

A New O-RAN Compression Approach for Improved Performance on Uplink Signals

Marcos Silva, Leonardo Ramalho, Igor Almeida, Eduardo Medeiros, Miguel Berg and Aldebaro Klautau

Abstract—This work evaluates the O-RAN compression methods specified for in-phase and quadrature (IQ) data compression, which are applied to transport the frequency domain representation of the radio signals. The methods were evaluated in terms of computational cost and quantization-noise ratio vs IQ bit-width. It was found that the O-RAN compression algorithm with the best performance highly depends on the signal power. Thus, one of the contributions of this work is a compression method that selects the best method for each physical resource block (PRB) instead of using a single compression method for a set of PRBs as in the current O-RAN specification. A complete description of how to implement the proposed method meeting the O-RAN standard is also provided in the paper. The selection of the best method for each PRB is particularly important for the uplink signals, where the power of the received signals can be very different depending on the UE channel.

Index Terms—C-RAN, O-RAN, Fronthaul, Compression

I. INTRODUCTION

MOBILE operators are under constant pressure to increase their capacity while constraining cost and providing competitive mobile services. Radio Access Networks (RANs) are facing a rapid increase in traffic demand, due to increasing number of devices connected to the network, and higher quality of service requested by the users [1]. In this context, Cloud Radio Access Network (C-RAN) is an enabling technology for future mobile networks, promoting centralized processing, resource sharing, and energy-efficient infrastructure [2].

In C-RAN, a baseband unit (BBU) is responsible for most of the baseband processing, and the remote radio unit (RRU) is responsible for the radio frequency (RF) processing and can implement some baseband processing. The radio signal is transported between the BBU and RRU over the fronthaul [3]. The fronthaul link usually uses a high data rate to meet the radio signal transport requirements. Many solutions have been proposed to decrease the fronthaul data rate, such as functional split and IQ data compression [4].

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.

M. Silva, L. Ramalho and A. Klautau are with LASSE - 5G Group, Av Perimetral km 01, Guamá, 66075-750, Federal University of Pará, Belém, Brazil (e-mails: marcos.lima.silva@itec.ufpa.br, [leonardolr.aldebaro]@ufpa.br).

I. Almeida is with Ericsson is with Ericsson Research, Rod. Eng. Ermênio de Oliveira Penteado, 57,5 - Parque São Lourenço, Indaiatuba, Brazil (e-mail:igor.almeida@ericsson.com).

E. Medeiros and M. Berg are with Ericsson Research, 164 80, Stockholm, Sweden (e-mails: [eduardo.medeiros, miguel.berg]@ericsson.com).

Digital Object Identifier: 10.14209/jcis.2022.3

There are different approaches in the C-RAN architecture for fronthaul signal compression, such as: point-to-point compression, distributed source coding, Compressive Sensing (CS) based compression, and spatial filtering [5]. In addition, the different functional splits allow new approaches to decrease the fronthaul traffic [6], [7].

The O-RAN alliance provides an open specification for implementing the fronthaul and IQ data compression [8], [9]. More specifically, the O-RAN alliance conducts studies on eight working groups (WG), each focusing on different areas, to make the new RAN concept possible [10], [11]. For example, the "WG4 - The Open Fronthaul Interfaces Workgroup" defines standardization for fronthaul interfaces between BBU and radio equipment. The WG4 works on providing specifications and standards for a secure fronthaul, allowing interoperability between multi-vendor [2], [10], [11].

In this context, this manuscript will focus on the evaluation of the compression methods specified in WG4 for uplink signals: Block Floating Point (BFP), Block Scaling (BS), and μ -Law [8]. Based on that knowledge, the main contributions of this work are the elucidation of the O-RAN compression methods and the evaluation of their performance and computational cost.

Furthermore, we found that depending on the signal power, the performance of the compression methods can vary significantly. The O-RAN standard specifies that various PRBs should be compressed with a single compression method. Therefore, the PRBs with different power levels may have different levels of signal-to-quantization-noise ratio (SQNR) after compression. In this context, another contribution of this work is a framework that allows using the compression method with the best SQNR for each PRB instead of using a single compression method for various PRBs. The manuscript presents how the current O-RAN specification could apply the proposed method. This is particularly relevant for uplink signals, where the RRU might receive signals from user equipments (UEs) with very distinct power levels. The studies were carried out considering a Long Term Evolution (LTE) baseband uplink receiver chain. The performance of the compression methods and the proposed approaches are analyzed for different scenarios.

In summary, the main contributions of the paper are:

- We evaluate the performance of the O-RAN compression methods: block floating point (BFP), Block Scaling (BS), and μ -law;
- We provide an overview of the computational cost for each compression method;

- We simulate different scenarios in an uplink LTE receiver chain;
- We propose a new compression approach that presents better performance when compared to each method individually. Indeed, the proposed implementation provided in this paper meets the O-RAN specification.

The remaining of this manuscript is structured as follows. Section II shows some details of the O-RAN specification and the compression methods specified for uplink signals. Section III shows the simulation results and the computational cost analysis of the methods. Section IV presents a new compression approach that is compliant with the O-RAN specification. Section V shows the performance of the proposed method considering an uplink receiver chain, including scenarios with two users that have different levels of signal attenuation. Finally, Section VI concludes the paper.

II. O-RAN COMPRESSION METHODS FOR UPLINK SIGNALS

The O-RAN specification defines several compression methods to transport the IQ samples over the fronthaul efficiently. The compression methods are combined with functional split 7.2x [8], similar to 3GPP functional split 7.2 [4]. The RRU implements a few physical baseband processing in the uplink, such as cyclic prefix removal and fast-fourier transform (FFT). Then, the output of the FFT can be compressed before transporting on fronthaul.

The compression methods process each PRB individually. A single PRB is a block composed of 12 complex samples corresponding to 12 subcarriers in an orthogonal frequency division multiplexing (OFDM) symbol, where the real and imaginary parts of the IQ sample are represented with 16-bit signed integers. For instance, an OFDM symbol with 1200 subcarriers is composed of 100 PRBs. In the current O-RAN specification, a given number of PRBs are compressed with the same compression method in each cell section, as specified in its IQ data frame format [8, Section 6.3].

After compression, each complex sample can be represented with 2 to 16 bits. The number of bits used to represent each complex sample component, real or imaginary, is denoted as $iqWidth$. Thus, each complex sample is represented with $2 \times iqWidth$. The inputs of the compression methods are the $iqWidth$ and the original PRB composed of 12 IQ samples. The output is composed of the compressed IQ data and eight bits of side information that depend on the compression method, per the O-RAN specification [8]. Not all methods need to use the entire side information field. In the following sections, the operation of each method is presented in more detail.

A. Block Floating Point Compression

In the BFP compression method, each complex sample of a PRB is converted to floating-point, with real and imaginary components represented by a mantissa, an $exponent$, and a sign. A single $exponent$ is shared among all elements of the PRB. In addition, the bits of the mantissa are reduced to a target value so that the compressed data uses fewer bits than the original PRB.

Algorithm 1 Block Floating Point Compression Algorithm

Input: $PRB [1, \dots, 12], iqWidth$
Output: $cPRB [1, \dots, 12], exponent$

- 1: $maxValue \leftarrow \max(\max(|\text{Re}(PRB)|), \max(|\text{Im}(PRB)|))$
- 2: $mostSignificantBit \leftarrow \lfloor \log_2(maxValue) + 1 \rfloor$
- 3: $exponent \leftarrow \max(mostSignificantBit - iqWidth, 0)$
- 4: **for** $i \leftarrow 1$ to 12 **do**
- 5: $reShiftRight \leftarrow \text{Re}(PRB[i]) \times 2^{-exponent}$
- 6: $imShiftRight \leftarrow \text{Im}(PRB[i]) \times 2^{-exponent}$
- 7: $\text{Re}(cPRB[i]) \leftarrow reShiftRight\{iqWidth:1\}$
- 8: $\text{Im}(cPRB[i]) \leftarrow imShiftRight\{iqWidth:1\}$
- 9: **end for**

Algorithm 2 Block Floating Point Decompression Algorithm

Input: $cPRB [1, \dots, 12], exponent$
Output: $dPRB [1, \dots, 12]$

- 1: **for** $i \leftarrow 1$ to 12 **do**
- 2: $\text{Re}(dPRB[i]) \leftarrow \text{Re}(cPRB[i]) \times 2^{exponent}$
- 3: $\text{Im}(dPRB[i]) \leftarrow \text{Im}(cPRB[i]) \times 2^{exponent}$
- 4: **end for**

Algorithm 1 shows a detailed description of the BFP compression method [8]. The first step is to find the maximum absolute value among all real and imaginary samples of the PRB (line 1). Then, an $exponent$ is calculated for this maximum number (lines 2 and 3). Next, the same $exponent$ is used to represent all other numbers of the PRB as floating point. Finally, the bits of the sign and the mantissa are represented with the target compression number of bits. The multiplication operations on lines 5 and 6 mean bitwise shift right by $exponent$ value, and lines 7 and 8 get the compression word least significant bits (LSBs) of the processed information.

