

Practical Evaluation of Opportunistic Error Correction

Xiaoying Shao, Cornelis H. Slump

x.shao@ewi.utwente.nl, c.h.slump@ewi.utwente.nl

University of Twente, Faculty of EEMCS, Signals and Systems Group

Abstract—In [1], we have proposed a novel cross-layer scheme based on resolution adaptive ADCs and fountain codes for the OFDM systems to lower the power consumption in ADCs. The simulation results show that it saves more than 70% power consumption in ADCs comparing to the current IEEE 802.11a system. In this paper, we investigate its performance in the real-world. Measurement results show that the FEC layer used in the IEEE 802.11a system consumes around 59 times of the amount of power in ADCs comparing to the LDPC codes from the IEEE 802.11n standard, whose power consumption in ADCs is around 26 times of the proposed cross-layer method. In addition, this new cross-layer approach only needs to process the well-received packets to save the processing power. The latter can not be applied in the current FEC schemes.

I. INTRODUCTION

Orthogonal Frequency Division Multiplexing (OFDM) has become a popular scheme for recent WLAN standards which operate at a high data rate [2]. OFDM has a high *Peak-to-Average Power Ratio* (PAPR), therefore it requires *Analog-to-Digital Converters* (ADC) with a high dynamic range. These high resolution ADCs can take up to 50% of the base-band power [3]. However, low power-consumption in battery-powered wireless receivers is a highly desirable feature.

In [1], we have proposed a novel error correction layer based on *adaptive ADCs* and *fountain codes* to mitigate the effects of a wireless channel at a lower power consumption level in ADCs comparing to traditional solutions. With this method, the resolution of ADCs is adapted to each channel condition instead of fixing to the high-resolution for the worst-case scenario. As a result, the power consumption of the ADC is reduced under most, i.e. non worst-case, channel conditions.

A further resolution reduction of the ADC can be achieved by using a novel opportunistic error correction scheme that integrates into the physical layer. This approach allows us to discard some part of the channel with deep fading. The current WLAN standards do not support this idea, as all sub-bands are considered equally important by the *Forward Error Correction* (FEC) layer. However, the opportunistic error correction method based on fountain codes does not have this disadvantage.

By using fountain codes, the receiver can recover the original data by collecting enough fountain-encoded packets. It does not matter which packets are received, only a minimum amount of packets have to be received correctly [4]. In other words, fountain-encoded packets are independent with respect to each other. By transmitting a fountain-encoded packet over

a sub-band of a channel. Thus, multiple packets are transmitted simultaneously, using frequency division multiplexing. The receiver discards fountain-encoded packets which are transmitted over the sub-band with deep fading. Correspondingly, the power consumption in ADCs decreases.

The performance of this new scheme has been investigated by the C++ simulation in [1]. With the same effective throughput, the simulation results have shown that this new algorithm allows a reduction of more than 70% power consumption in ADCs comparing to the traditional IEEE 802.11a system [1].

C++ simulation, with its highly accurate double-precision numerical environment, is on the one hand a perfect tool for the investigation of the algorithms. On the other hand, many imperfections of the real-world are neglected (e.g. the quantization noise is assumed to be dominant and the channel noise is ignored in [1]). So, simulation may show a too optimistic receiver performance. The uncertainties in the real life are mainly simplified assumptions in the simulation like perfectly known noise levels, additive Gaussian noise, omitted synchronization, etc. Therefore, we want to evaluate the performance of the opportunistic error correction scheme in the real-world.

In this paper we test an approach based on resolution adaptive ADCs and fountain codes to reduce the power consumption of ADCs in the experimental communication testbed built by the Signals and Systems Group, University of Twente. Since the resolution adaptive ADC is not a on-the-shelf product yet, we requantize the signal from the ADCs by software to mimic the effect of resolution adaptive ADCs. In addition, we test how the imperfect synchronization affects the opportunistic error correction scheme based on fountain codes.

