An LPC-Based Fronthaul Compression Scheme

Leonardo Ramalho, Maria Nilma Fonseca, Aldebaro Klautau,
Chenguang Lu, Miguel Berg, Elmar Trojer, and Stefan Höst

***Abstract*—Several new architectures are under investigation for cloud radio access networks, assuming distinct splits of functionality among the network elements. Consequently, the research on radio data compression for the fronthaul is based on assumptions that correspond to a wide variety of tradeoffs among data rate, signal distortion, latency and computational cost. This paper describes a method for LTE downlink point-to-point signal compression based on linear prediction and Huffman coding, which is suitable for low cost encoding and decoding units with stringent restrictions on power consumption. The proposed method can work at various compression factors, such as 3.3:1 at an average EVM of 0.9%, or 4:1 at an average EVM of 2.1%.**

***Keywords*—C-RAN, Fronthaul, LTE signal compression, LPC.**

I. Introduction

THE transmission of *in-phase* (I) and *quadrature* (Q) samples between *baseband unit* (BBU) and *remote radio unit* (RRU) in *cloud radio access networks* (C-RAN) may require very high data rates [1]. For a given antenna and carrier, each baseband complex-valued IQ sample of an LTE signal is represented by $2b$ bits (b for I and b for Q). For example, assuming a MIMO system with two transmission antennas, a sampling frequency of 30.72 MHz (for 20 MHz LTE signals) and $b = 15$ would require 2×30.72 Msps $\times$ 30 bits $\approx$ 1.84 Gbps, for downlink. In this case, if a compression of 2:1 is applied to IQ samples, the downlink signal of the given example could be transported over legacy Gigabit Ethernet. Hence, *fronthaul signal compression* (FSC) is an important enabling technology in C-RAN.

Point-to-point [2]–[7] and distributed (e. g. [8]) FSC methods have been proposed. In point-to-point FSC methods, as this work is focused on, there are many possible splits of functionality between BBU and RRU. The one adopted in [6] moved most of the modulation / demodulation processing from the BBU to the RRU. This allows the fronthaul to transport only the bits needed as input to QAM constellation mapping. A huge compression factor $F = 30$ (or 30:1) is then achieved, at the expense of having a significant parcel of the baseband processing at the RRU and lacking of radio standard transparency, which needs different implementations for various standards. In contrast, the focus here is on a time-domain method that is standard transparent. As indicated in [9], a great amount of redundancy can be compressed out in time domain.

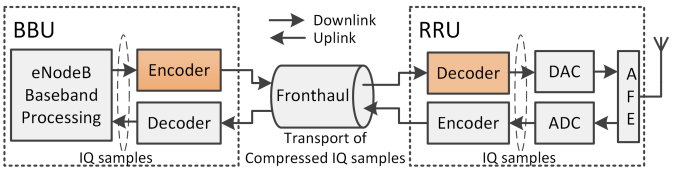

Fig. 1. Transmission of compressed IQ samples between BBU and RRU. In downlink, the encoder of BBU compresses the IQ samples and the decoder at RRU recovers the IQ samples.

Fig. 1 shows how time-domain methods can be used in a C-RAN architecture. This work focuses in downlink signals, so, the encoder at BBU side compresses the baseband IQ samples before transporting them over the fronthaul, and the decoder at RRU recovers the IQ samples with relatively low distortion.

Resampling is widely used in FSC [2]–[5] because the LTE signal is *oversampled*. For example, in 20 MHz LTE, only 1200 subcarriers are used out of 2048 subcarriers generated by IFFT. This corresponds to an oversampling factor of 1.7.

In [2], the LTE signal is resampled by a factor of $\frac{2}{3}$, scaled in blocks and then quantized. The decoder performs all inverse operations and the IQ samples are recovered under certain distortion. A method similar to [2] was adopted in [3] with distinct encoding strategies after resampling.

