [1]. The assumed OFDM

Primitive Param.		Derived Param.	
BW	20 MHz	$F_s = \lfloor \frac{n \cdot BW}{8000} \rfloor \cdot 8000$	22.856 MHz
N_{FFT}	2048	$\Delta f = F_s / N_{FFT}$	11.16 kHz
N_{used}	1681	$T_b = 1/\Delta f$	89.6 μs
n	8/7	$T_g = G \cdot T_b$	11.2 μs
G	1/8	$T_s = T_b + T_g$	100.8 μs

TABLE I
SYSTEM PARAMETERS

parameters are listed in Table I. The nominal bandwidth BW is assumed to be 20 MHz. Applying a sampling factor of $n = 8/7$, yields a sampling frequency $F_s = 22.856$ MHz. Denoting the useful symbol time by T_b and the length of the cyclic prefix by T_g, the fraction of $G = T_g/T_b$ was assumed to be $G = 1/8$.
A block diagram of the physical layer is depicted in Fig. 1. The binary data after randomization is

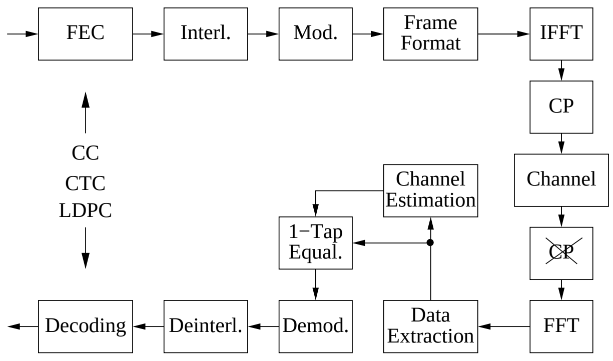

Fig. 1. Overview WirelessMAN-OFDMA uplink

fed into the FEC encoder. We exclusively consider convolutional codes (CC), convolutional turbo codes (CTC) and low-density parity-check (LDPC) codes, described in detail in Section III. After bitwise interleaving the bits are fed into the modulator. Mapping the bits to either QPSK, 16QAM or 64QAM symbols. A subsequent *Frame Formatter* is responsible for the assignment of the modulated symbols to subcarriers. According to the IEEE 802.16 WirelessMAN-OFDMA standard, a special subcarrier allocation pattern is used to account for the specialties dealing with an OFDMA uplink. The standard provides the so-called *PUSC* (partial usage of subchannels) subchannel allocation method. The smallest entity for allocation is a so-called *subchannel*. Each subchannel is made up of six so-called *tiles*, whereby a tile is made of four adjoint subcarriers lasting over three OFDM symbols. In other words, a tile is a 4×3 subgroup in the frequency-time grid. All modulation symbols within a tile are data symbols except the four symbols at the corners which are pilot-symbols used for channel estimation [3]. Subsequently the OFDM signal in time domain is computed via the IFFT. Finally, the cyclic prefix is added.
The channel is assumed to be a time-variant multipath channel modeling mobile users in an NLOS scenario. The receiver noise is modeled by an additive white Gaussian noise (AWGN) process added to the received signal.
Assuming perfect synchronization, the receiver extracts the useful symbol time and therefore removes the cyclic prefix. After the computation of the frequency domain signal via the FFT, the receiver extracts the user specific information (*Data Extraction*) and feds it to the channel estimator and the 1-tap equalizer. With the assumption that the delay spread of the channel is smaller than the cyclic prefix and the time variance of the channel during one OFDM symbol is negligible, the received symbols in frequency domain R_i are given by

$$R_i = H_i \cdot X_i + N_i, \quad i = 0, \ldots, N_{used} - 1.$$

Here, X_i is the transmitted symbol, H_i is the complex valued sample of the channel transfer function and N_i is the complex valued noise sample in subcarrier i. The channel estimator computes $\hat{H}_i$ which are estimates of the real channel factor H_i. The topic of channel estimation for IEEE 802.16 is addressed in [3]. In the following we assume perfect knowledge of the channel transfer function and therefore also perfect channel estimation ($\hat{H}_i = H_i$). The 1-tap (zero forcing) equalizer computes

$$\tilde{R}_i = \frac{R_i}{H_i} = X_i + \frac{N_i}{H_i}.$$

These equalized symbols are fed into the soft output demodulator computing log-like ratios (LLRs) for bits. After deinterleaving the LLRs are fed into the FEC decoders using this *soft input* for decoding.