The output of the BFP method is composed of the compressed PRB ($cPRB$) and an $exponent$ as side information. The $cPRB$ is a vector containing 12 compressed IQ samples, each formed by a sign and mantissa. Then, the $cPRB$ and the exponent are sent to the decoder, which performs the reverse operations. The decompressed PRB ($dPRB$) can be recovered with a bitwise shift left, as shown in lines 2 and 3 in Algorithm 2.

A remarkable feature of the BFP compression is that it can be lossless if the $exponent$ is zero. This happens when the amplitude of the elements in the original PRB is relatively low or compression word length is relatively high.

B. Block Scaling Compression

The BS compression method scales all complex samples of a PRB and quantizes the scaled samples, as shown in Algorithm 3. The first step is to find the maximum absolute value among all real and imaginary components of the PRB, which we will refer to hereinafter as $maxValue$. The second step is to quantize the $maxValue$ (line 2). A lookup table (LUT) can efficiently implement the quantization with two outputs: the $scalerValueComp$ (the quantized version of $1/maxValue$) and $scaler8Bits$ representing the scaler with 8 bits.

Algorithm 3 Block Scaling Compression Algorithm

Input: $PRB [1, \dots, 12]$, $iqWidth$
Output: $cPRB [1, \dots, 12]$, $scaler8Bits$

- 1: $maxValue \leftarrow \max(\max(|\text{Re}(PRB)|), \max(|\text{Im}(PRB)|))$
- 2: $[scalerValueComp, scaler8Bits] \leftarrow \text{LUT}(maxValue)$
- 3: **for** $i \leftarrow 1$ to 12 **do**
- 4: $reScaledSamples \leftarrow \text{Re}(PRB[i]) \times scalerValueComp$
- 5: $imScaledSamples \leftarrow \text{Im}(PRB[i]) \times scalerValueComp$
- 6: $\text{Re}(cPRB[i]) \leftarrow \text{quantize}(reScaledSamples, iqWidth)$
- 7: $\text{Im}(cPRB[i]) \leftarrow \text{quantize}(imScaledSamples, iqWidth)$
- 8: **end for**

Algorithm 4 Block Scaling Decompression Algorithm

Input: $cPRB [1, \dots, 12]$, $scaler8Bits$
Output: $dPRB [1, \dots, 12]$

- 1: $scalerValueDecomp \leftarrow \text{LUT}(scaler8Bits)$
- 2: **for** $i \leftarrow 1$ to 12 **do**
- 3: $\text{Re}(dPRB[i]) \leftarrow \text{Re}(cPRB[i]) \times scalerValueDecomp$
- 4: $\text{Im}(dPRB[i]) \leftarrow \text{Im}(cPRB[i]) \times scalerValueDecomp$
- 5: **end for**

The following steps in Algorithm 3 are: to scale all IQ symbols in the PRB (lines 4 and 5) and to quantize the scaled version of the PRB (lines 6 and 7). Each real and imaginary part is quantized with $iqWidth$ bits. Since $scalerValueComp \approx 1/maxValue$, all scaled values are between -1 and 1. Thus a single quantization range of $[-1, 1]$ can be used in all scenarios.

After the quantization, the compressed data is transported over the fronthaul and the $scaler8Bits$ as side information. In the decoder, all steps are reversed, as shown in Algorithm 4. In decompression, the first step is to recover the scaling value that will bring back the values to the original levels. This operation can also be performed with a look-up table, where $scalerValueDec = 1/scalerValueComp$. Then, all values are multiplied by the scaling value, and an approximation of the original values is recovered.

C. μ -Law Compression

In O-RAN, the μ -Law method implements companding to decrease the bitwidth of the compressed samples efficiently. Companding is a nonlinear scaling method, where small input values are more amplified than higher input values before quantization. The μ value consists of the μ -Law compression parameter, and the O-RAN specification uses $\mu = 8$ [8].

The O-RAN implementation of the μ -Law method is shown in Algorithm 5. The input is, again, a single PRB. The first operation is to find the maximum absolute value for each real and imaginary in all resource elements (line 1). Next, a variable called $compShift$ is calculated (line 2) with Algorithm 7, and is then used to scale the absolute value of the components (lines 6 and 7) before companding (lines 8 and 9). Algorithm 8 shows how the compand operations are defined on O-RAN, and its output is represented with $iqWidth - 1$ bits. Finally, the components of the compressed PRB are composed of the sign bit and the output of the compand method, which

Algorithm 5 μ -Law Compression Algorithm

Input: $PRB [1, \dots, 12]$, $iqWidth$
Output: $cPRB [1, \dots, 12]$, $compShift$

- 1: $maxValue \leftarrow \max(\max(|\text{Re}(PRB)|), \max(|\text{Im}(PRB)|))$
- 2: $compShift \leftarrow \text{compShiftCalculation}(maxValue)$
- 3: **for** $i \leftarrow 1$ to 12 **do**
- 4: $reSignSamples \leftarrow \text{sign}(\text{Re}(PRB[i]))$
- 5: $imSignSamples \leftarrow \text{sign}(\text{Im}(PRB[i]))$
- 6: $reAbsSll \leftarrow |\text{Re}(PRB[i])| \ll compShift$
- 7: $imAbsSll \leftarrow |\text{Im}(PRB[i])| \ll compShift$
- 8: $reComp \leftarrow \text{compand}(reAbsSll, iqWidth)$
- 9: $imComp \leftarrow \text{compand}(imAbsSll, iqWidth)$
- 10: $\text{Re}(cPRB[i]) \leftarrow [reSignSamples, reComp]$
- 11: $\text{Im}(cPRB[i]) \leftarrow [imSignSamples, imComp]$
- 12: **end for**

Algorithm 6 μ -Law Decompression Algorithm

Input: $cPRB [1, \dots, 12]$, $iqWidth$, $compShift$
Output: $dPRB [1, \dots, 12]$

- 1: **for** $i \leftarrow 1$ to 12 **do**
- 2: $[reSignSamples, reComp] \leftarrow \text{Re}(cPRB[i])$
- 3: $[imSignSamples, imComp] \leftarrow \text{Im}(cPRB[i])$
- 4: $reDecomp \leftarrow \text{decompand}(reComp, iqWidth)$
- 5: $imDecomp \leftarrow \text{decompand}(imComp, iqWidth)$
- 6: $reSignedDecomp \leftarrow [reSignSamples, reDecomp]$
- 7: $imSignedDecomp \leftarrow [imSignSamples, imDecomp]$
- 8: $\text{Re}(dPRB[i]) \leftarrow reSignedDecomp \gg compShift$
- 9: $\text{Im}(dPRB[i]) \leftarrow imSignedDecomp \gg compShift$
- 10: **end for**

are packed in a value of $iqWidth$ bits. All the compressed PRB are transported with the side-information $compShift$, which can be represented using only 3 bits.

The decoder recovers an estimate of the original PRB by performing all reverse operations, as shown in Algorithm 6. First, the decompression core is the decompand block (lines 4 and 5) described in Algorithm 9. Then, a version of the decompand value with sign is recovered on lines 6 and 7, and finally, the real and imaginary numbers are shifted right by $compShift$ bits.

