Scalable User-Centric Distributed Massive MIMO Systems with Restricted Processing Capacity

Marx M. M. Freitas, Daynara D. Souza, A. L. P. Fernandes, Daniel Benevides da Costa, *Senior Member, IEEE,* André Mendes Cavalcante, *Member, IEEE,* Luca Valcarenghi, *Senior Member, IEEE,* and João C. Weyl Albuquerque Costa, *Senior Member, IEEE*

***Abstract*—This paper investigates the performance of scalable user-centric (UC) distributed massive multiple-input multiple-output (D-mMIMO) systems with multiple central processing units (CPUs), commonly called cell-free mMIMO. Specifically, a framework incorporating processing capacity and inter-CPU communication constraints is proposed. Two methods are presented for limiting the number of radio units (RUs) serving each user equipment (UE). The first method is performed by the CPUs, while the second one is implemented at the UEs and RUs. Both methods prevent the computational complexity (CC) for channel estimation and precoding signals from increasing with the number of RUs. The backhaul signaling demands are presented and modeled, and it is considered that each CPU can serve only a restricted number of UEs managed by other CPUs to mitigate inter-CPU communication. Two strategies to adjust the RU clusters according to the network implementations are also proposed. We compare the proposed approaches with a traditional scalable UC system. Simulation results reveal that the proposed techniques allow UC systems to keep their spectral efficiency (SE) under minor degradation while reducing the CC by 98% and improving energy efficiency (EE). Besides, managing inter-CPU communication controls backhaul traffic effectively, and RU cluster adjustments further reduce CC.**

***Index Terms*—Cell-free networks, computational complexity, multiple CPUs, RU selection, user-centric approach.**

I. Introduction

User-centric (UC) distributed massive multiple-input multiple-output (D-mMIMO) systems, also referred to as cell-free (CF) mMIMO, have been envisaged as one of the most promising technologies for future mobile communication networks (6G and beyond) [1]–[3]. In these systems, several radio units (RUs) are spread out in the coverage area, and the user equipment (UE) is served by a subset of RUs, called RU cluster, providing a more uniform service and a better coverage probability than cell-based systems due to the enhanced macro-diversity and reduction of RU-UE distances [4]–[9]. Despite the benefits, computational complexity (CC) can still be a drawback in these systems.

Several baseline solutions consider that the complexity of UC systems grows with the number of UEs and RUs, which is not practical [4], [5]. In this regard, [8]–[10] proposed a framework to provide scalability to UC systems. Essentially, it limits the number of UEs each RU can serve simultaneously. Consequently, the network resources (i.e., processing requirement, fronthaul/backhaul signaling, and total power) remain finite even if the number of UEs goes to infinity. The authors showed that scalable UC systems can still provide uniform coverage with negligible spectral efficiency (SE) losses compared to the case when the UEs are served by all RUs. The conclusions hold for both centralized and distributed network implementations. In the former, channel estimation and combining processing tasks are carried out on the central processing units (CPUs), while in the latter, they occur on the RUs. However, although the network resources become independent of the number of UEs, the signal processing complexity can still grow with the number of RUs [9]. For instance, the number of complex multiplications required to perform channel estimation and precoding remains proportional to the number of RUs serving the UE [8]. Thus, a more in-depth investigation into this topic is necessary, as the literature regularly assumes that there are more RUs than UEs in the network.

Another limitation inherent to UC systems is that the RU selection processes are not adapted to the network implementations. They generally only intend to improve some key points, such as effective channel gain [11], reduce pilot contamination [9], among others [12]–[14]. Consequently, RU clusters may benefit one implementation over another. For instance, RU clusters with a large number of RUs can degrade the energy efficiency (EE) and CC of UC systems operating in distributed implementation while they can improve the SE for the centralized ones.

Most of the strategies in the literature to enhance the network performance (e.g., SE and EE) also consider that a single CPU is responsible for coordinating the signals of all RUs. In other words, those existing strategies do not evaluate the negative impacts in UC systems when employing multiple CPUs, such as increased signaling on backhaul links. That

This work was supported in part by the Innovation Center, Ericsson Telecomunicações Ltda., Brazil; in part by the National Council for Scientific and Technological Development (CNPq); and in part by the project CLEVER (project number 101097560). The project is supported by the Key Digital Technologies Joint Undertaking and its members (including top-up funding by the Italian Ministry of Research and University (MUR)).

Marx M. M. Freitas, Daynara D. Souza, A. L. P. Fernandes, and João C. Weyl Albuquerque Costa are with the Applied Electromagnetism Laboratory, Federal University of Pará - UFPA, Belém, PA, 66075-110 Brazil (e-mail:{marx;daynara;andrelpf;jweyl}@ufpa.br).

Daniel Benevides da Costa is with the Department of Electrical Engineering, King Fahd University of Petroleum & Minerals (KFUPM), Dhahran 31261, Saudi Arabia (email: danielbcosta@ieee.org).

André Mendes Cavalcante is with Ericsson Research, Ericsson Telecomunicações Ltda., Indaiatuba, SP, 13337-300 Brazil (e-mail: andre.mendes.cavalcante@ericsson.com).

Luca Valcarenghi is with the Telecommunications, Computer Engineering, and Photonics Institute (TeCIP), Scuola Superiore Sant'Anna, Pisa, 56127 Italy (email: luca.valcarenghi@santannapisa.it).

A preliminary version of this paper was presented at the IEEE International Conference on Communications (ICC), 2023.

is, the CPUs may need to share signaling to serve the UE since the UE's RU cluster can comprise RUs connected to different CPUs, as illustrated in Fig. 1. This signaling demand is called inter-CPU communication or inter-CPU coordination [15]–[18]. Furthermore, the CC required to perform channel estimation and precoding is typically modeled only for a single CPU scenario, which cannot be directly applied to multiple CPUs. Therefore, deeper investigations into these topics are indeed necessary since state-of-the-art solutions rely on UC systems composed of multiple CPUs to efficiently divide the network processing tasks [9], [10].