III. FORWARD ERROR CORRECTION

In the following we describe the FEC encoders for CC, CTC and LDPC codes defined in the WirelessMAN-OFDMA parts of [1],[2] and address the applied decoding algorithms.

A. Convolutional Code (CC)

In the WirelessMAN-OFDMA part, the CC is the only mandatory coding scheme, all the others like CTC and LDPC are optional. The CC mother code is the rate $R = 1/2$, memory $m = 6$ convolutional encoder with generator matrix $G(D) = (1 + D + D^2 + D^3 + D^6 \quad 1 + D^2 + D^3 + D^5 + D^6) =$

$(171 \quad 133)_{oct}$. The codewords are generated using the tailbiting method. This means, that the encoder starts in the same state as it ends up after encoding. For non-recursive encoders (like here), this is quite easy. The encoder shift register has to be initialized with the last part of the information sequence since this will be the state where the encoder ends up after encoding of this information sequence. For decoding we use the tailbiting BCJR algorithm described in [4], where the forward and backward recursion wrap around the trellis for some number of segments called the *wrap depth*.

In an optional encoding method, the CC can also be encoded using termination or zero-tailing (ZT-CC), where the encoder is forced to return to the all-zero state feeding a sufficient number of zeros into the encoder.

B. Convolutional Turbo Code (CTC)

The CTC is a parallel concatenated convolutional code or turbo code. The concept of turbo coding was introduced by Berrou [5]. An overview of the CTC encoder is depicted in Fig. 2. It consists of a CTC

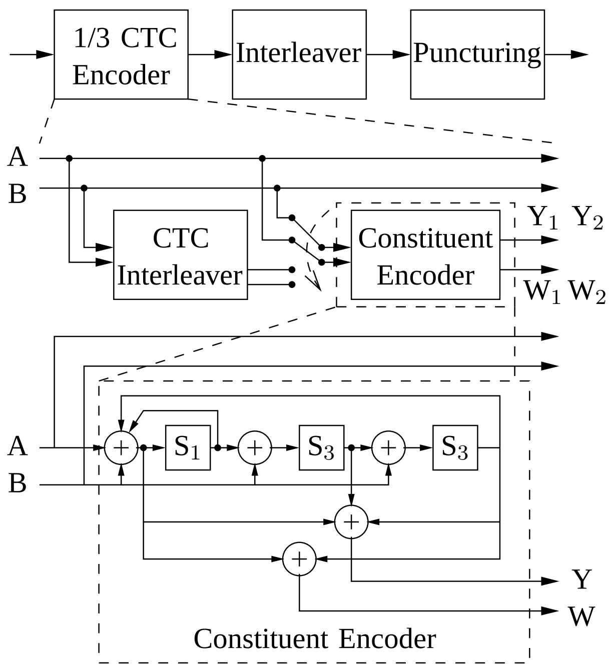

Fig. 2. Overview CTC Encoder

encoder with a natural rate of $R = 1/3$ followed by an additional interleaver and a final puncturing to obtain the desired rate. The CTC encoder consists of two identical constituent encoders separated by an CTC internal interleaver. The constituent encoder has a natural rate of $2/4$ and its minimal realization is depicted in the lower part of Fig. 2. It consists of $m = 3$ memory elements and consumes two input bits per time instance and produces four output bits. Furthermore, the encoder is recursive and systematic. The CTC encoding procedure is as follows: The two information bits A and B are fed directly to the output and in a first encoding step additionally into the constituent encoder, producing the parity bits Y_1 and W_1. Afterwards, in a second encoding step, the interleaved information symbols (A and B) are again fed into the constituent encoder, now producing the parity bits Y_2 and W_2. This means the info/code tuple of the CTC encoder is $AB/ABY_1W_1Y_2W_2$ and its natural rate is therefore $1/3$. Concerning the details of the CTC internal interleaver, the interleaver after CTC encoding and the puncturing we refer to [1].