In [4], resampling, block scaling, vector quantization (VQ) and entropy coding were adopted. The VQ improves performance in terms of rate-distortion, but impacts the computational cost at the encoding stage, which is approximately two orders of magnitude higher than the proposed method (decoding is simpler but resampling is still required in [4]).

Relatively low complexity FSC methods were shown in [7]. These methods encode the difference between the current and previous sample. They were tested with single-carrier signals and relatively large oversampling factors, but when applied to LTE signals, their performance was not competitive.

Instead of using resampling to reduce the LTE signal oversampling overhead, this work proposes a time-domain low complexity FSC method that specializes the well-known *linear predictive coding* (LPC, see e. g. [10]), taking into account the frame-based LTE structure.

The remaining of this paper is organized as follows. Section II describes the proposed FSC scheme. Section III shows the performance of the proposed method and compares it with the baselines [2] and [4]. Finally, Section IV concludes the paper.

This work was supported in part by the Innovation Center, Ericsson Telecomunicações S.A., Brazil, CNPq and the Capes Foundation, Ministry of Education of Brazil, and by the European Union through the 5G-Crosshaul project (H2020-ICT-2014/671598).

L. Ramalho, M. N. Fonseca, and A. Klautau are with Computer and Telecommunications Engineering Department at Federal University of Para, Belem 66615-170, Brazil (e-mails: {leonardolr, nilmafonseca, aldebaro}@ufpa.br).

C. Lu, M. Berg and E. Trojer are with Ericsson Research, Kista, Sweden (e-mails: {chenguang.lu, miguel.berg, elmar.trojer}@ericsson.com).

S. Höst is with Department of Electrical and Information Technology, Lund University, Sweden (e-mail: stefan.host@eit.lth.se).

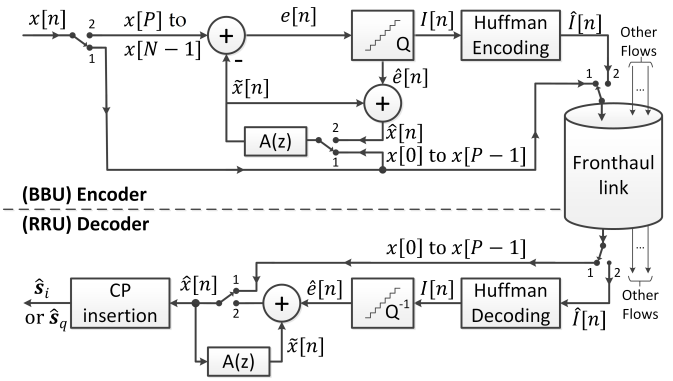

Fig. 2. Proposed PUSQH method using LPC and entropy coding for each individual OFDM symbol. The first P samples are passed to the receiver while the remaining $N-P$ are encoded with LPC and Huffman.

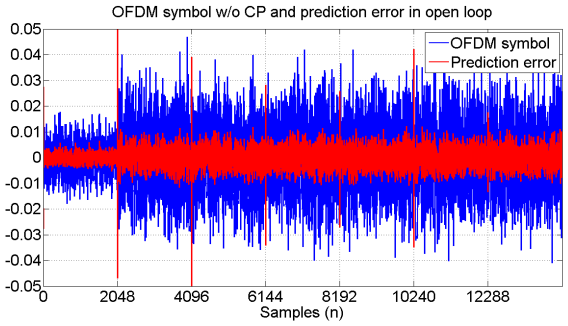

Fig. 3. Real part of an LTE signal $x[n]$ and its open loop prediction error $e[n]$ illustrating the *block effect* in the border of consecutive OFDM symbols. The LTE signal has BW of 20 MHz and each OFDM symbol has 2048 samples (without the CP).