A. Literature Review

The UC D-mMIMO literature has proposed several approaches to reduce network complexity under computational and signaling aspects [5]–[9]. For instance, [6] introduced the UC approach, demonstrating that UC D-mMIMO systems could achieve comparable performance to canonical D-mMIMO systems while reducing CC and fronthaul requirements. In [8], [9], the authors analyzed the scalability of D-mMIMO systems, presenting their performance in terms of SE for both centralized and distributed network implementations. The authors demonstrated that the CC of the network and signaling in the fronthaul links could be prevented from growing with the number of UEs, but they did not provide any analysis regarding the number of RUs. Moreover, [8], [9] claimed that their proposed strategies are effective for UC systems with multiple CPUs but did not detail the network's necessary signaling procedures and requirements to make it successful. That is, the authors in [8], [9] did not quantify the level of inter-CPU communication required by the network.

Strategies for reducing the number of RUs serving the UE were proposed in [12], [13]. Nevertheless, a mechanism to prevent network processing demands from growing with the number of RUs was not presented, i.e., the maximum number of RUs serving each UE was not restricted. In [19], the maximum number of RUs serving the UE was limited, defined as a parameter that can be adjusted to avoid losses in SE. However, the analysis did not account for the system's processing capacity limitation. In addition, a detailed investigation regarding CC and multiple CPUs was not provided.

In [15], an approach to mitigate inter-CPU communication was proposed. The authors considered a network composed of multiple virtual cells, each managed by an individual CPU. The UEs within a virtual cell are exclusively associated with the RUs inside that cell. Conversely, UEs at the cell edges can connect to RUs from different virtual cells (i.e., belonging to distinct CPUs). This approach reduced the effect of inter-CPU communication compared to traditional UC systems. Despite this advantage, the SE can decrease, while the signaling demands between CPUs still grow with the number of UEs.

In [16], the UE was initially connected to a primary CPU and subsequently linked to other CPUs, referred to as non-primary CPUs. The latter designates the UE as an inter-coordinated UE. This approach effectively controlled inter-CPU signaling while keeping the SE under minor degradation. To this end, it was considered that the number of inter-coordinated UEs that each CPU serves must be restricted. However, [16] did not quantify the backhaul signaling, and the evaluations focused only on the implementations of distributed processing. Regarding the adjustment of RU clusters under different network implementations, to the best of the authors' knowledge, no other works addressing this topic were found.

B. Contributions

This paper investigates the performance of scalable UC D-mMIMO systems by assuming that the CC to perform channel estimation and precoding signals does not grow with the number of RUs. In particular, it is considered a UC system where the UE is associated only with a finite number of RUs, i.e., the UE is connected only with the RUs having the strongest channel gains. To the best of the authors' knowledge, this is the first paper to propose an approach that limits the CC of UC systems from growing with the number of RUs. Moreover, a method is proposed to adjust the RU clusters according to the network implementation. The proposed method works in UC systems with and without processing capacity limitations, and it can be used as an alternative solution for reducing CC in UC systems without processing capacity limitations. As far as the authors are aware, this is also the first work that proposes a method for adjusting the RU clusters according to the network implementation in UC systems. Moreover, the work studies the feasibility of UC systems when the signaling requirements on backhaul links do not grow with the number of UEs, i.e., the inter-communication among CPUs is controlled. Overall, the main contributions of this paper can be summarized as:

- Two strategies for controlling the RU cluster size of UEs are proposed. The first one is conducted by the CPUs, while the second strategy is performed locally between UEs and RUs. Simulation results reveal that the proposed solutions allow the SE to be kept under minor degradation even if the CC is reduced by up to 98%. However, our results also demonstrate that the centralized implementation may require more processing capacity than distributed to avoid significant losses in the SE.
- Two methods for adjusting the RU clusters according to the network implementation are proposed. The results demonstrate that the proposed schemes can reduce CC and potentially increase EE.
- A framework is proposed to control the RU cluster size and manage signaling demands on backhaul links in each network implementation. Moreover, a model for the backhaul traffic is provided, accounting for data sharing, channel estimates, and precoding coefficients exchanged among CPUs.
- The CC is discussed by accounting for multiple CPUs, and the EE modeling is improved by considering the processing power consumption of various CPUs and backhaul links.

C. Paper Outline and Notations

The remainder of this paper is organized as follows. Section II presents the system model, including the channel estimation procedure, a framework for signaling requirements from multiple CPUs, and the downlink (DL) SE. Section III

presents the modeling of CC and EE. Sections IV and V introduce the proposed approaches to limit the network processing capacity and to perform RU cluster adjustment. Section VI plots illustrative numerical results and draws insightful discussions to reveal the effectiveness of the proposed approaches compared to prior baseline strategies. Finally, Section VII concludes the paper.

Notation: Boldface lowercase and uppercase letters denote vectors and matrices, respectively. The superscript $(\cdot)^{\mathrm{H}}$ denotes the conjugate-transpose operation, the $N \times N$ identity matrix is denoted as $\mathbf{I}_N$, and the cardinality of the set $\mathcal{A}$ is represented by $|\mathcal{A}|$. The trace, euclidean norm, and expectation operator are denoted as $\mathrm{tr}(\,.\,)$, $\|\,.\,\|$, and $\mathbb{E}\{\,.\,\}$, respectively. The notation $\mathcal{N}_{\mathbb{C}}\left(\mu, \sigma^2\right)$ stands for a complex Gaussian random variable with mean μ and variance σ^2.

II. System Model

We consider a D-mMIMO network composed of J CPUs, L RUs, and K single-antenna UEs, where $L > K$. Each RU is equipped with N antennas, resulting in a total of M antennas considering all RUs, with $M = NL$. The RUs are connected to the CPUs through fronthaul links, while the CPUs are interlinked through backhaul connections, as shown in Fig. 1. The fronthaul links undergo limited transmission capacity, while the backhaul ones are considered error-free and capable of supporting the data traffic. We utilize analog-to-digital converters (ADCs) to limit the data transmitted over the fronthaul links. Therefore, signals are quantized before being sent to the fronthaul. The system operates on time-division duplex (TDD) mode and it is assumed that the uplink (UL) and DL channels are reciprocal. Thus, channel estimation is performed only in the UL direction. We focus on DL transmissions and consider that the channel $\mathbf{h}_{kl} \in \mathbb{C}^{N\times 1}$ between the RU l and UE k undergoes an independent correlated Rician fading, being defined as [20]–[22]