Another important point concerning the constituent encoder is that again tailbiting is used to encode the codewords. For recursive encoders, tailbiting is not that easy as it is for non-recursive encoders. To ensure that the starting state is the same as the ending state, which is called circulation state, for recursive encoders an initial encoding of the whole sequence has to be performed. The initial encoding is started in the all-zero state and depending on the information sequence it ends up in a special state, S_{end}. Based on this ending state, the circulation state can be computed using linear algebra methods based on the state space description of the encoder. In order to get rid of this linear algebra approach computations, the IEEE 802.16 provides a so-called circulation state look-up table, where the correspondence between the final state S_{end} of the initial encoding process and the circulation state as a function of the information sequence length is listed. After determining S_{end} with an initial encoding starting in the all-zero state, the circulation state can be looked up in this table provided by the standard. Afterwards, the real encoding can be started, whereby the encoder state is initialized now with the circulation state. Hence, a tailbiting encoder needs two complete encoding processes which adds complexity to the encoder.

Complexity is also added to the decoder of the constituent code. Again we use the tailbiting MAP algorithm described in [4] for symbol-by-symbol a-posteriori decoding of the constituent code. The complexity added to the decoder compared to the case where the starting and ending state is known to the decoder is in the additional wrap around for the forward and backward recursion of the MAP decoder. Since the wrap around length can be kept small, the additional complexity is quite small. The overall CTC is decoded using the well-known iterative decoding process [5] where the two decoders exchange information based on LLRs of the information bits.

C. Low-Density Parity-Check (LDPC) Codes

With the optional irregular LDPC codes, k systematic information bits are encoded to n code bits by adding $r = n - k$ parity check bits. The LDPC codes are defined based on a parity check matrix $\mathbf{H}$ of size $(n - k) \times n$ that is expanded from a binary base matrix $\mathbf{H}_b$ of size $r_b \times n_b$, where $n = z \cdot n_b$ and $r = z \cdot r_b$. The base matrix size n_b is an integer equal to 24 and the expansion factor z is an integer $24 \leq z \leq 96$. From these values we can compute the minimal code length to $n_{min} = 24 \cdot 24 = 576$ bits and the maximum code length to $n_{max} = 24 \cdot 96 = 2304$ bits. In total there are five different base matrices [2], one for rate $1/2$ codes, two different for rate $2/3$ codes and two different for rate $3/4$ codes. The base matrix entries $p(i, j)$ are either '-1' indicating a replacement with a $z \times z$ all-zero matrix or integers $p(i, j) \geq 0$ indicating a replacement with a $z \times z$ permutation matrix. The permutation matrix represents a circular right shift on $p(i, j)$ positions. This means an entry $p(i, j) = 0$ must be replaced by a $z \times z$ identity matrix. For decoding we apply a belief propagation decoder using the sum-product algorithm [6].

IV. Simulation Results

In the following we present some simulation results. The tailbiting CC was decoded using the tailbiting MAP decoder described in [4]. The wrap depth was set to 50 trellis segments and the computation was carried out in the log-domain. The same decoder was used to decode the tailbiting constituent code of the CTC. For the CTC, iterative decoding was stopped after 8 iterations. Concerning the LDPC decoder, the maximum number of iterations of belief propagation decoding was limited to 100. The simulations were carried out for different channels, rates, lengths and modulation schemes. As a reference, we first provide some simulation results for the different FEC schemes in an AWGN channel. In Fig. 3 we depicted the bit error rate (BER) versus the signal-to-noise-ration (SNR). The code length was $N = 576$ bits, the code rate was $R = 1/2$ and the modulation schemes were QPSK as well as 16QAM. It can be seen that the CC performs worst. The performance of the CTC and the LDPC code is quite similar,