II. Proposed Method

In LPC, a predictor $A(z) = \sum_{i=1}^{P} a_i z^{-i}$ uses the P previously quantized signal samples to create a prediction $\tilde{x}[n]$ of the current sample $x[n]$. The prediction error $e[n] = x[n] - \tilde{x}[n]$, which ideally has a white spectrum, is then quantized and transmitted. The decoder should be able to reconstruct the sample $\hat{x}[n] = \hat{e}[n] + \tilde{x}[n]$ using the quantized version $\hat{e}[n]$ of the prediction error. At both encoder and decoder, the prediction $\tilde{x}[n] = \sum_{i=1}^{P} a_i \hat{x}[n-i]$ uses $\hat{x}[n]$ because the original signal $x[n]$ is not available at the decoder [10]. In this work, the structure of LPC was changed by inserting switches in the LPC encoder and decoder, as shown in Fig. 2. These switches help to decrease the maximum values of $|e[n]|$ and thus reducing the quantization noise.

The quantizer creates an index $I[n]$ of b_e bits to represent $\hat{e}[n]$. Entropy coding, e. g. Huffman, can be used to reduce the average rate to a value L closer to the *entropy* of $I[n]$ [10]. These well-known concepts of predictive and entropy coding were used to design the proposed PUSQH (*predictive uniform scalar quantization with Huffman coding*) method, which is described in the sequel.

PUSQH encodes individually each OFDM symbol without the CP (cyclic prefix). The real and imaginary (I and Q) components are also individually processed. More specifically, a given I or Q component of an OFDM symbol with N real-valued samples is denoted as $x[n]$, where N is the IFFT size, and encoded as depicted in Fig. 2. The scalar quantizer represents $\hat{e}[n], n = P, \ldots, N-1$ with b_e bits, and is followed by a Huffman encoder. The N_{CP} samples of the CP are not transmitted by the encoder. At the decoder, all inverse operations are performed to obtain the reconstructed symbol $\hat{\mathbf{s}} = \hat{\mathbf{s}}_i + j\hat{\mathbf{s}}_q$, with the CP included.

The first P samples of $x[n]$ are losslessly encoded, i. e., $\hat{x}[n] = x[n]$, for $n = 0, \ldots, P-1$. At the beginning of each OFDM symbol, these first P samples are used to initialize the predictor memory. This is indicated by the switches at position 1 in Fig. 2. The prediction error $e[n]$ is calculated for the remaining samples $x[n], n = P, \ldots, N-1$, quantized and transmitted as $\hat{I}[n]$. The switches at position 2 indicates that after the predictor memory is filled with the first P samples, the remaining iterations use predictions that rely on the lossy encoding of $e[n]$.

The mentioned lossless coding of the first P samples is used to avoid the *block effect* among OFDM symbols, which is illustrated in Fig. 3. The *block effect* creates relatively large prediction errors between OFDM symbols. Circumventing this effect is a crucial feature of PUSQH to achieve improved performance, since this large $e[n]$ degrades the prediction $\tilde{x}$ based on quantized samples $\hat{x}$. The *block effect* is eliminated here by simply transmitting the original first P samples ($x[0]$ to $x[P-1]$) as side information using the original resolution (e. g., 30-bits per complex sample). It is assumed that the system knows where each OFDM symbol starts and ends (the two endpoint indexes). This is a sensible assumption: for example, the CPRI specification supports for the downlink the option of cyclic prefix (CP) insertion at the RRU [11].

There are many methods to calculate the coefficients $\mathbf{a} = [a_1, \ldots, a_P]$ of $A(z)$. For example, given the estimated values $\hat{R}(\tau)$ of the autocorrelation function, the corresponding Toeplitz autocorrelation matrix $\mathbf{R}$ and vector $\mathbf{r} = [\hat{R}(1), \ldots, \hat{R}(P)]$ can be used as inputs to the Levinson-Durbin algorithm [10], which obtains the filter coefficients $\mathbf{a} = \mathbf{R}^{-1}\mathbf{r}$. If $A(z)$ is calculated off-line, as it is in this work, its design does not impact the computational cost of real-time operation and can focus on performance. The following strategy takes into account the mentioned *block effect* and leads to improved performance: $R(k, \tau)$ is obtained for each OFDM symbol k (i. e. for each I or Q sequence $x[n]$) and their average $\hat{R}(\tau) = \mathbb{E}[R(k, \tau)]$ is used, where $\mathbb{E}[\cdot]$ is the expected value.