$$\mathbf{h}_{kl} = \underbrace{\sqrt{\frac{\kappa_{kl}}{1+\kappa_{kl}}}\mathbf{h}_{kl}^{\mathrm{LOS}}\, e^{j\theta_{kl}}}_{\overline{\mathbf{h}}_{kl}} + \underbrace{\sqrt{\frac{1}{1+\kappa_{kl}}}\mathbf{h}_{kl}^{\mathrm{NLOS}}}_{\widetilde{\mathbf{h}}_{kl}}, \quad (1)$$

where $\overline{\mathbf{h}}_{kl} \in \mathbb{C}^{N\times 1}$ means the deterministic line-of-sight (LOS) component, while $\widetilde{\mathbf{h}}_{kl} \sim \mathcal{N}_{\mathbb{C}}(\mathbf{0}_N, \widetilde{\mathbf{R}}_{kl}) \in \mathbb{C}^{N\times 1}$ stands for the small-scale fading with statistical covariance matrix[1] $\widetilde{\mathbf{R}}_{kl} = \mathbb{E}\{\widetilde{\mathbf{h}}_{kl}\widetilde{\mathbf{h}}_{kl}^{\mathrm{H}}\} \in \mathbb{C}^{N\times N}$. The term $\theta_{kl} \sim \mathcal{U}[0, 2\pi)$ denotes random phase shifts that may occur in LOS components due to the UEs mobility, and the Rician factor κ_{kl} is the power ratio between the LOS and non-line-of-sight (NLOS) components. The latter can be computed as $\kappa_{kl} = p_{\mathrm{LOS}}/(1 - p_{\mathrm{LOS}})$, with p_{LoS} being the probability of the LOS component's existence [24]. Furthermore, $\mathbf{h}_{kl}^{\mathrm{NLOS}} \sim \mathcal{N}_{\mathbb{C}}(\mathbf{0}_N, \mathbf{R}_{kl}^{\mathrm{NLOS}})$ represents the effects of the NLOS propagation.

Assuming that the RUs are equipped with half-wavelength-spaced uniform linear arrays (ULAs), the covariance matrix of the NLOS channel $\mathbf{h}_{kl}^{\mathrm{NLOS}}$, i.e., $\mathbf{R}_{kl}^{\mathrm{NLOS}}$, can be computed following the local scattering model for spatial covariance presented in [23, Sec. 2.6]. Thus, the covariance matrix of the term $\widetilde{\mathbf{h}}_{kl}$ in (1) is given by $\widetilde{\mathbf{R}}_{kl} = \mathbb{E}\{\widetilde{\mathbf{h}}_{kl}\widetilde{\mathbf{h}}_{kl}^{\mathrm{H}}\} = \mathbf{R}_{kl}^{\mathrm{NLOS}}/(\kappa_{kl}+1)$, which implies that $\mathbf{R}_{kl} = \mathbb{E}\{\mathbf{h}_{kl}\mathbf{h}_{kl}^{\mathrm{H}}\} = (\overline{\mathbf{h}}_{kl}\overline{\mathbf{h}}_{kl}^{\mathrm{H}} + \widetilde{\mathbf{R}}_{kl})$. Moreover, the LOS channel between the UE k and RU l can be expressed as [23]

$$\mathbf{h}_{kl}^{\mathrm{LoS}} = \sqrt{\beta_{kl}}\left[1, \cdots, e^{\mathrm{j}(N-1)\pi\sin(\varphi_{kl})\cos(\psi_{kl})}\right]^{\mathrm{T}}, \quad (2)$$

where φ_{kl} denotes the azimuth angle, ψ_{kl} is the elevation angle of the LOS component, and β_{kl} is the large-scale fading gain, which can be calculated as $\beta_{kl} = \mathrm{tr}(\mathbf{R}_{kl})/N$.

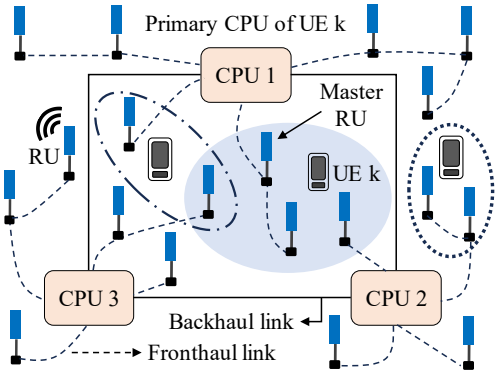


Fig. 1: UC D-mMIMO system with multiple CPUs. Each CPU is connected to a subset of RUs, and the UEs can be associated with RUs linked to different CPUs.

A. Network Implementations

UC D-mMIMO systems are commonly implemented in centralized or distributed manners according to processing capabilities. The centralized implementation places most baseband functions on the CPUs. Therefore, the CPUs are responsible for channel estimation and precoding [9]. Furthermore, they encode and quantize the DL signals. In the distributed implementation, essential processing functions, such as channel estimation, are moved to the RUs. Consequently, the CPUs are only responsible for encoding and quantization.

The centralized implementation usually offers superior interference mitigation since the CPUs can access global channel state information (CSI), which includes channel estimates and statistics. Conversely, the distributed one can be less complex and avoids the need to transmit the pilot signals on fronthaul links [7]. In this regard, several network procedures, such as channel estimation, interference mitigation, and CC, vary depending on the network implementation. In order to compute the combining and precoding vectors, this paper utilizes the partial MMSE (P-MMSE) and partial regularized zero-forcing (P-RZF) schemes for centralized implementation. For the distributed one, the local partial MMSE (LP-MMSE) and maximum ratio (MR) are utilized. These techniques have been chosen due to their scalability features [9].