The outline of this paper is as follows. Opportunistic error correction layer is applied to lower the power consumption in wireless OFDM-based receivers. First, the basic idea of the opportunistic error correction scheme is described, which is followed by the system setup. In Section IV, the measurement setup is depicted. We compare the FEC layer from the IEEE 802.11a standard [5] and the one from the IEEE 802.11n standard [6] with the opportunistic error correction layer in the measurements. Finally, the measurement results are analyzed. The paper ends with a discussion of the results.

II. OPPORTUNISTIC ERROR CORRECTION

Opportunistic error correction is based on fountain codes. In this paper, we use a kind of fountain codes, i.e. *Luby Transform* (LT) codes [7] in the proposed error correction layer. Other fountain codes (e.g. Raptor codes [8]) can also be applied.

Consider a block of size K packets s_1, s_2, \dots, s_K to be encoded by a fountain code. A packet has m bits and considered a unit. At each clock cycle, labeled by n , one fountain-encoded packet is generated by selecting a set of source packets randomly and computing the bitwise sum (XOR) of these source packets [4]. The fountain codes can supply an unlimited number of encoded packets based on s_1, s_2, \dots, s_K . In practical systems, only a fixed number of packets N is generated.

At the receiver side, enough packets are required for successful decoding. The required number of received packets N is slightly larger than the number of source packets K and is defined as:

$$N = K(1 + \varepsilon) \quad (1)$$

where ε is the percentage of extra packets and is called the overhead.

After receiving N packets, the receiver can recover the source packets by the *message-passing* algorithm [9] which has a linear decoding cost. We have shown that decoding the fountain codes using the message-passing algorithm combined with Gaussian elimination allows small block sizes, e.g. $K = 500$, with small overhead $\varepsilon = 3\%$ in [1]. Small block sizes are needed to keep the decoding delay low, which is important in real-time applications such as WLAN applications.

Fountain codes are designed for *Erasur Channels*. However, wireless channels are noisy fading channels, not erasure channels. In practical systems, fountain codes are used in combination with other error correction algorithms to convert the noisy channels into erasure channels, often *Low-Density Parity-Check* (LDPC) codes [9]. In this paper, LDPC codes are used together with a *Cyclic Redundancy Check* (CRC) to make the wireless channel behave like an erasure channel.

Our FEC encoding scheme is performed in the following order. First, a fountain-encoded packet is created. Then, the CRC is added. Finally, the packet is encoded by the LDPC code.

At the receiver, each fountain-encoded packet is first LDPC decoded if its energy is above or equal to a threshold (i.e. corresponding to $\text{BER} \leq 10^{-5}$). The threshold is determined by the channel energy, the channel noise and the quantization noise. So, the resolution of ADCs (i.e. the quantization noise) can affect the threshold in a given channel. If the system allows packets from the low-energy channel to be lost, the required resolution of ADCs can be reduced.

The received packet is discarded if its energy is below the threshold. If the LDPC decoding fails, the received packet is discarded as well. If the LDPC decoding succeeds, the CRC is used to identify any errors undetected by the LDPC codes. If the CRC decoder detects an error, the receiver assumes that the

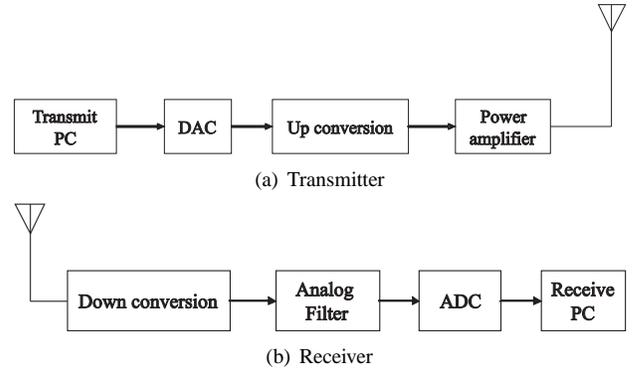


Fig. 1. Block diagram of the testbed

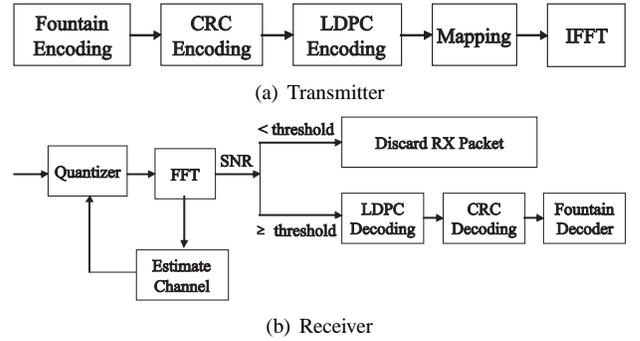


Fig. 2. System model of opportunistic error correction layer: transmitter (top) and receiver (bottom).