III. PERFORMANCE AND COMPUTATIONAL COST OF THE O-RAN COMPRESSION METHODS

In this section, we evaluate the performance of each method in terms of the SQNR (Signal-to-Quantization Noise Ratio) and the number of bits to represent the complex samples. The SQNR consists of the ratio between the signal power in the input and the distortion inserted by the compression method, i.e., $\text{SQNR} = \mathbb{E}[|PRB|^2] / \mathbb{E}[|dPRB - PRB|^2]$, where PRB is the compressor input and $dPRB$ is the output of the decompression. Thus, the SQNR evaluates only the performance of the compression methods, i.e., other distortions such as quantization noise from analog-to-digital conversion and thermal noise in the receiver chain are not yet taken into account. The SQNR is calculated for different values of $iqWidth$, ranging from 1 to 8 bits.

Algorithm 7 Compshift Calculation Algorithm

Input: $maxValue$
Output: $compshift$
if $maxValue \geq 2^{14}$ **then**
 $compshift \leftarrow 0$
else if $2^8 \leq maxValue < 2^{14}$ **then**
 $compshift \leftarrow 15 - \lfloor \log_2(maxValue) + 1 \rfloor$
else
 $compshift \leftarrow 7$
end if

Algorithm 8 Compand Algorithm

Input: $absll, iqWidth$
Output: $comp$
1: $W \leftarrow iqWidth, A \leftarrow absll$
2: $B \leftarrow 2^{15-W}, C \leftarrow 2^{16-W}, E \leftarrow 2^{17-W}$
3: $D \leftarrow 2^{W-3}, F \leftarrow 2^{W-2}$
4: **if** $A > 32767$ **then** ▷ Saturation A value
5: $A \leftarrow 32767$
6: **end if**
7: **if** $A \leq 8192$ **then** ▷ Companding calculation
8: $comp \leftarrow A/B$
9: **else if** $A \leq 16384$ **then**
10: $comp \leftarrow A/C + D$
11: **else**
12: $comp \leftarrow A/E + F$
13: **end if**

Algorithm 9 Decomband Algorithm

Input: $absdec, iqWidth$
Output: $decomp$
1: $W \leftarrow iqWidth, A \leftarrow absdec$
2: $B \leftarrow 2^{15-W}, C \leftarrow 2^{16-W}, E \leftarrow 2^{17-W}$
3: $D \leftarrow 2^{W-3}, F \leftarrow 2^{W-2}, G \leftarrow 2^{W-1} - 1$
4: **for** $i \leftarrow 1$ **to** 12 **do**
5: **if** $A > G$ **then** ▷ Saturation A value
6: $A \leftarrow G$
7: **end if**
8: **if** $A \leq F$ **then** ▷ Decombanding calculation
9: $decomp \leftarrow A \times B$
10: **else if** $A \leq F + D$ **then**
11: $decomp \leftarrow A \times C - 8192$
12: **else**
13: $decomp \leftarrow A \times E - 32768$
14: **end if**
15: **end for**

For the performance analysis of the methods, PRBs were created with an LTE signal processing chain with random data. It is worth mentioning that the uplink signals that are compressed have not passed any channel equalization, and therefore are under the effects of attenuation and noise. The original real and imaginary parts are 16-bit signed integers, i.e., they assume values in the range $[-2^{15}, +2^{15} - 1]$.

As expected, the performance of the methods varies with $iqWidth$ as well as input power or standard deviation σ of the

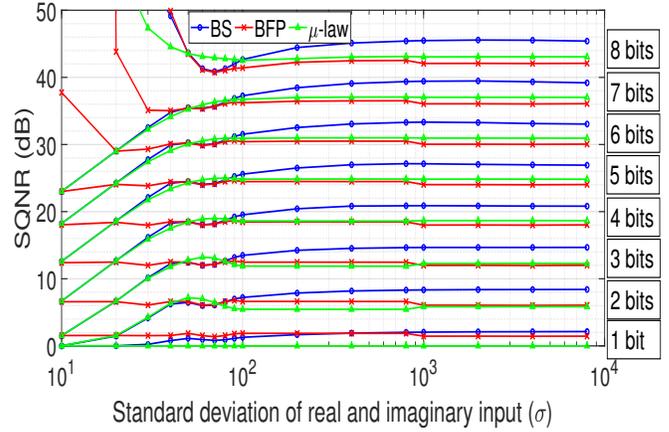


Figure 1: Performance of the methods for different values of $iqWidth$ and standard deviation of the real and imaginary parts of the input signal.

signal that is being compressed. Fig. 1 shows the SQNR versus σ for different values of $iqWidth$. For each value of $iqWidth$ there are three curves, one for each compression method.

Fig. 1 shows that the SQNR increases approximately by 6 dB for each 1 bit increase in $iqWidth$. For instance, when considering $\sigma = 10^3$, the SQNR of the BS compression increases from 2.1 dB to 45.5 dB in steps of 6.2 dB when $iqWidth$ varies from 1 to 8 bits. However, when σ is low, the performance of the BS compression decreases with σ when $iqWidth \leq 7$ bits. This is more noticeable for $\sigma \leq 10^2$.

Similar behavior is obtained with the μ -Law compression, where the performance decreases with σ . On the other hand, the performance of the BFP compression does not degrade as σ decreases. In fact, for some values of $iqWidth$, the SQNR of the BFP compression increases as σ decreases, as shown in Fig. 1 for $iqWidth = 6, 7, \text{ and } 8$ bits.

Based on the results of Fig. 1, the method with the best SQNR for a given $iqWidth$ depends on σ . For example, when $\sigma \leq 40$, the BFP compression consistently achieves the best SQNR for each $iqWidth$, and for $\sigma > 10^2$, BS compression achieves the best SQNR for $iqWidth > 1$ bit.

In fact, the best method depends on the elements of each PRB. For example, it is clear that when the *exponent* in Algorithm 1 is 0, BFP compression becomes lossless, and it would be the best compression method.

SQNR is an important metric to evaluate the distortion that the compression methods insert in the signal. However, during the implementation of the compression methods on an RRU, it is necessary to take into consideration their computational complexity. Therefore, the methods were also evaluated in terms of the computational cost by counting the number of operations required by each compression method to encode and decode one PRB.

The computed operations are: getting sign bit (sign), calculating the magnitude of the number, conditional testing, multiplication, addition, subtraction, bit shift, search on LUT, and quantization. The number of operations for each method is summarized in Tables I and II for compression and decom-

Table I: Computational cost for compression per PRB.

	Block Floating Point	Block Scaling	μ -Law
Sign	0	0	24
Absolute Value	24	24	24
Conditional	40	23	127
Multiplication	0	24	0
Addition	1	0	24
Subtraction	1	0	5
Bit Shift	39	0	53
LUT	0	1	0
Quantization	0	24	0

Table II: Computational cost for decompression per PRB.

	Block Floating Point	Block Scaling	μ -Law
Conditional	0	0	96
Multiplication	0	24	0
Subtraction	0	0	31
Bit Shift	24	0	54
LUT	0	1	0
Dequantization	0	24	0

pression, respectively. However, it should be noted that the exact number of operations can be different depending on the implementation, hardware, and instruction set available.

Based on Table I, Block Floating Point and Block Scaling perform similar number of operations, while μ -Law has the highest computational cost. On the decompression side, Table II shows that the methods with the lowest number of operations are Block Floating Point, Block Scaling, and μ -Law, respectively.

Details about the results of Tables I and II are discussed in the following paragraphs. First, we evaluate the BFP method. The 24 absolute value operations shown in Table I are used in line 1 of Algorithm 1 to calculate the magnitude of the real and imaginary components of the PRBs. Then, 23 conditional operations are used to find the *maxValue*. On line 2 of Algorithm 1, \log_2 and floor operations can be implemented by evaluating where the most significant bit 1 is located, so that it could be implemented with bit shift and conditional operations. Thus, the worst case would lead to 15 bit shift operations and 16 conditional bit checking, considering that the original information is represented with 16 bits. On line 3, the *exponent* is found with one addition, one subtraction, and one conditional operation. Finally, lines 5 and 6 perform two bit shift operations for each complex PRB, so the loop uses a total of 24 bit shift operations. Note that the multiplication by a power of 2 can be implemented with bit shift operations. In the BFP decompression, only 24 bit shifts are necessary, 2 for each complex *cPRB*, as shown in Algorithm 2.