The average output rate (R) achieved by the proposed method in bits per IQ component is shown in Eq. (1):

$$R = \frac{L(N-P) + bP}{(N + N_{\text{CP}})}, \tag{1}$$

where L is the average number of bits when entropy coding is applied to $I[n]$, N is the number of samples per OFDM symbol, P is the predictor order, b is the number of bits for uncompressed I or Q component and N_{CP} is the number of samples in the CP. Indeed, the compression factor can be calculated as $F = b/R$.

Regarding computational cost, the CP insertion is a simple operation and Huffman encoding/decoding can be efficiently implemented by a look-up table [12]. The cost of the remain-

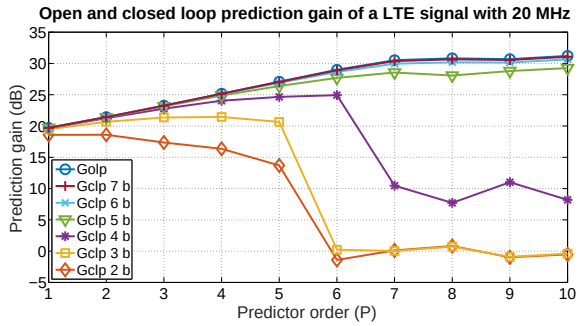

Fig. 4. Prediction gains of an LTE signal with BW of 20 MHz, where e. g., "7 b" indicates $b_e = 7$ bits.

TABLE I
RATE-DISTORTION RESULTS OF PUSQH FOR DISTINCT COMBINATIONS OF TRAINING (C1 OR C6) AND TEST (C1-C8) CONFIGURATIONS.

			$P = 4$		$P = 6$	
ID	LTE parameters: BW - M - aRBs	Train (ID)	Avg. EVM	R (bits)	Avg. EVM	R (bits)
C1	20 - 64 - 100	C1	1.65%	4.39	1.02%	4.47
C2	20 - 16 - 100		1.65%	4.39	1.02%	4.47
C3	20 - 4 - 100		1.65%	4.39	1.02%	4.47
C4	20 - 64 - 10		1.65%	3.85	1.01%	3.9
C5	10 - 64 - 50	C6	1.62%	4.45	1.05%	4.48
C6	10 - 16 - 50		1.62%	4.45	1.05%	4.48
C7	10 - 4 - 50		1.62%	4.45	1.04%	4.48
C8	10 - 16 - 8		1.56%	3.91	1.03%	3.97

ing linear prediction stage can be estimated as follows.

In the PUSQH method, the compression of LTE signals is performed with P MAC operations followed by two additions for $N - P$ samples out of $N + N_{\text{CP}}$ samples that compose $\hat{\mathbf{s}}_i$ or $\hat{\mathbf{s}}_q$. Hence, in average, the proposed method requires $C_{\text{P}} = \frac{2(P+2)(N-P)}{N+N_{\text{CP}}}$ arithmetic operations per complex-valued sample in order to implement prediction.

The method in [2] requires $C_{\text{R}} \approx 2\left(\frac{N_f}{2} + \frac{2}{3}\right)$ arithmetic operations, where N_f is the order of a symmetric FIR filter in the resampling implementation. The method in [4] have the same computational costs of [2] plus the VQ searching operations. Thus the method in [4] requires $C_{\text{VQ}} = C_{\text{R}} + \frac{(L_{\text{VQ}}+1)N_{\text{SO}}}{L_{\text{VQ}}}$ arithmetic operations, where L_{VQ} is the dimension of VQ and N_{SO} is the number of searching operations. For example, adopting $P = 4$ for PUSQH, $N_f = 64$ as in [3] for the baseline [2], and $L_{\text{VQ}} = 2$ and $N_{\text{SO}} = 288$ for the baseline [4], the computational cost per input complex-valued sample for the proposed method, [2] and [4] are such that $C_{\text{R}} \approx 5C_{\text{P}}$ and $C_{\text{VQ}} \approx 60C_{\text{P}}$. These comparisons assumed that addition, multiplication and MAC operations have the same cost.