whole packet has been lost. Once the receiver gets N surviving fountain-encoded packets, it starts to recover the source data.

III. SYSTEM SETUP

The opportunistic error correction layer is based on resolution adaptive ADCs and fountain codes. This proposed cross layer can be applied in the OFDM system. In this paper, we evaluate its performance in the testbed, as shown in Fig. 1. It is assembled as a cascade of the following modules: PC, DAC, RF up-converter, power amplifier, antenna, and the reverse chain for the receiver. In the receiver, there is no power amplifier and band-pass RF filter before the down-converter but a low-pass baseband filter before the AD converter to remove the aliasing.

A. The Transmitter

The data is generated offline in C++. The generation consists of the random source bits selection, the FEC encoding and digital modulation. Any FEC scheme and digital modulation can be applied. The FEC layer in the current IEEE 802.11a system is based on *Rate Compatible Punctured Codes* (RCPC). RCPC has a good performance for random bit errors. An interleaver is used to mitigate the burst errors. Although this solution works well in practical systems, it is not optimal. First, because packets that have encountered a low-energy channel are still processed by the decoders. It will waste processing power. In addition, the error correction layer is based on worst case scenarios. This means that for most

packets, the code rate and hence capacity can be increased. Furthermore, the resolution of the applied ADCs is fixed for IEEE 802.11a receivers, which is designed for worst case conditions. However, worst case condition does not always happen.

In Fig. 2(a), the opportunistic error correction scheme is depicted to reduce the power consumption in ADCs. The key idea is to generate additional packets by the fountain encoder. First, a block of source bits are divided into a set of packets and encoded by the fountain encoder. Then, a CRC checksum is added to each fountain-encoded packet, and LDPC encoding is applied. On each sub-carrier, a fountain-encoded packet is transmitted. Thus, multiple packets are transmitted simultaneously, using frequency division multiplexing.

The generated data is stored in a file. A server software in the transmit PC uploads the file to the Adlink PCI-7300A board¹ which transmits the data to DAC (AD9761)² via the FPGA board. After the DAC, the baseband analog signal is up-converted to 2.3 GHz by a Quadrature Modulator (AD8346)³ and transmitted using a conical skirt monopole antenna.

B. The Receiver

The reverse process takes place in the receiver. The received RF signal is first downconverted by a Quadrature Demodulator (AD8347)⁴, then pass the 8-th order low-pass Butterworth analog filter to remove the aliasing. The baseband analog signal is quantized by the ADC (AD9238)⁵ and stored in the receive PC via the Adlink PCI board.

The received data is processed offline in C++. The receiver should synchronize with the transmitter and estimate the channel using the preambles and the pilots, which are defined in [5]. Timing and frequency synchronization is done by the Schmidl & Cox algorithm [10] and the channel is estimated by the *zero forcing* algorithm. With the estimated channel information, the resolution of the adaptive ADC can be reduced to the minimum for each channel realization. In addition, the *residual carrier frequency offset* is estimated by the four pilots in each OFDM symbol.

After the synchronization and the channel estimation, decoding can start. Fig. 2(b) depicts the opportunistic error correction decoding scheme. With the estimated channel knowledge, the SNR of each sub-carrier can be derived. If the SNR of the sub-carrier is equal to or above the threshold, the received fountain-encoded packet will go through LDPC decoding, otherwise it will be discarded. This means that the receiver is allowed to discard low-energy sub-carrier (i.e. packets) to lower the dynamic range of the ADC and hence the receiver is allowed to discard the erroneous packets. As only packets with a high SNR are processed by the receiver, this will not happen often. When the receiver collects enough fountain-encoded packets, it starts to recover the source data.