Next, we analyze the required operations of Block Scaling compression. Line 1 of Algorithm 3 shows that the method also calculates a *maxValue* and requires the same operations as in the BFP method: 24 absolute values and 23 conditional operations. After that, a table look-up¹ is done in line 2. Finally, in the loop from line 3 to line 8 of Algorithm 3, 24 multiplications are performed, and 24 numbers are quantized.

¹The LUT function can be implemented with different approaches, such as binary search or memory read. The former requires $\log_2 N_{out}$ comparisons on average and the latter requires a memory with N_{in} words, where N_{out} is the number of possible output values and N_{in} is the number of possible inputs, considering the integer input.

In the decompression, a LUT operation is again used to calculate the *scalerValueDecomp*, followed by the loop that executes 24 dequantization and 24 multiplications on the *cPRBs* to recover estimation of the original PRB (*dPRB*), as shown in Algorithm 4.

In the compression, the μ -Law method also needs the *maxValue*, as done in BFP and BS, which requires 24 absolute values and 23 conditional operations. Then, in line 2 of Algorithm 5 *compshift* is found with 8 conditional operations, as shown in Algorithm 7. In the next step, 24 sign operations are performed on lines 4 and 5, followed by 24 bit shift operations on lines 6 and 7. The output of the bit shift left block passes through the compand block in lines 8 and 9, which requires 5 subtractions and 5 bit shift left to calculate lines 2 and 3 in Algorithm 8. Additionally, the 12 real samples and 12 imaginary samples require in the compand: 96 conditional operations, 24 additions, and 24 bit shift operations (the division in the compand can be implemented with bit shift, since the denominators *B*, *C*, and *E* are powers of two). In total, the number of operations required in the μ -Law compression is shown in Table I.

The μ -Law decompression requires fewer operations than compression, as shown in Table II. The sign and absolute values of the compressed samples can be obtained with bit read from the compressed bitstream, so there is no need to implement a math operation to recover them. The decompand shown in Algorithm 9 and in lines 4 and 5 of Algorithm 6 requires 7 subtractions and 6 bit shift operations to calculate *B*, *C*, *D*, *E*, *F*, and *G*. Counting for all sample components, the decompand would require up to 96 conditional operations, considering Algorithm 9. Additionally, the loop can use up to 24 bit shifts left to implement the multiplications and 24 subtractions.

IV. NEW COMPRESSION APPROACH

Because SQNR is influenced by input power, the best method for each PRB may be different, depending on its resource elements.

In order to take advantage of the best method in most cases, this work proposes choosing the best method in each PRB instead of selecting a single compression method for many PRBs as proposed in the O-RAN specification [8]. This approach can improve SQNR regardless of the signal levels, which might not be possible when a single compression method is used for a group of PRBs.

The proposed approach is illustrated in Fig. 2. The encoder compresses the PRB with the best method and sends to the decoder the following information: the compressed IQ samples (IQ comp), the side information of the compression method, and the method that was used to compress the IQ samples. Thus, two new operations are necessary: i) the encoder needs to inform the decoder which compression method was used for each PRB, and ii) the encoder needs to select which compression method would provide the best SQNR. Both issues are treated in the following sections.

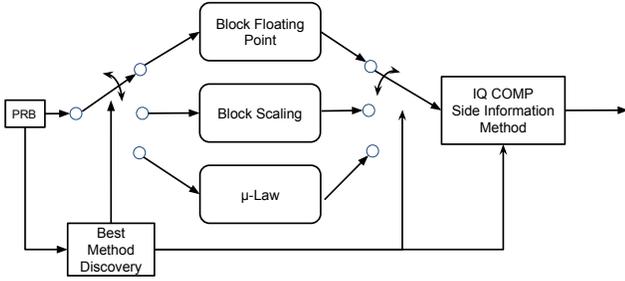


Figure 2: Overview of the compression structure with the choice of the best method.

Side information					Current O-RAN Specification							
Compression method	name	minimum	maximum	number of bits	udCompParam							
Block Floating Point	exponent	0	15	4 bits	B7	B6	B5	B4	B3	B2	B1	B0
Block Scaling	scaler	0	255	8 bits								
u-Law	compshift	0	7	3 bits	compBitWidth							

Side information					Proposed Frame Format							
Compression method	name	minimum	maximum	number of bits	udCompParam							
Block Floating Point	exponent	0	15	4 bits	B7	B6	B5	B4	B3	B2	B1	B0
Block Scaling	scaler	0	127	7 bits	1							
u-Law	compshift	0	7	3 bits	0	1	0	0	0			

Figure 3: Current *udCompParam* format [8] and proposed format.

A. Frame format for each compressed PRB

The O-RAN IQ frame format has two fields related to the compression methods: *udCompHdr* and *udCompParam*. The former specifies the compression method to be used for a group of PRBs and the corresponding *iqWidth*. The latter is an 8-bit field that specifies the side information for each PRB, which the decoder uses to recover the original representation of the PRBs [8], [9].

As shown in Section II, the side-information is the *exponent*, *compShift*, and *scaler8bits* for BFP, μ -Law, and BS methods, respectively. The exponent is a number between 0 and 15, and the *compShift* is a value between 0 and 7, so the former requires 4 bits and the latter needs 3 bits, respectively. The side information for BS uses all 8 bits reserved to it.

In order to compress each PRB independently, we propose to inform the used compression method as second side information. This can be sent along with the original side information of each method. In this case, the *scaler8bits* should be reduced to 7 bits, and no changes are required for the *exponent* and *compShift*. The difference in the frame format between the current specification and the proposed frame format is shown in Fig. 3, where the filled cells represent the bits used to inform *exponent*, *scaler8bits*, or *compShift*.

The proposed format bits B7 and B6 of *udCompParam* represent which method was used to compress the PRB. In this case, B7 and B6 equal to 00, 1X, and 01 correspond to BFP, BS, and μ -Law, respectively, where "X" means do not care.

There are two differences in the new format: i) the compBitWidth is not transported in the new format for μ -Law compression because it is equivalent to the *iqWidth* that is

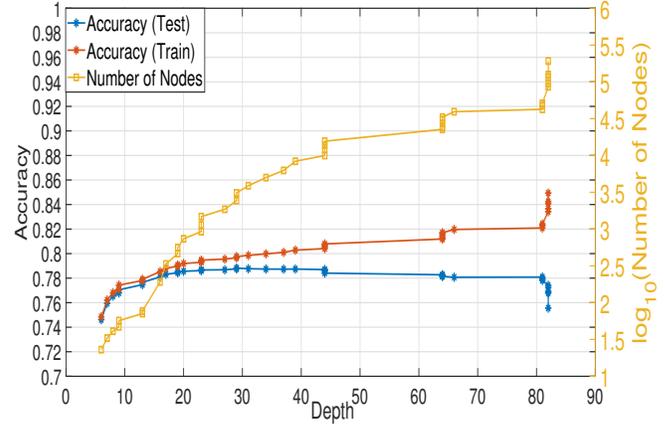


Figure 4: Accuracy and number of nodes in the decision tree versus depth.

already informed on *udCompHdr* field [8]; and ii) the side information for BS method is reduced from 8 to 7 bits. The latter can decrease the performance of the BS method in some cases, but it is not so significant when we select the best method for each PRB, as will be shown later.

B. Selection of the best method

Based on the proposed implementation shown in Fig. 2, the encoder should determine the method with the lowest distortion. Three approaches are investigated below: minimum mean squared error (MMSE), decision tree, and heuristic based on Dynamic Range (DR).