PUSQH is also very competitive with respect to latency. A method that relies on resampling using a FIR filter of order N_f has to cope with a minimum latency of $N_f/2$ imposed by the filter's group delay. Using PUSQH, the lossless transmission of the first P samples allows the decoder to output these samples as soon as they are received and, for $n > P$, the process continues without further delay. And for improved latency, the CP can be transmitted over the fronthaul, buffered at the RRU, and repeated at the end of the OFDM symbol. In the sequel, PUSQH is evaluated with respect to rate-distortion.

III. Evaluation Methodology and Results

The proposed method was evaluated using the *error vector magnitude* (EVM) versus the average number of bits per component sample R or compression factor F, which are common figures of merit associated to FSC methods. The average RMS EVM was calculated here as indicated in Annex E of [13]. As a guideline, the overall link should have a maximum EVM of 17.5, 12.5, 8 and 3.5% to support QPSK, 16 , 64 and 256QAM [13], respectively. Obviously the signal distortion introduced by FSC methods should be kept as low as possible because the downlink signal will be later corrupted by e. g. noise, channel impairments, phase noise in the upconverter and intermodulation distortion in the amplifier.

The adopted evaluation methodology took into account that an LTE frame uses several distinct signals. Care must be exercised when evaluating FSC methods that rely on training data such as entropy coding and predictors. Here, disjoint sets of *training* and *test* data with a variety of LTE signals were used, and it was avoided to design quantizers or predictors using a specific LTE signal and evaluate the FSC using only the same kind of signal.

When choosing the predictor order P, a convenient figure of merit is the *open-loop prediction gain* $G_{\text{olp}} = 10\log_{10}\left(\frac{\mathbb{E}[|x[n]|^2]}{\mathbb{E}[|e[n]|^2]}\right)$ in dB [10], where the quantizer is not used ($b_e \to \infty$). As an example, for a 20 MHz LTE signal, this G_{olp} increases linearly from $P = 1$ to 6, and then starts to saturate, as depicted in Fig. 4.

The value of G_{olp} is convenient because it does not depend on the adopted quantizer. However, the actual performance is better inferred from the *closed-loop gain* $G_{\text{clp}} = 10\log_{10}\left(\frac{\mathbb{E}[|x[n]|^2]}{\mathbb{E}[|\hat{e}[n]|^2]}\right)$, which uses the power of closed-loop prediction error [10]. For $b_e \geq 6$, the two gains are essentially the same, i. e., $G_{\text{clp}} \approx G_{\text{olp}}$, as indicated in Fig. 4.

Table I lists results for distinct configurations for the LTE signals, with identifiers (ID) C1 to C8. The configurations C1 to C4 correspond to an LTE bandwidth (BW) of 20 MHz, while C5 to C8 have BW = 10 MHz. In LTE, RB stands for *resource block* and there are 100 and 50 available RBs when BW is 20 and 10 MHz, respectively. The simulations also take into account the number of active RBs (aRBs), which indicates the cell loading. For example, C4 and C8 correspond to a lightly loaded situation given that only 10 and 8 RBs are being used (out of 100 and 50), respectively. The other signals in Table I are fully loaded. The modulation order M adopted is 64, 16 or 4. The motivation to use predictor orders of $P = 4$ and $P = 6$ came from the prediction gains shown in Fig. 4. In all cases of Table I the quantizer has $b_e = 6$ bits.

The predictor $A(z)$, quantizers and Huffman codes were designed from a training sequence with configuration C1 for tests with 20 MHz signals (C1-C4) and configuration C6 for testing with 10 MHz signals (C5-C8). The predictor was found to be relatively robust to mismatched conditions, when the configuration of the test signal differs from the one used for training with respect, e. g., to modulation and number of aRBs.