B. Uplink Training and Channel Estimation

Each coherence block comprises τ_c samples, where τ_p samples are dedicated for UL pilot signals and τ_d for DL data. During the UL training phase, the UEs send pilot sequences

[1]The statistical covariance matrix represents the large-scale fading of the system, being a function of the spatial channel covariance, path loss, antenna gains, and shadowing [23].

of τ_p-length to the RUs for channel estimation. Then, the UL channels are estimated by correlating the received signals with a known pilot sequence and using phase-unaware linear minimum mean square error (LMMSE) estimation. The pilot signals are assumed to be mutually orthogonal and independent of the number of UEs K to ensure the scalability of the pilot resources. Thus, a pilot t can be reused by some UEs if the number of UEs is greater than the number of pilot signals, i.e., $K > \tau_p$. Let $\mathcal{P}_k \subset \{1, \ldots, K\}$ denote the subset of the UEs assigned to the pilot t, including the UE k. The received pilot signal at RU l can be expressed as [8]

$$\mathbf{y}_{tl}^{\text{pilot}} = \sum_{i \in \mathcal{P}_k} \sqrt{\tau_p \eta_i}\, \mathbf{h}_{il} + \mathbf{n}_{tl}, \tag{3}$$

where $\mathbf{n}_{tl} \sim \mathcal{N}_{\mathbb{C}}\left(\mathbf{0}_N, \sigma_{ul}^2 \mathbf{I}_N\right)$ denotes the noise and η_i is the power that the UE i transmits in the UL direction. The channel estimation procedure differs in each network implementation. In the distributed implementation, the channel vector $\mathbf{h}_{kl}$ is estimated locally by RU l after receiving the pilot signals $\mathbf{y}_{tl}^{\text{pilot}}$ sent by the UEs in (3). In the centralized one, the RUs forward the received pilot signals $\mathbf{y}_{tl}^{\text{pilot}}$ to the CPUs, which then perform the channel estimation. Note that due to the use of ADCs, the signals must be quantized before being sent to the fronthaul links. Therefore, in the centralized implementation, the received pilot signals $\mathbf{y}_{tl}^{\text{pilot}}$ are quantized before being sent to CPUs. In contrast, the quantization of pilot signals is not necessary for the distributed implementation, as channel estimation occurs locally in the RUs.

In the distributed implementation, the LMMSE channel estimation is given by [8]

$$\widehat{\mathbf{h}}_{kl} = \sqrt{\tau_p \eta_k} \mathbf{R}_{kl} \boldsymbol{\Psi}_{tl}^{-1} \mathbf{y}_{tl}^{\text{pilot}}, \tag{4}$$

where $\mathbf{R}_{kl} = \mathbb{E}\{\mathbf{h}_{kl}\mathbf{h}_{kl}^{\text{H}}\} = (\overline{\mathbf{h}}_{kl}\overline{\mathbf{h}}_{kl}^{\text{H}} + \widetilde{\mathbf{R}}_{kl})$ and $\boldsymbol{\Psi}_{tl} = \mathbb{E}\{(\mathbf{y}_{tl}^{\text{pilot}})(\mathbf{y}_{tl}^{\text{pilot}})^{\text{H}}\} = \sum_{i \in \mathcal{P}_k} \eta_i \tau_p (\overline{\mathbf{h}}_{il}\overline{\mathbf{h}}_{il}^{\text{H}} + \widetilde{\mathbf{R}}_{il}) + \sigma_{\text{ul}}^2 \mathbf{I}_N$. The term $\boldsymbol{\Psi}_{tl}$ denotes the covariance matrix of the received signal $\mathbf{y}_{tl}^{\text{pilot}}$. One can note that $\boldsymbol{\Psi}_{tl}$ also indicates the presence of pilot contamination since it contains the sum of the covariance matrices of all UEs sharing pilot t.

In the centralized implementation, the LMMSE channel estimation is computed as [25]

$$\widehat{\mathbf{h}}_{kl} = \sqrt{\tau_p \eta_k} \mathbf{R}_{kl} \widetilde{\boldsymbol{\Psi}}_{tl}^{-1} \widetilde{\mathbf{y}}_{tl}^{\text{pilot}}, \tag{5}$$

where $\widetilde{\mathbf{y}}_{tl}^{\text{pilot}} \approx \alpha_{\text{p},l} \mathbf{y}_{tl}^{\text{pilot}} + \boldsymbol{q}_t$ stands for the quantized pilot signal received on the CPU connected to RU l, while $\widetilde{\boldsymbol{\Psi}}_{tl} \approx \alpha_{\text{p},l}^2 \boldsymbol{\Psi}_{tl} + \alpha_{\text{p},l}\left(1 - \alpha_{\text{p},l}\right) \boldsymbol{\Psi}_{tl}$ denotes the covariance matrix of $\widetilde{\mathbf{y}}_{tl}^{\text{pilot}}$. The term $\alpha_{\text{p},l}$ represents a distortion factor associated with the number of bits b_l^{p} used to quantize the pilot signals, and $\boldsymbol{q}_t \sim \mathcal{N}_{\mathbb{C}}(\mathbf{0}_N, \mathbf{R}_{\boldsymbol{q}_t}) \in \mathbb{C}^{N \times 1}$ means the quantization noise with covariance $\boldsymbol{R}_{\boldsymbol{q}_t} = \alpha_{\text{p},l}\left(1 - \alpha_{\text{p},l}\right) \boldsymbol{\Psi}_{tl}$. Note that (5) degenerates to (4) when $\alpha_{\text{p},l} = 1$. The relationship between $\alpha_{\text{p},l}$ and b_l^{p} is obtained from [26], [27].

C. Downlink Data Transmission

In UC systems, each UE is associated with a subset of RUs called RU cluster. These clusters are intended to be dynamic and capable of adapting to variations in network conditions, such as UE position and channel properties. To represent the RU clusters of each UE, we proceed as follows: First, let $\mathcal{M}_k \subset \{1, \ldots, L\}$ represent the indexes of the RUs serving the UE k. Second, let vector $\mathbf{c}_k = [\text{c}_{k1}, ..., \text{c}_{kL}] \in \mathbb{N}^{1 \times L}$ denote the RUs that establish a connection with UE k, such that $\text{c}_{kl} = 1$ if the RU serves the UE k, and $\text{c}_{kl} = 0$ otherwise. Therefore, the connections between the UE k and RUs are expressed as

$$\text{c}_{kl} = \begin{cases} 1 & \text{if } l \in \mathcal{M}_k \\ 0 & \text{if } l \notin \mathcal{M}_k. \end{cases} \tag{6}$$

The matrix $\mathbf{D}_{kl} \in \mathbb{N}^{N \times N}$ is also utilized to describe which antennas of the RU l serve the UE k. It is assumed that all N antennas of RU l serve the UE k; thus $\mathbf{D}_{kl} = \mathbf{I}_N$ when $\text{c}_{kl} = 1$. Otherwise, $\mathbf{D}_{kl} = \mathbf{0}_N$. The vector $\mathbf{c}_{kl}$ can also be utilized to compute the number of UEs that RU l serves and the number of RUs serving the UE k. For instance, let $\mathcal{D}_l$ represent the indexes of the subset of UEs that RU l serves. The cardinalities of $\mathcal{D}_l$ and $\mathcal{M}_k$ can be computed as $|\mathcal{D}_l| = \sum_{k \in \mathcal{D}_l} \text{c}_{kl}$ and $|\mathcal{M}_k| = \sum_{l \in \mathcal{M}_k} \text{c}_{kl}$. These cardinalities can also be represented by L_k and K_l, where $L_k = |\mathcal{M}_k|$ and $K_l = |\mathcal{D}_l|$. It is noteworthy that scalable UC D-mMIMO systems usually assume that K_l is constrained to prevent it from being a function of the number of UEs. Thus, it is assumed that $K_l \leq \tau_p$ [9].