The MMSE approach consists of encoding and decoding the PRBs with all methods, calculating the mean squared error (MSE) between PRB and *dPRB* for each method, and selecting the method that produces the lowest MSE. Naturally, this brute force method requires a high computational cost, but it gives an upper bound for the SQNR with the three available compression methods.

On the other hand, the decision tree consists of using parameters to estimate the best compression method. In this case, the parameters considered estimating the best method are *iqWidth*, *maxValue*, and *dynamic range*. The *maxValue* is the maximum absolute value among all real and imaginary components of the PRB, and the *dynamic range* is the difference between the *maxValue* and the minimum absolute value among all real and imaginary components of the PRB.

A predictor based on classification and regression tree (CART) algorithm [12] is evaluated based on the following predictor variables: *iqWidth*, *maxValue* and *dynamic range*. In this case, a dataset is created with the same signals described in Section III, where the variables are defined for each PRB, and the best method is found with the MMSE approach described before. Then, the dataset is used to train the tree with the CART algorithm.

One critical parameter for the performance of a decision tree is the tree *depth*, i.e., the maximum distance between the root and a leaf node. Based on empirical data, the performance of the decision tree has a strong correlation to its depth. The

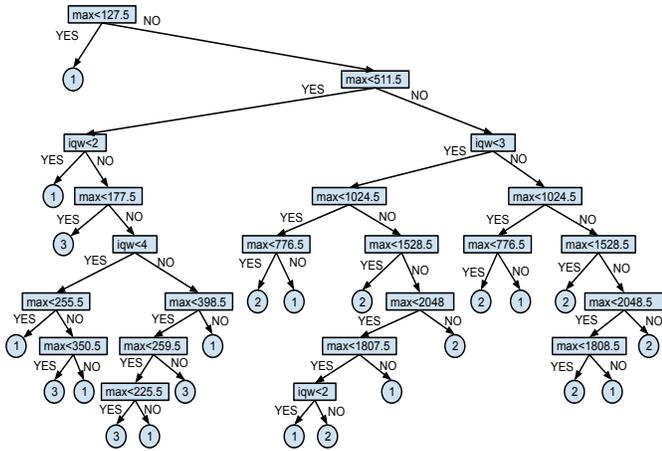


Figure 5: Binary decision tree with depth equal to 8, after using CART algorithm and applying pruning. The numbers 1,2, and 3 on the leaves indicate that the predicted best compression method is BFP, BS and μ -Law, respectively.

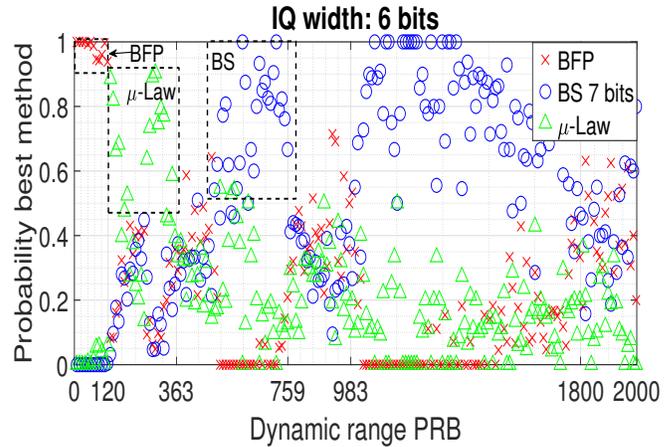


Figure 6: Probability of the best method for each value of dynamic range.

decision network generated for the data evaluation showed a maximum depth equal to 76. However, Fig. 4 shows that, for the same network, the best results in terms of accuracy were achieved around the depth values of 25 to 35. The interval between 25 and 35 showed better performance because, for low values, the tree does not have enough parameters to be well trained, and, for high values, overfitting occurs in the network.

The method used to decrease the tree depth is known as *pruning*. This tool allows a decrease in the number of operations used on the network, contributing to or hindering the decision tree's performance depending on the generated network. Fig. 5 shows the result of the decision tree generated for the respective dataset with a depth equal to 8.

The heuristic DR consists of a decision tree method based on *dynamic range*. To elaborate this method, we verify that all O-RAN compression methods use the *maxValue*, and it is a natural choice to estimate the best compression method with low complexity. For instance, depending on its value, the BFP method can compress the PRB without losses, as described in Section II-A. Then, inspired by the behavior of the BFP method, the same signals used in Section III were simulated with all compression methods for *iqWidth*=6 bits, and the probability of the methods show the lowest MSE was evaluated for a dynamic range between *maxValue* and minimum absolute value. In this case, the BS method was simulated with a side-information of 7 bits to use a compression method for each PRB, as described in Section IV-A. Then, the heuristic DR method was developed based on the calculated probabilities. Fig. 6 shows the probability results, and Fig. 7 shows the flowchart tree, which represents the heuristic method designed with few nodes to keep the complexity low. The *T* in Fig. 7 corresponds to the difference between *maxValue* and *minValue*. The values used in the decision nodes correspond to the limits at which a method is most likely to be the best, as shown in Fig. 6.

In Fig. 6, the BFP shows a probability of almost 1 when the

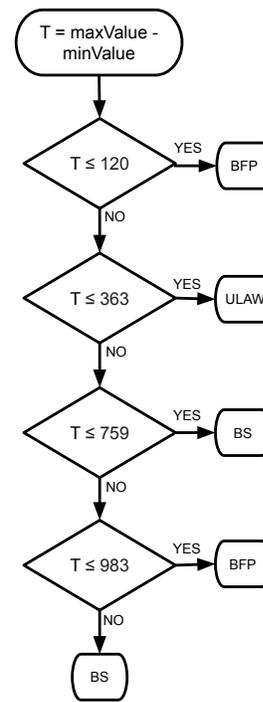


Figure 7: Heuristic tree based on the dynamic range to estimate the best compression method for each PRB.

dynamic range is lower than 120. Then, for a dynamic range between 120 and 363, there are many points where the μ -Law method shows high probabilities (> 0.6). Another range that presents a predominant best method is between 363 and 759, where the BS method has a probability higher than 0.6 of being the best method, and the range between 759 and 983, where the BFP method has a higher probability of being the best method. However, there are many values of dynamic range where there is no predominant best method, for example, when the dynamic range is between 1800 and 2000.

The computational cost for the MMSE method consists of

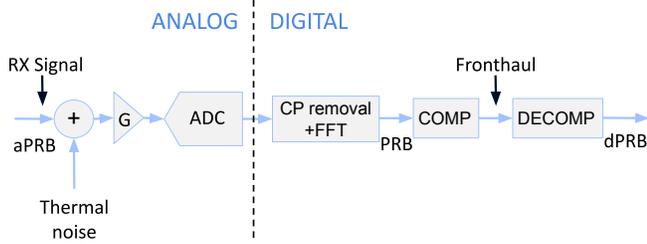


Figure 8: Baseband LTE receiver chain processing.

the total operations in compression and decompression of all three methods plus the mean squared error of each output in relation to the input for each compression method, where the mean squared error is calculated over 12 elements of the PRB. Then, the minimum is found, which corresponds to the best method. However, the cost of the other two methods are much lower, since they are based on conditional operations, where the binary decision tree based on CART algorithm uses 8 conditional operations in the worst case, and the heuristic tree uses 4 conditions in the worst case.