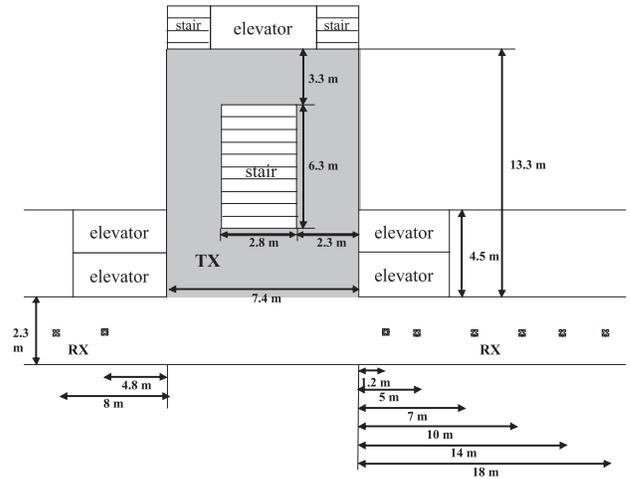


Fig. 3. Measurement Setup: antennas are 0.9 m above the concrete floor. The measurements are done in the corridor of the Signals and Systems Group. The receiver is positioned at left/right side of the corridor (i.e. the cross positions) and the transmitter is at the gray part as shown in the figure. The room contains one coffee machine, one garbage bin and one glass cabin.

IV. MEASUREMENTS

Measurements are carried out in the corridor of the Signal and Systems Group, located at the 9th floor of Building Hogeekamp in University of Twente, the Netherlands. The measurement setup is shown in Fig. 3. The transmitter (TX) was positioned in an open place in front of the elevator (i.e. the gray area in Fig. 3), while the receiver antenna (RX) was in the left/right side of the corridor (i.e. the cross positions in Fig. 3). The transmit antenna was moved arbitrarily in the gray area of Fig. 3. 56 measurements were done in this scenario with a non-line-of-sight situation. The average transmitting power is around -38 dBm and the distance between the transmitter and the receiver is around 7 ~ 25 meters. The measurements were conducted at 2.3 GHz carrier frequency and 20 MHz bandwidth.

In order to investigate whether the opportunistic error correction layer performs better than the other FEC layers in the real-world, the following FEC layers are compared in the measurements:

- FEC I: convolutional codes ($R = \frac{1}{2}$) with interleaving defined in the IEEE 802.11a standard.
- FEC II: the (324,648) LDPC code from the IEEE 802.11n standard.
- FEC III: the opportunistic error correction layer based on fountain codes.

In [1], these FEC schemes have been compared in the C++ simulation. In the simulation, they can be compared by using the same source bits. Different channel bits can go through the same random frequency selective channels. However, it does not apply in the real environment. The wireless channel is time-variant even when the transmitter and the receiver are stationary (e.g. the moving of elevator with the closed door can affect the channel). Hence, we should compare them by using the same channel bits.

¹ADLINK, 80 MB/s High-Speed 32-CH Digital I/O PCI Card

²Analog Devices, 10-Bit, 40 MSPS, dual Transmit D/A Converter.

³Analog Devices, 2.5 GHz Direct Conversion Quadrature Modulator.

⁴Analog Devices, 800 MHz to 2.7 GHz RF/IF Quadrature Demodulator

⁵Analog Devices, Dual 12-Bit, 20/40/65 MSPS, 3V A/D Converter.

Because not every stream of random bits is a codeword of a certain coding scheme, it is not possible to derive its corresponding source bits from any sequence of random bits, especially for the case of FEC I and FEC III. Fortunately, the decoding of FEC II is based on the parity check matrix. Any stream of random bits can have its unique sequence of source bits with its corresponding syndrome matrix. The receiver can decode the received data based both on the parity check matrix and the syndrome matrix. So, FEC I can use the same channel bits with FEC II, same for FEC II and FEC III. In such case, they can be compared under the same channel condition (i.e. channel fading, channel noise and the distortion caused by the hardware.). During the measurements, both sequences of channel bits are transmitted in one burst (i.e. 2 blocks) in order to have their channels as similar as possible.