Once the RU clusters are formed, the network can proceed with the processes of channel estimation, precoding, and DL transmission for the UEs that each RU serves. Let $\mathbf{x}_l^{\text{dist}} = \sum_{i=1}^{K} \alpha_{il} \mathbf{D}_{il} \mathbf{w}_{il}\left(s_i + q_{il}\right)$ and $\mathbf{x}_l^{\text{cent}} = \alpha_l \sum_{i=1}^{K} \mathbf{D}_{il} \mathbf{w}_{il} s_i + \boldsymbol{q}_l$ denote the data signals sent by RU l in the distributed and centralized implementations. Here, $s_i \in \mathbb{C}$ is the unity-power symbol intended for UE i, and α_{il} and α_l represent the distortion factors associated with the number of bits used to quantize the DL signals in distributed and centralized implementations, respectively. Besides, $q_{il} \sim \mathcal{N}_{\mathbb{C}}(0, \sigma_{q_{il}}^2)$ and $\boldsymbol{q}_l \sim \mathcal{N}_{\mathbb{C}}(\mathbf{0}_N, \mathbf{R}_{\boldsymbol{q}_l}) \in \mathbb{C}^{N \times 1}$ are the quantization noises applied to the DL signals, with covariances $\sigma_{q_{il}}^2 = \alpha_{il}\left(1 - \alpha_{il}\right)$ and $\boldsymbol{R}_{q_l} = \alpha_l\left(1 - \alpha_l\right) \sum_{i \in \mathcal{M}_k} \mathbb{E}\left\{\mathbf{w}_{il}\mathbf{w}_{il}^{\text{H}}\right\}$. An achievable DL SE can be computed as [9], [25]

$$\text{SE}_k^{(\text{dl})} = \frac{\tau_d}{\tau_c} \log_2\left(1 + \frac{\text{DS}_k}{\text{IS}_k - \text{DS}_k + \text{QN}_k + \sigma_{dl}^2}\right), \tag{7}$$

where $\text{IS}_k = \sum_{i=1}^{K} \mathbb{E}\{\left|\sum_{l=1}^{L} \widetilde{\alpha}_{il} \mathbf{h}_{kl}^{\text{H}} \mathbf{D}_{il} \mathbf{w}_{il}\right|^2\}$ stands for the interference, $\text{DS}_k = \left|\sum_{l=1}^{L} \mathbb{E}\{\widetilde{\alpha}_{kl} \mathbf{h}_{kl}^{\text{H}} \mathbf{D}_{kl} \mathbf{w}_{kl}\}\right|^2$ denotes the desired signal, QN_k means the quantization noise, and σ_{dl}^2 is the receiver noise variance. One can note that $\widetilde{\alpha}_{il}$ is the distortion factor associated with the number of bits used to quantize the DL signals. Both $\widetilde{\alpha}_{il}$ and QN_k are computed differently in each network implementation. Thus, $\widetilde{\alpha}_{il} = \alpha_{il}$ and $\widetilde{\alpha}_{il} = \alpha_l$ for distributed and centralized implementations, respectively, whereas QN_k is given by

$$\text{QN}_k = \begin{cases} \mathbb{E}\left\{\left|\sum_{l=1}^{L} \mathbf{h}_{kl}^{\text{H}} \mathbf{D}_{kl} \boldsymbol{q}_l\right|^2\right\}, & \text{for CI} \\ \mathbb{E}\left\{\left|\sum_{l=1}^{L} \mathbf{h}_{kl}^{\text{H}} \sum_{i=1}^{K} \mathbf{D}_{il} \mathbf{w}_{il} q_{il}\right|^2\right\}, & \text{for DI.} \end{cases} \tag{8}$$

where CI and DI represent the centralized and distributed implementations, respectively. The term $\mathbf{w}_{il}$ is the precoding

vector generated to mitigate the inferences between the UE i and RU l. In the distributed implementation, $\mathbf{w}_{il}$ satisfies $\mathbb{E}\{\|\mathbf{w}_{il}\|^2\} = \rho_{il}$, with ρ_{il} being the power allocated to the UE i regarding the RU l. In the centralized one, the fraction of power assigned to $\mathbf{w}_{il}$ is calculated based on the collective precoding vector $\mathbf{w}_i = \left[\mathbf{w}_{i1}^{\mathrm{T}}, ..., \mathbf{w}_{iL}^{\mathrm{T}}\right]^{\mathrm{T}} \in \mathbb{C}^{M\times 1}$, which must satisfy $\mathbb{E}\{\|\mathbf{w}_i\|^2\} = \rho_i$, where ρ_i represents the transmit power assigned to UE i by all its serving RUs.

D. CPUs Requirements and Backhaul Signaling

In UC D-mMIMO systems involving multiple CPUs, it is essential to note that the RU cluster serving a specific UE may comprise RUs connected to different CPUs, as depicted in Fig. 1. Consequently, signal processing tasks must be efficiently distributed among these CPUs [7], [9], [10], [18]. This paper considers two classes of CPUs within the network to distribute the processing load among them more effectively. Specifically, one of the CPUs associated with the UE's RU cluster is designated as its primary CPU, i.e., the one responsible for the majority of its signal processing tasks. The remaining CPUs serving the UE are called secondary CPUs [16]. It is assumed that each UE is associated with a primary CPU, which is one of the CPUs belonging to the UE's RU cluster. Therefore, UEs may be assigned to different primary CPUs, since the UEs can be positioned in distinct positions in the coverage area.