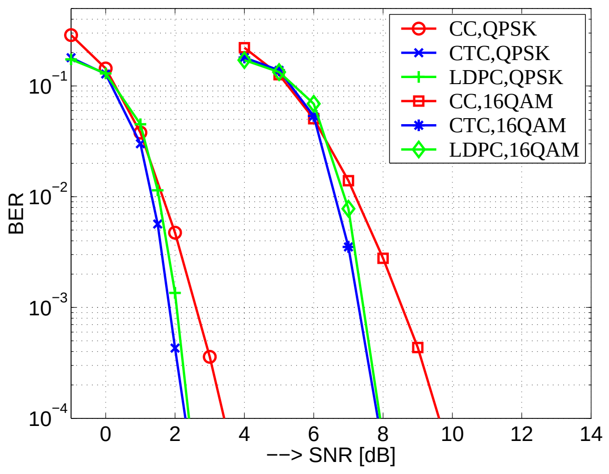

Fig. 3. FEC comparison, BER, AWGN, R=1/2, N=576

whereby there is a 'tenth of a dB advantage' for the CTC. To reach a BER of 10^{-4} the CC needs round 1 dB more for QPSK and round 1.8 dB more for 16QAM compared to CTC and LDPC.
In the following we present simulation results using the afore described WirelessMAN-OFDMA physical layer of IEEE 802.16 in the uplink. We modeled a single user allocating the whole bandwidth. The assumed channel was the well-known TU30 channel, modeling a mobile user in the 'typical urban' environment moving with an relative velocity of 30 km/h to the base station at a carrier frequency of 3.5 GHz. The results for the different FEC schemes are depicted in Fig. 4. The solid lines indicate the

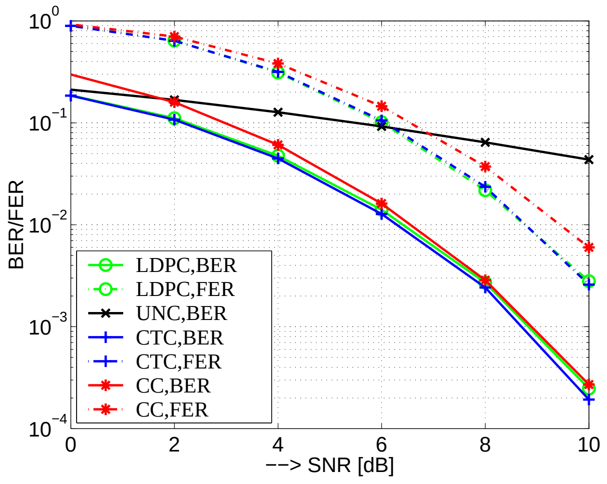

Fig. 4. FEC comparison TU30, QPSK, R=1/2, N=576

BER and the dash-dotted lines the frame error rate (FER). The rates and code lengths are the same as for the AWGN channel. The order in performance is still the same as for the AWGN channel but for this scheme, the disadvantage using the CC reduces, especially for the BER. At high SNR all three FEC schemes lie within 0.2 dB. For low SNR it can be seen that the curves for CTC and LDPC merge with the uncoded curve which is due to their systematic nature, whereas the CC is non-systematic. For a FER of 10^{-1} there is still a 0.5 dB advantage of CTC and LDPC compared to the CC and at a FER of 10^{-2} its an 0.7 dB advantage. In Fig. 5 we depicted the results using a rate of $3/4$ instead of $1/2$. The