In the measurements, we transmit more than 300 blocks of source packets over the air. Each block consists of 88200 source bits. The source bits are encoded by FEC I and III, respectively. The encoded bits are shared with FEC II as just explained. Afterwards, they are mapped into QAM-16 symbols before the OFDM modulation.

FEC I, II and III are compared with the same effective throughput (i.e. 10% packet loss). One packet is 54 Bytes⁶, so 10% packet loss is equivalent to a BER of 2.3×10^{-4} . For FEC I and II, the lost packets will be retransmitted. For FEC III, fountain codes replaces the retransmission protocol. With FEC III, each burst is encoded by a LT code (with parameters $c = 0.03$, $\sigma = 0.3$) and decoded by the message-passing algorithm and Gaussian elimination together. From [1], we know that 3% overhead is required to recover the source packets successfully. To each fountain-encoded packet, a 7-bit CRC is added before the (175,255) LDPC encoder is applied. Since the effective code rate in our measurements is $0.5 \times 0.9 = 0.45$, we can lose around 30%⁷ of the total transmitted packets in FEC III.

At present, there is no feedback channel in our testbed. No retransmission can occur, so we use the corresponding BER value for the 10% packet loss in FEC I and II. In addition, the modulation scheme is fixed to QAM-16 in our measurements. Each measurement corresponds to the fixed position of the transmitter and the receiver. It is possible that some measurements might fail in decoding. If the received data per measurement has a BER lower than 10^{-3} in FEC I and II, we assume that the measurement succeeds otherwise we assume it fails. For the case of FEC III, if the packet loss is more than 30% as expected, we assume that the measurement fails.

V. RESULTS

In total, 56 measurements have been done. There are 6 blocks of data transmitted over each measurement: 3 blocks for FEC I and II and 3 blocks for FEC II and III. Not every

⁶a common used value.

⁷ $30\% \approx 1 - \frac{R}{R_1 \times R_2}$, where R is the effective code rate (i.e. 0.45), R_1 is the code rate of LT codes (i.e. $\frac{1}{1.03} \approx 0.97$) and R_2 is the code rate of the (175,255) LDPC code with 7-bit CRC (i.e. $\frac{168}{255} \approx 0.66$)

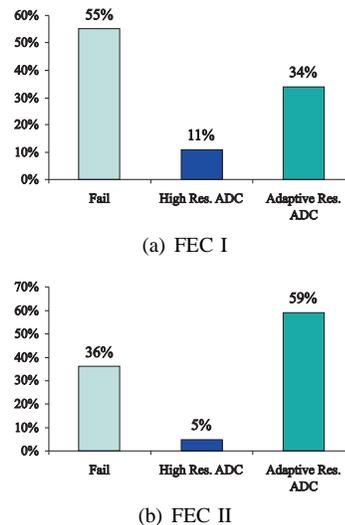


Fig. 4. Statistical analysis of the measurement data for FEC I (top) and FEC II (bottom). By using FEC I, 55% of measurements are failed, 11% of measurements require high-resolution ADCs and 34% of measurements can use resolution adaptive ADCs. For the case of FEC II, 36% of measurements are failed, 5% of measurements need high-resolution ADCs and 59% of measurements can use resolution adaptive ADCs.

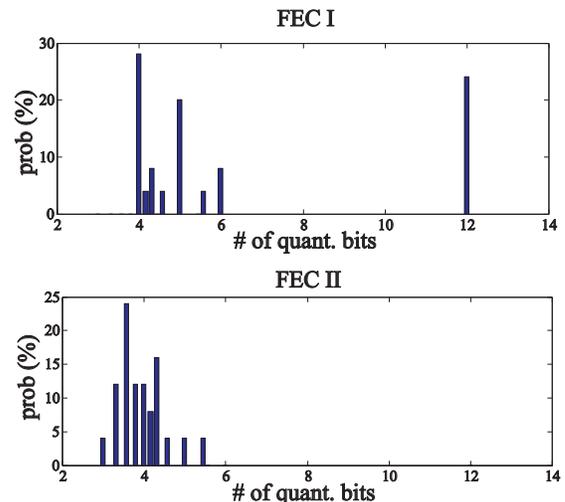


Fig. 5. Comparison in the number of quantization bits between FEC I (top) and FEC II (bottom). Both FEC schemes succeed in 45% of measurements. In those measurements, high-resolution ADCs are only required by FEC I.

measurement succeeds in decoding. With FEC III, 68% of measurements succeed which is almost the same as FEC II (i.e. 66%). FEC I only has 45% of successful measurements and performs the worst comparing to FEC II and FEC III.