This section fills a gap in the existing literature by providing a framework explaining how the CPUs should operate in each network implementation and highlighting the signaling requirements each implementation can impose on backhaul links. Moreover, a detailed modeling for the signaling demands on backhaul links, which considers the impacts of instantaneous CSI and data sharing, is also provided.

1) Centralized implementation: Each CPU belonging to the RU cluster of UE k estimates the channel vector $\widehat{\mathbf{h}}_{kl}$ for all its RUs serving the UE k. As for the primary CPU of UE k, it also computes the combining and precoding vectors of UE k. Hereinafter, the primary CPU of UE k will also be denoted as CPU j_k. It is considered that the primary CPU of UE k is the CPU connected to the RU with the strongest channel gain serving the UE k [16]. The procedure for associating a primary CPU with a UE is detailed in Subsection IV-A.

Regarding the signaling demands, the primary CPU of UE k requests other CPUs in the network to forward the channel estimates of the UE k and interfering UEs to compute the combining and precoding ($\mathbf{w}_k$) vectors. After generating $\mathbf{w}_k$, the CPU j_k sends $\mathbf{w}_k$ and s_k to the secondary CPUs. Then, the secondary CPUs of UE k quantize and forward the DL data s_k to their respective RUs. Let $\mathcal{J}_k^{\mathrm{sec}}$ denote the subset of secondary CPUs associated with the RU cluster of UE k. The number of complex scalars that the secondary CPUs have to exchange with the primary CPU of UE k via backhaul in each coherence block can be modeled as

$$\mathrm{BT}_k^{\mathrm{pri}} = \sum_{j\in\mathcal{J}_k^{\mathrm{sec}}} \left(2L_{jk}N + \tau_d\right), \tag{9}$$

where L_{jk} denotes the number of RUs in the secondary CPU j serving the UE k. Moreover, L_{jk} is multiplied by 2 in (9) to account for the transmission of channel estimates and precoding vectors over backhaul links. Similarly, the number of complex scalars exchanged via backhaul between a subset of CPUs and the primary CPU of UE k to transmit channel estimates of interfering UEs is given by

$$\mathrm{BT}_k^{\mathrm{int}} = \sum_{i\neq k, i\in\mathcal{I}_k} \sum_{j'\in\mathcal{J}_i} L_{j'i}N, \tag{10}$$

where $\mathcal{I}_k$ denotes the subset of interfering UEs affecting the signal of UE k and $\mathcal{J}_i$ represents the subset of CPUs associated with each interfering UE i, with $j' \neq j_k$. One can note that (10) may overestimate the backhaul traffic, as channel estimates from interfering UEs are sent redundantly to CPU j_k, i.e., they are transmitted for each UE utilizing CPU j_k as its primary CPU. However, there is no need for redundant transmission of channel estimates for these interfering UEs to CPU j_k since the UEs that use CPU j_k as their primary CPU may share the same interfering UEs. In this case, these channel estimates can be transmitted only once. Thus, (10) can be rewritten as

$$\mathrm{BT}_k^{\mathrm{int}} = \sum_{i\neq k, i\in\widetilde{\mathcal{I}}_k} \sum_{j'\in\mathcal{J}_i} L_{j'i}N, \tag{11}$$

where $j' \neq j_k$ and $\widetilde{\mathcal{I}}_k$ represents the subset of interfering UEs whose channel estimates have not yet been sent to CPU j_k. Hence, the total number of complex scalars exchanged on the backhaul links is calculated as

$$\mathrm{BT}_k^{\mathrm{cent}} = \mathrm{BT}_k^{\mathrm{pri}} + \mathrm{BT}_k^{\mathrm{int}}. \tag{12}$$

2) Distributed implementation: The primary CPU of UE k encodes the DL data. Then, it sends the DL data to the secondary CPUs. The latter quantizes and forwards s_k to their respective RUs for DL data precoding. Thus, the total number of complex scalars exchanged between the primary CPU of UE k and the secondary CPUs is given by

$$\mathrm{BT}_k^{\mathrm{dist}} = \sum_{j\in\mathcal{J}_k^{\mathrm{sec}}} \tau_d. \tag{13}$$

Remark 1: *The computation of* (9) *and* (11) *relies on the assumption that the CPUs share the RU clusters of the UEs with each other. For instance, CPU j reports which RUs connected to it serve the UE k. Thus, CPU j does not need to send $L \times N$ complex scalars (to share the estimated channel and precoding of UE k) to other CPUs. Instead, it only transmits $L_{jk} \times N$ elements. This assumption is reasonable as long as the formation of RU clusters is based on channel statistics. Hence, the signaling required to share the RU clusters across the backhaul links is negligible, as the channel statistics are constant throughout the data transmission*[2].

E. Reducing Inter-CPU Communication

To further reduce the number of complex scalars exchanged between CPUs, we rely on one of our previous works [16], which proposes a strategy to mitigate the inter-communication

[2]In practice, channel statistics can change due to UE mobility or scheduling. However, a deeper investigation into this topic is out of the scope of this paper.

between CPUs, also called inter-CPU coordination. Specifically, it is considered that the network has two classes of UEs. The first one comprises the UEs using the CPU j as a primary CPU, denoted as $\mathcal{K}_j^{\text{pri}}$ in this paper. The second one includes the UEs that utilize CPU j as a secondary CPU, represented by subset $\mathcal{K}_j^{\text{sec}}$. The approach proposed in [16] states that each CPU can serve only a limited number of UEs as a secondary CPU, denoted as K^{sec}. Thus, the CPUs associated with UE k can modify its RU cluster to meet this condition, i.e., $\left|\mathcal{K}_j^{\text{sec}}\right| \leq K^{\text{sec}}$. Let $\mathbf{e}_k = [\mathrm{e}_{k1}, \ldots, \mathrm{e}_{kL}] \in \mathbb{N}^{1\times L}$ represent the RU cluster of UE k. The CPUs modify $\mathbf{e}_k$ to

$$\mathbf{c}_k = \mathbf{e}_k \wedge \mathbf{f}_k, \tag{14}$$

where $\wedge$ is the logical operation AND. Besides, $\mathbf{f}_k = [\mathrm{f}_{k1}, \ldots, \mathrm{f}_{kL}] \in \mathbb{N}^{1\times L}$ is a fine-tuned version of the RU cluster of UE k, which can be expressed as