The proposed FSC scheme is contrasted to [2] and [4] in Fig. 5, which was generated with an LTE test signal of

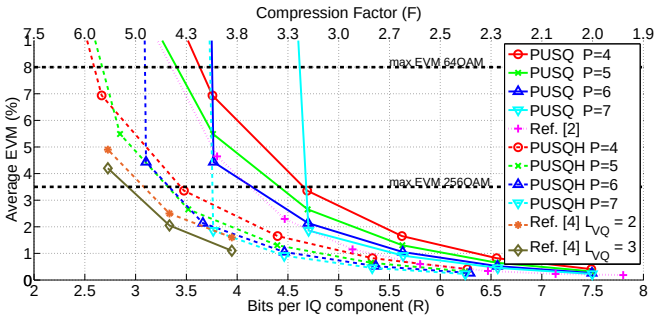

Fig. 5. EVM versus both rate R and compression factor F for PUSQH, PUSQ (proposed method without Huffman) and the baselines [2] and [4].

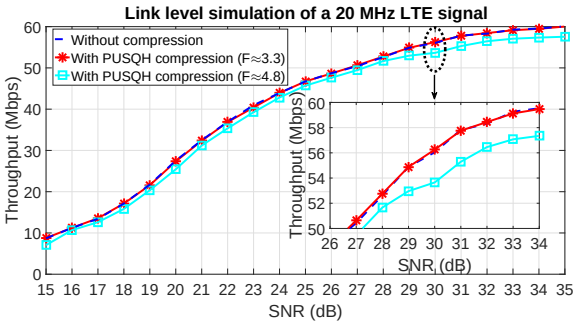

Fig. 6. LTE link level simulation with and without compression. The throughput is the rate of received transport block bits without errors.

configuration C1 (BW = 20 MHz) and corresponds to 10 LTE frames (1400 OFDM symbols). The parameters of LPC were changed as follows: the predictor order was varied from $P = 4$ to 7 and the number of quantization bits from $b_e = 2$ to 8 bits. The results in Fig. 5 indicate that PUSQH outperforms [2] when $R < 5.5$ bits, while achieving similar rate-distortion performance for higher R. As expected from information theory, the VQ-based method of [4] has the potential of outperforming alternatives based on scalar quantization with respect to rate-distortion. But for EVM lower than 2%, [4] with $L_{\text{VQ}} = 2$ brings no significant advantage while it increases the computational cost C_{VQ} by more than an order of magnitude when compared to the proposed method. Increasing L_{VQ} to 3, decreases R in approximately 0.5 bit for an EVM $\approx 1\%$, but in this case $C_{\text{VQ}} \approx 600 C_P$, which can be unfeasible in some C-RAN architectures. Fig. 5 also shows the proposed method without Huffman encoding (PUSQ) which has lower computational cost and lower performance than PUSQH, but it is still close to the performance of [2] for $P = 6$ and $P = 7$.

To obtain further insight on the impact of the PUSQH over the link, a complete downlink simulation was performed over the Extended Pedestrian A (EPA) fading channel model with a Doppler frequency of 5 Hz. A thousand sub-frames of a 20 MHz LTE signal were simulated with and without compression. See reference channel R.9 in annex A of [14] for more details of the LTE signal. The predictor order is $P = 6$ and the predictor error is quantized with $b_e = 6$ and $b_e = 4$ bits, corresponding to compression factors $F \approx 3.3$ and $F \approx 4.8$. Fig 6 shows that the compression factor can be controlled to avoid decreasing the throughput if needed. The method achieves performance of throughput very close to that of the uncompressed signal, specially for low F (e. g. $F \approx 3.3$). But, when $F \approx 4.8$, the impact is still low, e. g., for a channel with SNR = 30 dB, the loss of throughput with PUSQH is about 2.4 Mbps when compared to the uncompressed case.