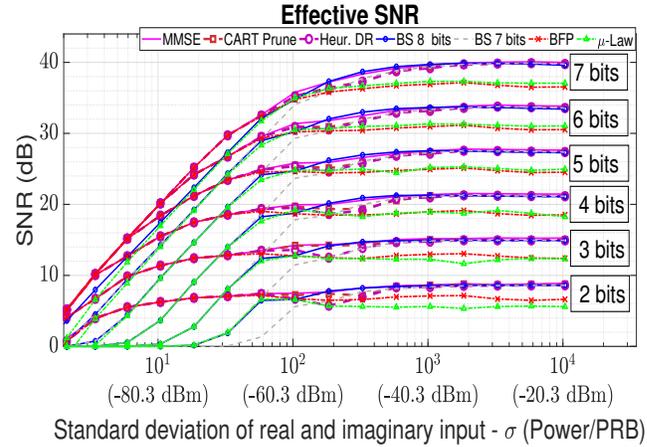
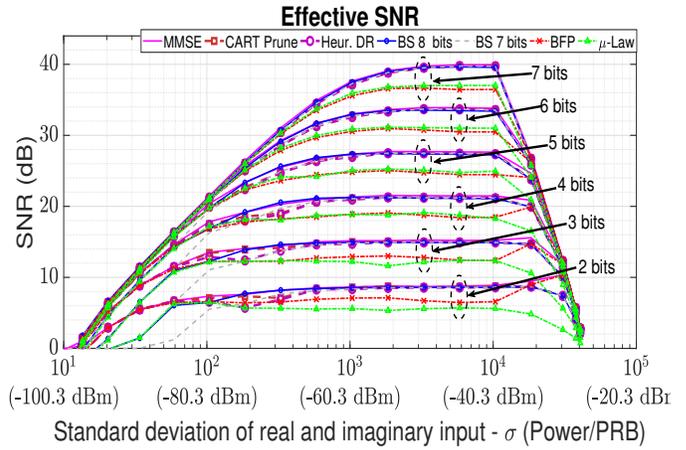
V. PERFORMANCE OF THE PROPOSED COMPRESSION APPROACH

The proposed compression approach was evaluated considering an LTE baseband receiver chain with additive thermal noise, a programmable gain amplifier, and an analog-digital converter (ADC). Then, in the digital domain, the frequency representation of the signal is recovered, and the compression and decompression are performed, as shown in Fig. 8. We assume a lossless fronthaul link.

Thermal noise, ADC quantization, and compression distortions were the main parameters taken into account to simulate the distortions in the uplink signal. Fig. 8 shows the complete chain of the LTE signal processing up to the output of the decompression unit. The RX signal corresponds to the attenuated signal in receiver input, added to the thermal noise, and amplified by the programmable-gain amplifier (PGA). The PGA gain G can be adjusted to change the dynamic range of the receiver. The ADC quantizes the amplified signal originating digital samples represented with 16 bits. Then, the PRBs are recovered based on the digital signal by removing the cyclic prefix (CP) and implementing the fast Fourier transform (FFT). The standard deviation of the PRBs is represented with σ . Finally, the PRBs are compressed and decompressed. In Fig. 8, the aPRB and dPRB correspond to the representation of the attenuated PRB on the input of the receiver chain and the decompressed PRB, respectively.

Based on the above receiver chain, the effective signal-to-noise ratio (SNR_{eff}) is defined as the ratio between the power of aPRB and the power of the overall noise (aPRB-dPRB), i.e., $\text{SNR}_{\text{eff}} = \mathbb{E} [|\text{aPRB}|^2] / \mathbb{E} [|\text{dPRB} - \text{aPRB}|^2]$. This metric was used to calculate the performance of the MMSE method in the LTE processing chain. It allows evaluating the performance of the entire receiver chain, including uplink signal compression.

The PGA gain directly influences the digital signal power. A higher gain provides improved resolution above the ADC


 Figure 9: Effective SNR versus σ (power of the input signal) of the methods for different values of $iqWidth$ and $G = 40$ dB.

 Figure 10: Effective SNR versus σ (power of the input signal) of the methods for different values of $iqWidth$ and $G = 60$ dB.

quantization noise but results in lower maximum power values allowed in the receiver. In contrast, a lower gain allows higher power in the receiver but lower resolution.

As shown in Fig. 1, the performance of the compression methods (especially for BS and μ -Law) depends on σ . The greater σ , the higher is the SQNR of the BS and μ -Law compression methods. Thus, the PGA gain and the power of aPRB influence the performance of each compression method because σ is proportional to them. Therefore, the effective SNR was evaluated for different configurations of the PGA gain and for different signal power levels to verify the performance for several cases on the receiver chain.

We first analyzed the performance of the proposed methods for a single UE, whose uplink signal was created with Reference Measurement Channel (RMC) A3-5 defined in [13] with a bandwidth of 10 MHz (50 PRBs) and Quadrature Phase Shift Keying (QPSK) modulation. The power range of the received signal was -100 dBm (-117 dBm/PRB) to -20 dBm (-37 dBm/PRB), and the power of the thermal noise in the receiver is assumed to be $P_n = -174 + \text{NF} + 10 \times \log_{10}(F_s)$, where -174 is the PSD in dBm per Hz, $\text{NF} = 3$ dB is the

noise figure and $F_s = 15.36 \times 10^6$ is the sampling frequency adopted in the simulation. The parameters of the ADC are the resolution of $b = 16$ bits to quantize a range of $[-1, 1]$, which results in a quantization step size of $\Delta = 2/2^b$.

Fig. 9 and Fig. 10 show the performance of the compression methods for PGA gain $G = 40$ and 60 dB, respectively. Both figures show the performance for one UE in terms of SNR_{eff} versus standard deviation of the real and imaginary input (σ) and SNR_{eff} versus the power of the received signal in dBm. Depending on the adopted gain, the receiver's dynamic range can vary and influence the values of SNR_{eff} . For example, with $G = 40$ dB, the range of the power is approximately from -100 dBm to -20 dBm, and for $G = 60$ dB, the range is about from -120 dBm to -40 dBm. In Fig. 10, SNR_{eff} drops quickly when the input power becomes higher than -40.3 dBm ($\sigma > 10^4$) due to saturation on the ADC.

Furthermore, the same σ can be obtained with different received power, depending on G ; for instance, $\sigma = 10^2$ is obtained with received power equal to -60.3 dBm and -80.3 dBm gain of 40 and 60 dB, respectively. Recall from Fig. 1 that the performance of the compression methods depends on σ , the gain of the analog part influences σ and, consequently, the performance of the compression methods.

The simulation was conducted for different $iqWidth$ values: from 2 to 7 bits. In Fig. 9 and 10, there are seven curves for each $iqWidth$. Three curves represent the proposed compression approach, where the best compression algorithm is chosen for each PRB based on MMSE, CART, or heuristic, as shown in Section IV. The other four curves represent the compression of all PRBs using a single compression algorithm: Block Scaling with 8 bits of side information (BS 8 bits), Block Scaling with 7 bits of side information (BS 7 bits), Block Floating Point (BFP) and μ -Law.

Fig. 9 shows that the MMSE method presents a higher SNR for all cases when compared to each method individually, and the other two proposed methods present higher SNR for most cases. For low values of σ , these methods can present SNR about 10 dB higher than the BS or μ -law, for instance, when $\sigma = 10$ and $iqWidth = 4$ or 3 bits. Also, for higher values of σ , the methods can present improved SNR, 3 dB higher when compared to the BFP and the μ -Law methods. Similarly, in Fig. 10, the proposed methods also have a better performance than the other methods in the most intervals analyzed.

To clarify the improvement of the proposed methods, we use as reference the SNR_{eff} achieved with the MMSE to select the best compression algorithm for each PRB. Then, Fig. 11 and Fig. 12 show the average differences between the reference and each compression method for $G = 40$ and 60 dB, respectively. The average is taken over the $iqWidth$ values shown in Fig. 9 and Fig. 10, and is given versus σ and the power of the attenuated signal.

In Fig. 11, the improvement of using the proposed approach based on MMSE can be as significant as 8 dB around $\sigma = 10^1$ when compared to BS 8 bits and μ -Law. When σ becomes higher, the performance of block scaling increases, and the SNR_{eff} of BFP decreases. At $\sigma = 10^4$, the MMSE approach shows an average $\text{SNR}_{\text{eff}} \approx 3$ dB higher than the BFP. In higher σ values, the performance of the MMSE method and

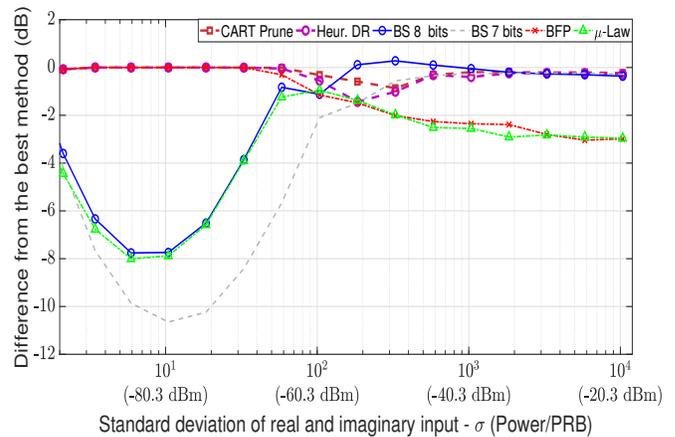


Figure 11: Average SNR_{eff} difference between the proposed method and each O-RAN compression method, when $G = 40$ dB.