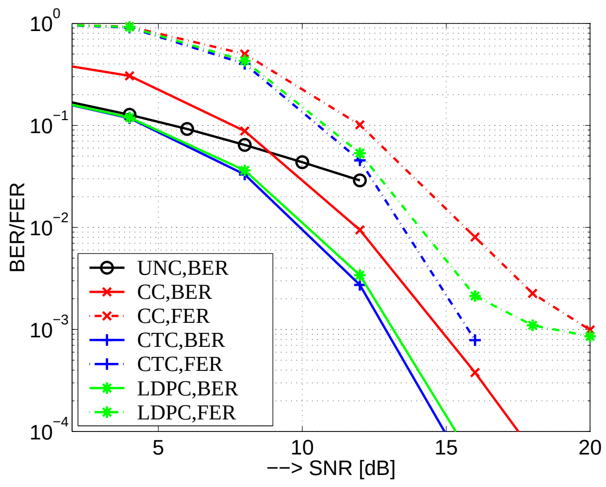

Fig. 5. FEC comparison TU30, QPSK, R=3/4, N=576

difference between CTC and LDPC is still quite small, but the performance gain compared to the CC is quite big. At a FER of $10^{-1}(10^{-2})$ the CTC outperforms the LDPC by about $0.2(0.6)$ dB and the CC by about $1.4(2.1)$ dB. Moreover, the FER curve of the LDPC code shows an error floor at 10^{-3}. Though the used code length of 576 bits seems to be quite short for turbo like codes it is in the mid range for the CTC, where even shorter block lengths has been standardized. As already mentioned, the length of LDPC codes start at 576 code bits and lasts up to 2304 code bits. The three most left hand side curves in Fig.6 show the influence of an increasing code length N for a rate 1/2 LDPC code. The gain for N=1920 compared to N=576 at a BER of 10^{-4} is round 0.8 dB. So there is some gain but not that much. The remaining BER curves in Fig.6 show the performance for different code rates and modulation schemes using the CC with rate 1/2 and N=576. To double the bandwidth efficiency from rate 1/2 QPSK to 1/2 16QAM is at an additional 7 dB for a BER of 10^{-4}. Moreover, it is approximately the same loss in performance using the rate 3/4 QPSK scheme with a bit less bandwidth efficiency compared to the rate 1/2 16QAM scheme.

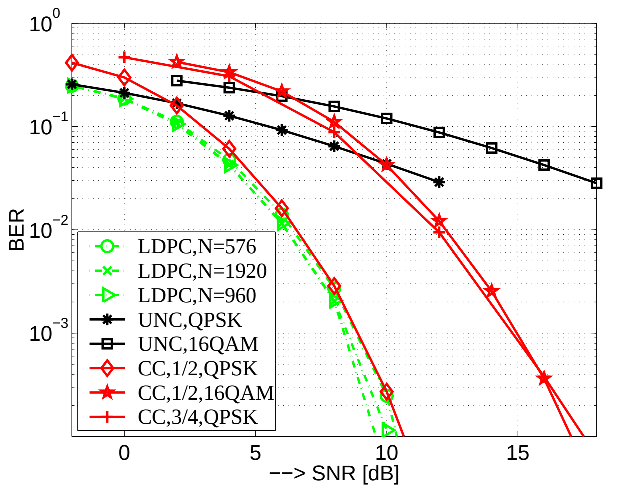

Fig. 6. TU30, LDPC with different code lengths N and comparison of different rate/modulation schemes for CC, N=576

V. Conclusions

The performance gain using advanced coding techniques like CTC and LDPC compared to the CC is quite small for rate 1/2 codes. One reason for this is that the standard only provides short to moderate code lengths ($N <= 2304$) which is the most crucial parameter for this class of codes. Only for higher code rates like 3/4, CTC and LDPC result in a noticeable performance gain compared to the mandatory CC. The performance of CTC and LDPC is about the same and by changing some decoding parameters the small advantage of one of them can be interchanged. Nevertheless, LDPC decoding is less complex than CTC decoding.

References

[1] IEEE Std 802.16-2004, *Standard for Local and metropolitan area networks, Part 16: Air Interface for Fixed Broadband Wireless Access Systems*, October 2004.

[2] IEEE P802.16e/D7, *Draft IEEE Standard for Local and metropolitan area networks, Part 16: Air Interface for Fixed Broadband Wireless Access Systems, Amendment for Physical and Medium Access Control Layers for Combined Fixed and Mobile Operation in Licensed Bands*, April 2005.

[3] B. Baumgartner, M. Mayrock, M. Reinhardt, and M. Bossert, *Performance of Channel Estimation for IEEE 802.16e*, accepted for publication in Proceedings 10th International OFDM-Workshop 2005, Hamburg, Germany.

[4] John B. Anderson, and Stephen M. Hladik, *Tailbiting MAP Decoders*, IEEE Journal on Selected Areas in Communications, vol.16, no.2, pp. 297-302, February 1998.

[5] C. Berrou, A. Glavieux, and P. Thitimasjshima, *Near Shannon Limit Error-Correcting Coding and Decoding: Turbo-Codes*, Proceedings of the International Conference on Communication, pp.1064-1070, May 1993, Geneva, Switzerland.

[6] David J. C. MacKay, *Good Error-Correcting Codes Based on Very Sparse Matrices*, IEEE Transactions on Information Theory, vol.45, no.2, pp. 399-431, March 1999.