As mentioned earlier, the wireless channel is time-variant even when the transmitter and the receiver are placed at the same position. So, we are going to analyze the measurements for FEC I and II and the ones for FEC II and III separately.

A. FEC I vs. FEC II

From Fig. 4, we can see that more measurements succeed in FEC II (i.e. 64%) than in FEC I (i.e. 45%). With FEC I, 11% of the measurements need high-resolution ADCs. For the

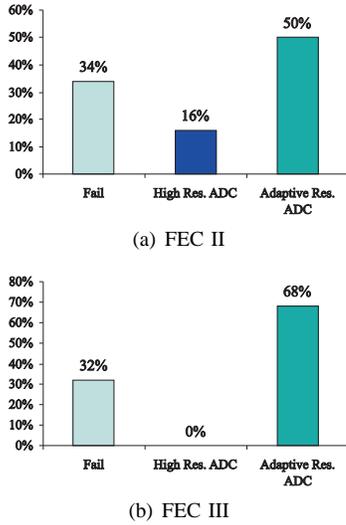


Fig. 6. Statistical analysis of the measurement data for FEC II (top) and FEC III (bottom). By using FEC II, 34% of measurements are failed, 16% of measurements require high-resolution ADCs and 50% of measurements can use resolution adaptive ADCs. For the case of FEC III, 32% of measurements are failed, no measurements need high-resolution ADCs and 68% of measurements can use resolution adaptive ADCs.

case of FEC II, only 5% of the measurements require high-resolution ADCs.

The successful measurements in FEC I also succeed in FEC II. In those 45% measurements, the received data from ADC is first checked in the software whether they can be requantized by a lower resolution ADC comparing to the ADC in the testbed. If it allows, it will be requantized by a resolution adaptive ADC then decoded by FEC I and FEC II, respectively. With the minimum resolution for each case, the average BER is 3.07×10^{-4} for FEC I and 1.24×10^{-4} for FEC II.

Fig. 5 shows that FEC I and II have different requirement for the minimum resolution of ADCs. In those 45% measurements, FEC II does not require high-resolution ADCs which are needed by FEC I. For a CMOS-integrated ADC, the power consumption scales linearly with the number of quantization levels [3]. On average, FEC I demands 1004 quantization levels but FEC II only needs 17 levels. Correspondingly, the power consumption in ADCs by using FEC I is around 59 times of that by using FEC II.

B. FEC II vs. FEC III

Fig. 6 is the statistical analysis of the measurement data shared by FEC II and III. From this figure, we can see that FEC III has slightly more successful measurements (i.e. 68%) comparing to FEC II (i.e. 66%). In addition, the measurement data tells that high resolution ADCs are not necessary for FEC III once the measurement succeeds. However, it does not happen to FEC II. Around 25% of the successful measurements requires high-resolution ADCs (i.e. 12-bit ADC) by using FEC II.

Both FEC schemes only succeed in around 57% of the measurements. For those 57% successful measurements, the received digital data is first checked whether they can be