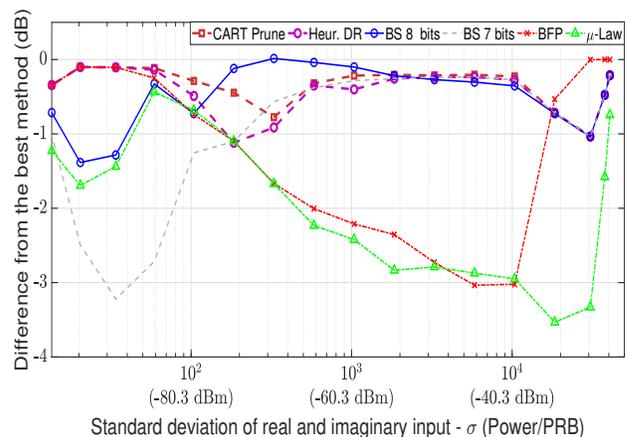


Figure 12: Average SNR_{eff} difference between the proposed method and each O-RAN compression method, when $G = 60$ dB.

BS 8 bits are similar. Actually, for $\sigma = 2 \times 10^2$, the performance of the BS 8 bits is slightly higher than the MMSE method. This happens because the MMSE approach uses the version of Block Scaling with 7 bits of side information (BS 7 bits), which is worse than BS 8 bits. The CART and heuristic methods have performance similar to the MMSE approach. The difference between CART/heuristic methods and MMSE is small throughout the analyzed σ range in Fig. 11, where the most significant difference is around 1.5 dB in $\sigma = 10^2$ for the heuristic method and 1 dB in $\sigma = 5 \times 10^2$ for the CART method.

In Fig. 12, where $G = 60$ dB, the improvements of using the MMSE method are decreased when compared to $G = 40$ dB. For example, compared to BS 8 bits, the maximum improvement of using the proposed compression approaches is about 1 dB when $\sigma = 2 \times 10^1$. This happens because most of the distortion is dominated by thermal noise since input power is lower for the higher gain $G = 60$ dB. For higher values of σ , the behavior is similar to when $G = 40$ dB, where

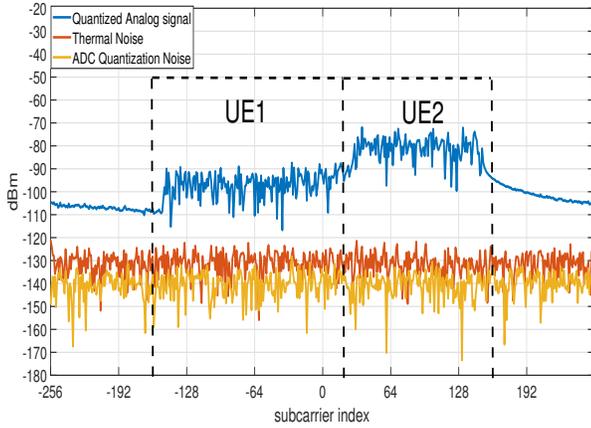


Figure 13: Power spectral density of the uplink received signal with 2 UEs with parameters of the Scenario 3 in Table III.

the improvement of choosing the best method for each PRB achieves a $SNR_{eff} \approx 3$ dB higher at $\sigma = 10^4$. The difference remains low compared to the other two methods proposed during all analyzed intervals with a maximum difference around 1 dB to σ between 2×10^1 and 5×10^1 .

In summary, the proposed compression approach that selects the best compression algorithm for each PRB based on MMSE shows the best SQNR, regardless of σ . The CART tree has a lower computational cost but provides a similar SQNR to the MMSE case. Also, the proposed heuristic tree has lower complexity and shows similar performance for most cases. Fig. 9, 10, 11, and 12 show that the effective SNR can be impacted by the receiver chain and the compression method. The proposed approaches for selecting the best method for each PRB bring improvements on SNR_{eff} compared to the O-RAN specification that uses a single compression method for a set of PRBs. This is particularly important when the RRU receives signals from multiple UEs simultaneously multiplexed in frequency with distinct attenuation levels.

The simulations were also conducted with two users (UE1 and UE2) to show the advantage of using the proposed methods when there are signals with different power levels in the uplink receiver. The thermal noise and ADC parameters are the same as those previously described for simulation with a single UE. UE1 and UE2 transmit LTE uplink signals simultaneously in different subcarriers, following RMC A3-4 and A4-5, respectively. Then, the SNR_{eff} was analyzed for each compression method, considering distinct scenarios with different power levels per UE and the number of PRBs per UE, as shown in Table III.

For example, Fig. 13 shows the power spectral density (PSD) of the uplink signal in Scenario 3 of Table III. The subcarrier spacing is 15 kHz; UE1 uses the subcarriers from indexes -160 to 30 (15 PRBs) and presents -81.8 dBm/PRB at the uplink receiver. In contrast, UE2 uses the subcarriers from indexes 31 to 160 (10 PRBs) and presents -65 dBm/PRB. Furthermore, Fig. 13 also shows the PSD of thermal and ADC noise.

In Scenario 1 of Table III, each IQ sample was compressed

Table III: Performance analysis of the LTE processing chain for the BS, BFP and μ -Law compression methods and the proposed method for two UEs in different scenarios.

	Scenario 1		Scenario 2		Scenario 3	
	UE1	UE2	UE1	UE2	UE1	UE2
UE	4		6		7	
IQ Width	4		6		7	
PGA gain	60 dB		40 dB		40 dB	
RX signal power (dBm)	-90	-75	-80	-65	-70	-55
Number of PRBs	15	10	15	10	15	10
σ	29.7	46.3	19.5	36.3	29.5	46.3
SNR (dB)	13.7	30.4	21.7	38.5	31.8	48.5
without compression						
SNR _{eff} (dB)	10.5	20.1	14.1	29.8	27.5	37.9
with BS						
SNR _{eff} (dB)	12.3	18.4	21.7	29.8	31.4	36.3
with BFP						
SNR _{eff} (dB)	10.3	18.9	13.9	29.4	27.5	36.7
with μ -Law						
SNR _{eff} (dB)	12.4	19.8	21.7	30.5	31.4	37.7
with MMSE						
SNR _{eff} (dB)	12.3	19.1	21.7	30.5	31.4	36.7
with CART prune						
SNR _{eff} (dB)	12.3	18.7	21.7	30.4	31.4	36.6
with heuristic DR						

with $iqWidth$ equal to 4 bits, the power signal of UE1 and UE2 are -90 and -75 dBm, resulting in $\sigma = 29.7$ and 46.3, respectively. The SNR_{eff} is shown for each O-RAN compression method and also for the proposed approaches. In Scenario 1, when we compare only the O-RAN compression methods, the BFP is the best method for UE1, while BS is the best method for UE2. If the current O-RAN specification were applied, only one compression method could be used for both UEs, which would not simultaneously provide the best performance for both UEs. On the other hand, when the proposed methods are applied, the best performance of both UEs are achieved simultaneously.

Similar behavior can be observed for other configurations of IQ width, PGA gains, and power levels, as shown in Scenarios 2 and 3. In these cases, the MMSE method guarantees the best SNR_{eff} for both UEs, except for UE2 of scenario 3 where BS presents a gain of ≈ 0.2 dB above the MMSE. The other two proposed methods (CART and Heuristic DR) also present the best performance, except also for UE2 of Scenario 3, where BS presents a gain of ≈ 1 dB above the two methods. In contrast, each O-RAN compression method (BFP, BS, and μ -Law) individually does not provide the best SNR_{eff} for both UEs.