$$\mathrm{f}_{kl} = \begin{cases} 1 & \text{if } \left(k \in \mathcal{K}_j^{\text{pri}}\right) \vee \left(\left|\mathcal{K}_j^{\text{sec}}\right| < K^{\text{sec}}\right) \\ 1 & \text{if } \left(\left|\mathcal{K}_j^{\text{sec}}\right| = K^{\text{sec}}\right) \wedge \left(\mathrm{G}_{kj} > \mathrm{G}_{(i_{\text{min}})j}\right) \\ 0 & \text{otherwise} \end{cases}, \tag{15}$$

where $\vee$ is the logical operation OR. The term $\mathrm{G}_{kj} = \sum_{l\in\mathcal{L}_{kj}} \beta_{kl}$ represents the partial sum gain, with $\mathcal{L}_{kj}$ being the subset of RUs serving the UE k that are connected to CPU j, and i_{min} standing for the UE using the CPU j as a secondary CPU presenting the smallest partial sum gain. It is noteworthy that the UE i_{min} is dropped from all RUs connected to CPU j if $\mathrm{G}_{kj} > \mathrm{G}_{(i_{\text{min}})j}$. Note that (14) limits the number of UEs using CPU j as a secondary CPU, since $\left|\mathcal{K}_j^{\text{sec}}\right| \leq K^{\text{sec}}$. Therefore, this paper models the total number of complex scalars that the secondary CPUs have to exchange with primary CPUs through the backhaul as

$$\mathrm{BT}^{\text{pri}} = \sum_{j\in\mathcal{J}^{\text{sec}}} \sum_{k\in\mathcal{K}_j^{\text{sec}}} \tau_d + \mathbb{I}\left(2L_{jk}N\right), \tag{16}$$

where the binary indicator $\mathbb{I} \in \{0, 1\}$ specifies the network implementation type, with $\mathbb{I} = 1$ corresponding to the centralized implementation, and $\mathbb{I} = 0$ to the distributed one. The term $\mathcal{J}^{\text{sec}}$ denotes the subset of CPUs acting as a secondary CPU. It can be seen that the condition $\left|\mathcal{K}_j^{\text{sec}}\right| \leq K^{\text{sec}}$ prevents (16) from growing with the number of UEs K. That is, even if the number of UEs K goes to infinity, the CPU j will serve at most K^{sec} UEs as a secondary CPU. In other words, the number of sums performed in $\sum_{k\in\mathcal{K}_j^{\text{sec}}} \tau_d + \mathbb{I}\left(2L_{jk}N\right)$ will be upper-bounded by K^{sec}, since $\left|\mathcal{K}_j^{\text{sec}}\right| \leq K^{\text{sec}}$. This condition is not met in a traditional UC system since $\left|\mathcal{K}_j^{\text{sec}}\right|$ is not upper-bounded by K^{sec}. Instead, it would be a function of the number of UEs, such that $\left|\mathcal{K}_j^{\text{sec}}\right| \leq K$. For instance, in the worst-case scenario, a CPU j could serve all UEs as a secondary CPU, leading to the second summation of (16) becoming $\sum_{k=1}^{K} \tau_d + \mathbb{I}\left(2L_{jk}N\right)$.

Moreover, one can note that BT^{pri} can be further reduced by limiting the number of RUs serving the UE k to a maximum value called C_{max}, such that $L_k \leq \mathrm{C}_{max}$. Section IV discusses how to compute C_{max} in more detail. Finally, note that (16) is equivalent to $\mathrm{BT}^{\text{pri}} = \sum_{k=1}^{K} \mathrm{BT}_k^{\text{dist}}$ for $\mathbb{I} = 0$, and $\mathrm{BT}^{\text{pri}} = \sum_{k=1}^{K} \mathrm{BT}_k^{\text{pri}}$ for $\mathbb{I} = 1$.

F. Required Fronthaul and Backhaul Bit rates

The fronthaul bit rate required by UC D-mMIMO systems varies depending on the network implementation. In the distributed implementation, the bit rate scales with the number of UEs served by the RU (i.e., K_l). In the centralized one, it scales with the number of antennas N deployed at RU. The required fronthaul traffic (in bit/s) can be expressed as [25]

$$R_{\text{fh},l} = 2B\frac{\tau_d}{\tau_c}\left(N\left[b_l + b_l^{\text{p}}\frac{\tau_p}{\tau_d}\right]\mathbb{I} + \sum_{k\in\mathcal{D}_l} b_{kl}\left(1-\mathbb{I}\right)\right), \tag{17}$$

where B represents the system bandwidth, while b_l and b_{kl} are the number of quantization bits per sample used for DL data transmission in centralized and distributed implementations, respectively. In addition, b_l^{p} is the number of quantization bits per sample used for pilot signals. One can note that the fronthaul link requirements are constrained by the number of quantization bits for $\mathbb{I} = 1$. Conversely, fronthaul requirements may rise with the number of UEs served by RU l (K_l) when $\mathbb{I} = 0$. However, the maximum value of $R_{\text{fh},l}$ remains constant in a scalable system, i.e., $K_l \leq \tau_p$. The bit rate R_{bh} exchanged in all backhaul links is modeled in this paper as

$$R_{\text{bh}} = \frac{2B}{\tau_c}\left(\sum_{k=1}^{K} b_k^{\text{bh}}\left[\left(\mathrm{BT}_k^{\text{cent}}\right)\mathbb{I} + \mathrm{BT}_k^{\text{dist}}(1-\mathbb{I})\right]\right), \tag{18}$$

where b_k^{bh} represents the number of bits utilized to quantize the signals traveling in the backhaul links. It is assumed that $b_k^{\text{bh}} = b_l^{\text{p}} = b^{\max}$, with $b^{\max}$ being the maximum number of quantization bits per sample. The purpose is to transmit pilots and signals on backhaul links at maximum resolution to reduce quantization errors in channel estimation and data received on secondary CPUs. The number of quantization bits per sample utilized for DL data transmission (b_{kl}) is computed by considering that all RUs are operating at their maximum capacity, i.e., $K_l = \tau_p$. Thus, b_{kl} can be computed from (17) for $\mathbb{I} = 0$ as $b_{kl} = \left\lfloor R_{\text{fh},l}^{\max}\tau_c/2B\tau_d\tau_p \right\rfloor$ for all k in $\mathcal{D}_l$, where $\lfloor . \rfloor$ is the floor operation, and $R_{\text{fh},l}^{\max}$ denotes the maximum transmission capacity of each fronthaul link. Similarly, b_l can be calculated as $b_l = \lfloor R_{\text{fh},l}\tau_c/(2B\tau_d N) - b_l^{\text{p}}(\tau_p/\tau_d)\rfloor$. This paper assumes that $R_{\text{fh},l}^{\max} = 10\,\text{Gbps}$ and $b^{\max} = 12$. It is also considered that $b_l = b_{kl}$ to perform a fair comparison between centralized and distributed implementations[3]. Therefore, $b_l = b_{kl} = \min(b_l, b_{kl})$.