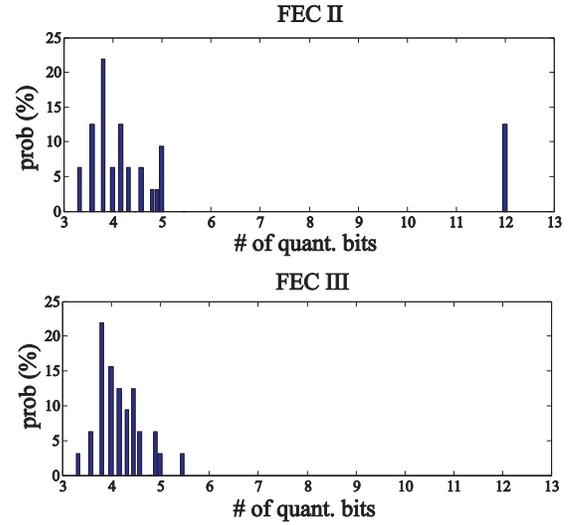


Fig. 7. Comparison in the number of quantization bits between FEC II (top) and FEC III (bottom). Both FEC schemes succeed in 57% of measurements. In those measurements, high-resolution ADCs are only required by FEC II.

requantized by a lower resolution ADC comparing to the one in the testbed. If possible, the received data will go through resolution adaptive ADCs in the C++ simulation and the lowest number of quantization bits will be found for each measurement data. With the minimum resolution for each scenario, FEC II has an average BER of 2.4×10^{-4} and FEC III is error free.

Fig. 7 shows the required number of quantization bits for FEC II and III in those 57% successful measurements. Around 12.5% of those successful measurements, high-resolution ADCs are needed in FEC II. The average number of quantization levels for FEC III is around 3.8% (i.e. 20 levels) of the case for FEC II (i.e. 528 levels). In other words, using FEC II consumes 26 times of power in ADCs comparing to FEC III.

From the measurement results, we find that FEC III works better than FEC II, especially in a channel with deep fading (i.e. the dynamic range of the channel is larger than 10 dB). The measurement results show that FEC II either needs high-resolution ADCs or fails in the channel with deep fading. Although FEC III might also fail in those deep fading channels, the measurement data shows that it performs more efficient than FEC II.

Let us take one measurement as an example. In this measurement, both FEC II and III fail. Fig. 8 shows the estimated channel information of this measurement. As we can see, it has a dynamic range of more than 40 dB. With FEC II, around 89% of the encoded packets can not be decoded. With FEC III, about 65% of fountain-encoded packets are lost during the transmission. As mentioned earlier, we only allow 30% of fountain-encoded packets to be lost in order to recover the source data by fountain codes. So, FEC III also fails.

However, if we want to have reliable communication in such a channel without changing the mapping scheme and the code rate, we can retransmit the lost packets for FEC II and transmit

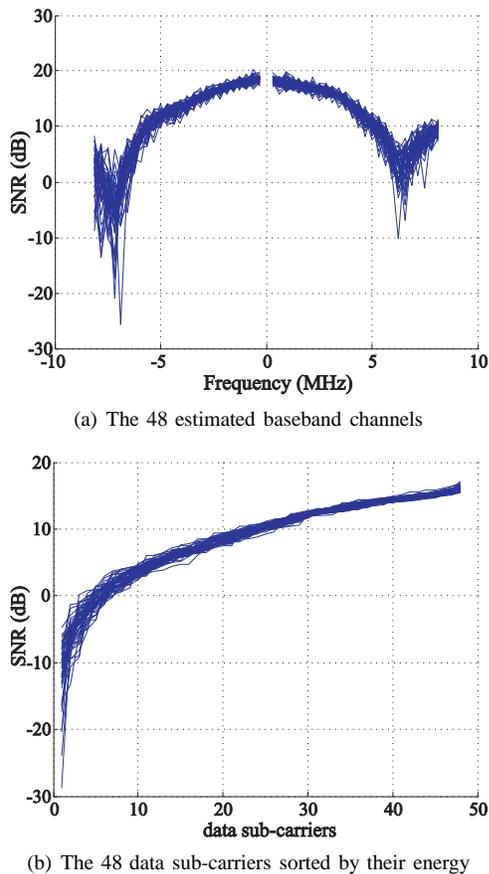


Fig. 8. The estimated baseband channels of a measurement, which is an example that FEC III performs more efficient than FEC II in the channel with deep fading. The channel is estimated every 64 OFDM symbols. One measurement consists of 3 blocks of data and each block consists of 1024 OFDM symbols. So, there are 48 estimated channels in one measurement. No data is transmitted at the DC frequency.

more fountain-encoded packets to the receiver for the case of FEC III once the required feedback channel is implemented. In such case, around 9 blocks of data need to be transmitted by using FEC II in order to receive 1 block of data correctly, but with FEC III only 2 blocks of fountain-encoded packets should be sent.