More specifically, in Scenario 2, the BFP is the best O-RAN compression method for both UEs, but the implementation of the proposed methods (MMSE, CART or Heuristic DR) brings improvements on the SNR_{eff} of UE2, in about 0.7 dB. On the other hand, in Scenario 3, among the O-RAN compression methods, the BFP provides the best SNR_{eff} for UE1 and BS provides the best SNR_{eff} for UE2, while the MMSE method provides almost the same performance for both UE1 and UE2.

Therefore, the proposed methods provide better performance than the current O-RAN specification in terms of SNR_{eff} , specially for cases where the power of the PRBs are very different.

VI. CONCLUSION

This article presented the compression methods of the O-RAN specification to transport IQ samples through the fronthaul efficiently. The performance and computational cost of the methods were verified. It was found that BFP compression presents the best SQNR for low values of σ , and the BS compression has the best SQNR for high values of σ . This distinction is important because the uplink signals might have very different power levels on RRU, depending on the channel experienced by each UE. In terms of computational cost, it was identified that BFP compression has the lowest complexity, followed by BS and μ -Law compressor. Finally, we propose compressing each PRB with the best compression algorithm, regardless of the power levels. This approach achieves improved performance than what is possible with O-RAN, which applies the same compression method on all PRBs on a given cell sector.

REFERENCES

- [1] M. Masoudi, S. Lisi, and C. Cavdar, "Cost-Effective Migration Toward Virtualized C-RAN With Scalable Fronthaul Design," *IEEE Systems Journal*, vol. 14, no. 4, pp. 5100–5110, 2020, doi: 10.1109/JSYST.2020.2982428.
- [2] L. Gavrilovska, V. Rakovic, and D. Denkovski, "From Cloud RAN to Open RAN," *Wireless Personal Communications*, pp. 1–17, 2020, doi: 10.1007/s11277-020-07231-3.
- [3] A. Checko, H. L. Christiansen, Y. Yan, L. Scolari, G. Kardaras, M. S. Berger, and L. Dittmann, "Cloud RAN for Mobile Networks—A Technology Overview," *IEEE Communications Surveys Tutorials*, vol. 17, no. 1, pp. 405–426, 2015, doi: 10.1109/COMST.2014.2355255.
- [4] L. M. P. Larsen, A. Checko, and H. L. Christiansen, "A Survey of the Functional Splits Proposed for 5G Mobile Crosshaul Networks," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, pp. 146–172, 2019, doi: 10.1109/COMST.2018.2868805.
- [5] Peng, Mugen and Sun, Yaohua and Li, Xuelong and Mao, Zhendong and Wang, Chonggang, "Recent advances in cloud radio access networks: System architectures, key techniques, and open issues," *IEEE Communications Surveys Tutorials*, vol. 18, no. 3, pp. 2282–2308, 2016, doi: 10.1109/COMST.2016.2548658.
- [6] S. Lagén, L. Giupponi, A. Hansson, and X. Gelabert, "Modulation compression in next generation ran: Air interface and fronthaul trade-offs," *IEEE Communications Magazine*, pp. 3–4, 2020, doi: 10.1109/MCOM.001.2000453.
- [7] C. Nahum, L. Ramalho, A. Klautau, E. Medeiros, I. Almeida, and E. Trojer, "Functional Split and Frequency-Domain Processing for Fronthaul Traffic Reduction," *IEEE Communications Letters*, vol. 25, no. 8, pp. 2758–2762, 2021, doi: 10.1109/LCOMM.2021.3083102.
- [8] O-RAN Alliance, "Control, User and Synchronization Plane Specification," *O-RAN Fronthaul Working Group, ORAN-WG4.CUS.0-v03.00*, 2020.
- [9] A. Umesh, T. Yajima, T. Uchino, and S. Okuyama, "Overview of O-RAN Fronthaul Specifications," *NTT DOCOMO Technical Journal*, vol. 21, no. 1, pp. 46–59, 2019.
- [10] O-RAN Alliance, "O-RAN: Towards an Open and Smart RAN," *White Paper, October*, 2018.
- [11] S. Abeta, T. Kawahara, A. Umesh, and R. Matsukawa, "O-RAN Alliance Standardization Trends," *NTT DOCOMO Technical Journal*, vol. 21, no. 1, pp. 38–45, 2019.
- [12] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and regression trees*. Wadsworth Publishing Company, 1984, doi: 10.1201/9781315139470.
- [13] 3GPP, "Evolved Universal Terrestrial Radio Access (E-UTRA); Base Station (BS) Radio Transmission and Reception." *3GPP TS 36.104 version 14.3.0 Release 14*, 2017.



Marcos Silva was born in Belém, Brazil, in 1997. He received his BSc. in Telecommunications Engineering from Federal University of Pará (UFPA) in 2022. His research interests include digital signal processing and coding, information theory and massive MIMO.



Leonardo Ramalho (Member, IEEE) received the B.Sc. degree in Computer Engineering and the M.Sc. and the Ph.D. degrees in Electrical Engineering from the Federal University of Pará (UFPA), Belém, Brazil, in 2011, 2013 and 2016, respectively. Since 2011 he has been with LASSE-5G and IoT Research Group. Currently, he is a professor in the department of Telecommunications and Computer Engineering of the UFPA. His current research interests include embedded systems, IoT, digital signal processing, fronthaul technologies and digital communications.



Igor Almeida received the B.Sc. degree in Computer Engineering and the M.Sc. degree in Electrical Engineering from the Federal University of Pará (UFPA), Belém, Brazil, in 2010 and 2013, respectively, where he is currently pursuing the Ph.D. degree in Electrical Engineering. He has been with the Ericsson Research since 2016, involved with topics, such as 5G, fronthaul networking, synchronization, and wireless communications.



Eduardo Medeiros received the M.Sc. degree in Electrical Engineering from the Federal University of Pará, Belém, Brazil, in 2010, and the Licentiate and Ph.D. degrees from Lund University, Lund, Sweden, in 2015 and 2018, respectively. He is currently employed as a Senior Researcher with the Ericsson Research, Stockholm, Sweden. His current research interests include signal processing for broadband communications and fronthaul for 5G systems. He was a co-recipient of two best paper awards from the IEEE International Conference on Communications and three best paper awards from the IEEE Communications Society's Transmission, Access, and Optical Systems Technical Committee.



Miguel Berg received B.S. degrees in Electrical Engineering and Computer Science from Mid Sweden University, Sweden, in 1995, followed by Licentiate and Ph.D. degrees in wireless communication systems from the Royal Institute of Technology (KTH), Sweden, in 1999 and 2002 respectively. As a Principal Researcher at Ericsson Research, Stockholm, Sweden, he is currently leading research on lower layer split (LLS) for 4G/5G RAN, participating in O-RAN Open Fronthaul standardization, and is involved in 6G LLS research. Since 2012 he has also made significant contributions to research and development for the Ericsson Radio Dot System, targeting indoor small cells. Before joining Ericsson in 2007, he held research positions at Lund University (2006–2007) and KTH (2004–2006) and an engineering position at Radio Components Sweden AB (2002–2003). He has co-authored about 40 papers in peer-reviewed journals and conferences, and around 100 patent applications.



Aldebaro Klautau received the bachelor (Universidade Federal do Pará, UFPA, 1990), M. Sc. (Universidade Federal de Santa Catarina, UFSC, 1993) and Ph. D. degrees (University of California at San Diego, UCSD, 2003) in Electrical Engineering. Since 1996, he has been with UFPA and is now full professor, the ITU Focal Point, and directs LASSE. He was a visiting scholar at Stockholm University, UCSD and The University of Texas at Austin. He is a researcher of the Brazilian National Council of Scientific and Technological Development (CNPq),

a senior member of the IEEE and senior member of the Brazilian Telecommunications Society (SBTrT). His work focuses on machine learning and signal processing for communications and embedded systems.