IV. Conclusion

This work proposed a method for LTE signal compression that is based on predictive and Huffman coding. The performance of the new method was evaluated by simulations, showing competitive compression with the resampling methods. The LPC filter has significantly fewer taps than the ones typically used in resampling schemes. Lower computational cost and latency are other advantages of the proposed method.

Due to limited space, the method was evaluated here only for downlink, but it can be also used for uplink with few adaptations, as well as for OFDM signals in general.

References

[1] C.-L. I, J. Huang, R. Duan, C. Cui, J. Jiang, and L. Li, "Recent progress on C-RAN centralization and cloudification," *Access, IEEE*, vol. 2, pp. 1030–1039, 2014.

[2] B. Guo, W. Cao, A. Tao, and D. Samardzija, "LTE/LTE-A signal compression on the CPRI interface," *Bell Labs Technical Journal*, vol. 18, no. 2, pp. 117–133, Sept 2013.

[3] D. Samardzija, J. Pastalan, M. MacDonald, S. Walker, and R. Valenzuela, "Compressed Transport of Baseband Signals in Radio Access Networks," *Wireless Communications, IEEE Transactions on*, vol. 11, no. 9, pp. 3216–3225, September 2012.

[4] H. Si, B. L. Ng, M. S. Rahman, and J. Zhang, "A novel and efficient vector quantization based CPRI compression algorithm," *CoRR*, 2015. [Online]. Available: http://arxiv.org/abs/1510.04940

[5] K. Nieman and B. Evans, "Time-domain compression of complex-baseband LTE signals for cloud radio access networks," in *Global Conference on Signal and Information Processing (GlobalSIP), 2013 IEEE*, Dec 2013, pp. 1198–1201.

[6] J. Lorca and L. Cucala, "Lossless compression technique for the fronthaul of LTE/LTE-advanced cloud-RAN architectures," in *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2013 IEEE 14th International Symposium and Workshops on a*, June 2013, pp. 1–9.

[7] A. Vosoughi, M. Wu, and J. R. Cavallaro, "Baseband signal compression in wireless base stations," in *Global Communications Conference (GLOBECOM), 2012 IEEE*, Dec 2012, pp. 4505–4511.

[8] S.-H. Park, O. Simeone, O. Sahin, and S. Shamai Shitz, "Fronthaul Compression for Cloud Radio Access Networks: Signal processing advances inspired by network information theory," *Signal Processing Magazine, IEEE*, vol. 31, no. 6, pp. 69–79, Nov 2014.

[9] C. Lu, M. Berg, L. Ramalho, and A. Klautau, "Bit-Rate Bound Derivation for Compressed Time-domain Fronthaul," in *EuCNC 2016 workshop Towards Converged X-Haul for 5G Networks*, June 2016. [Online]. Available: http://5g-crosshaul.eu/wp-content/uploads/2016/09/Bit-rate-bound-for-FH-presentation_EUCNC-Workshop-W04b_Ericsson.pdf

[10] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*. Kluwer Academic Publishers, 1992.

[11] "Common Public Radio Interface (CPRI) specification v6.1," http://www.cpri.info/downloads/CPRI_v_6_1_2014-07-01.pdf, July 2014.

[12] Z. Aspar, Z. Yusof, and I. Suleiman, "Parallel Huffman decoder with an optimized look up table option on FPGA," in *TENCON 2000. Proceedings*, vol. 1, 2000, pp. 73–76 vol.1.

[13] 3GPP TS 36.104, "Evolved Universal Terrestrial Radio Access (E-UTRA); Base Station (BS) radio transmission and reception," 2014. [Online]. Available: http://www.3gpp.org/dynareport/36104.htm

[14] 3GPP TS 36.101, "Evolved Universal Terrestrial Radio Access (E-UTRA); User Equipment (UE) radio transmission and reception," 2016. [Online]. Available: http://www.3gpp.org/dynareport/36101.htm