In addition, a packet encoded by FEC II is transmitted over all the sub-carriers. It can not be predicted whether the received packet is decodable with the estimated channel knowledge. So, all the received packets should be processed. However, it is not the case in FEC III. By using FEC III, each fountain-encoded packet is transmitted over one sub-carrier. Each sub-carrier can be modeled as a flat fading channel [1]. For the used LDPC code to convert the wireless channel into an erasure channel, it has a BER of 10^{-5} or lower when $\text{SNR} \geq 12\text{dB}$. From Fig. 8(b), we can see that around $\frac{2}{3}$ data sub-carriers have SNR lower than 12 dB. That explains why we lose around 65% fountain-encoded packets during the transmission. Also, it means that we can discard the fountain-encoded packets whose SNR is lower than the threshold. Therefore, FEC III consumes less processing power than FEC

II.

VI. CONCLUSIONS

In this paper, we have compared the FEC layer from the IEEE 802.11a standard (i.e. FEC I) and the FEC layer from the IEEE 802.11n standard (i.e. FEC II) with the opportunistic error correction scheme (i.e. FEC III) in the practical OFDM-based system. The real wireless channel is time-variant, so FEC I and II share the same channel bits to be compared in the same channel condition. Same for FEC II and FEC III. In 56 measurements, FEC III has the most successful measurements comparing to FEC I and II. Among those successful measurements, the received data is requantized by resolution adaptive ADCs. Measurement results show that FEC I consumes around 59 times of the amount of power in ADCs comparing to FEC II, whose power consumption in ADCs is around 26 times of FEC III. In addition, FEC III performs more efficiently than FEC II in the channel with deep fading. With FEC III, the receiver only needs to process the packet from the high-energy sub-carrier. Correspondingly, the processing power can be decreased which can not be achieved by using FEC I and II.

VII. ACKNOWLEDGEMENTS

We thank Geert Jan Laanstra for the technical support and Roel Schiphorst, Marnix Heskamp, Niels A. Moseley and Wu Yan for the useful comments and suggestions. Also, we thank the Dutch Ministry of Economic Affairs under the IOP Generic Communication - SenterNovem Program for the financial support.

REFERENCES

- [1] X. Shao, R. Schiphorst, and C. H. Slump, "An Opportunistic Error Correction Layer for OFDM Systems," *EURASIP Journal on Wireless Communications and Networking*, 2009.
- [2] A. Bahai, B. Saltzberg, and M. Ergen, *Multi-carrier Digital Communications: Theory and Applications of OFDM*. Springer, 2004.
- [3] J. Thomson and etc, "An integrated 802.11 a baseband and MAC processor," in *2002 IEEE International Solid-State Circuits Conference (ISSCC)*, vol. 2, 2002.
- [4] D.J.C. MacKay, "Fountain Codes," *IEE Communications*, vol. 152, no. 6, pp. 1062–1068, 2005.
- [5] IEEE, "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, High-Speed Physical Layer in the 5 GHz Band (IEEE 802.11a Standard, Part 11)," 1999.
- [6] IEEE, "Draft Standards for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, Enhancements for Higher Throughput (IEEE 802.11n Standard, Part 11)," Jan, 2007.
- [7] M. Luby, "LT Codes," *Proceedings of the 43rd Annual IEEE Symposium on Foundations of Computer Science*, pp. 271–282, 2002.
- [8] A. Shokrollahi, "Raptor Codes," *IEEE Transaction on Information Theory*, vol. 52, 2006.
- [9] D.J.C. MacKay, *Information Theory, Interference, and Learning Algorithms*. Cambridge University Press, 2003.
- [10] T. Schmidl, D. Cox, T. Inc, and T. Dallas, "Robust frequency and timing synchronization for OFDM," *IEEE Transactions on Communications*, vol. 45, no. 12, pp. 1613–1621, 1997.