III. Computational Complexity and Energy Efficiency

A. Computational Complexity

The CC required for signaling processing tasks differs between network implementations for both CPUs and RUs. Therefore, this section also utilizes the binary indicator $\mathbb{I} \in \{0, 1\}$ to distinguish them. The CC required from each CPU j in giga operations per second (GOPS) can be expressed as

$$CC_{\text{CPU},j} = S_f\left(CC_{\text{CPU},j}^{\text{cecb}} + CC_{\text{CPU},j}^{\text{rcp}}\right)\mathbb{I} + CC_{\text{CPU},j}^{\text{basic}}, \tag{19}$$

[3]One can further decrease or enhance the values of b_{kl} and b_l. However, a deeper investigation regarding selecting the best values of b_{kl} and b_l is out of the scope of this paper.

where $CC^{\text{cecb}}_{\text{CPU},j} = CC^{\text{est}}_{\text{CPU},j} + CC^{\text{comb}}_{\text{CPU},j}$ denotes the number of complex multiplications that CPU j needs to perform channel estimation and generate the combining vectors. $CC^{\text{rcp}}_{\text{CPU},j} = (1+\tau_d)\, N \sum_{l\in\mathcal{J}_l} K_l$ stands for the CC associated with reciprocity calibration and precoding, with $\mathcal{J}_l$ being the subset of RUs connected to CPU j. Additionally, $S_f = 8N_{sc}/T_s\tau_c 10^9$ is a scaling factor that converts $CC^{\text{cecb}}_{\text{CPU},j}$ and $CC^{\text{rcp}}_{\text{CPU},j}$ into GOPS, where T_s is the orthogonal frequency-division multiplexing (OFDM) symbol duration, and N_{sc} represents the number of subcarriers. The last term of (19) is the CC in GOPS associated with higher-layer control/network functions, channel coding, mapping/demapping, and OFDM modulation/demodulation. These are computed following [28].

It is worth mentioning that $CC^{\text{est}}_{\text{CPU},j}$ and $CC^{\text{comb}}_{\text{CPU},j}$ are obtained from Table I, where $\mathcal{S}_k = \{i : \mathbf{D}_k\mathbf{D}_i \neq \mathbf{0}_{LN\times LN}\}$ represents the subset of UEs that are partially served by the same RUs as UE k. The term $\mathcal{K}^{\text{all}}_j$ denotes the subset of all UEs that CPU j is serving, thus $\mathcal{K}^{\text{all}}_j = \mathcal{K}^{\text{pri}}_j \cup \mathcal{K}^{\text{sec}}_j$. The CC required from RU l in GOPS can be computed as [29]

$$CC_{\text{RU},l} = S_f \left(CC^{\text{cecb}}_{\text{RU},l} + CC^{\text{rcp}}_{\text{RU},l}\right)(1-\mathbb{I}) + CC^{\text{other}}_{\text{RU},l}, \quad (20)$$

where $CC^{\text{cecb}}_{\text{RU},l} = CC^{\text{est}}_{\text{RU},l} + CC^{\text{comb}}_{\text{RU},l}$ denotes the number of complex multiplications required by RU l to perform channel estimation and generate the combining vectors. $CC^{\text{rcp}}_{\text{RU},l} = (1+\tau_d)\, NK_l$ is the CC associated with reciprocity calibration and precoding application. The specific values of $CC^{\text{est}}_{\text{RU},l}$ and $CC^{\text{comb}}_{\text{RU},l}$ are calculated in Table I. The last term of (20) is obtained as $CC^{\text{other}}_{\text{RU},l} = CC^{\text{DFT}} + CC^{\text{bbf}}$, where $CC^{\text{DFT}} = 8NN_{DFT}\log_2(N_{DFT})/T_s 10^9$ is the CC in GOPS due to discrete Fourier transform (DFT) operations, with $N_{DFT} \leq N_{sc}$ being the dimension of the DFT [30]. We have assumed that $N_{DFT} = N_{sc}$. Moreover, $CC^{\text{bbf}} = 40Nf_s/10^9$ represents the CC in GOPS related to baseband filtering, considering a filter with ten taps in a polyphase filtering scheme, where f_s is the sampling frequency [31].

B. Energy Efficiency

The energy efficiency (EE) in bit/Joule is calculated as the ratio between the sum throughput in bit/s and the total power consumed in Watts (W), being expressed as

$$\text{EE}_{\text{tot}} = \frac{B\sum_{k=1}^{K}\text{SE}_k}{\sum_{l=1}^{L}\{P_l + P_{\text{fh},l}\} + P^{\text{proc}}_{\text{CPUs}} + P_{\text{bh}}}, \quad (21)$$

where P_l denotes the total power consumption in RU l, while $P_{\text{fh},l}$ represents the power consumed by the fronthaul link connected to RU l. Additionally, $P^{\text{proc}}_{\text{CPUs}}$ is the power that all CPUs need for processing tasks, and P_{bh} accounts for power consumption in all backhaul links. P_l is calculated as $P_l = \mathbb{E}\{\|\mathbf{x}_l\|^2\}/\gamma_l + NP_{\text{tc},l} + P^{\text{proc}}_{\text{RU},l}$, where $0 < \gamma_l \leq 1$ represents the efficiency of the power amplifier, $P_{\text{tc},l}$ is the power required for each antenna of RU l to operate internal components like converters and filters, and $P^{\text{proc}}_{\text{RU},l}$ accounts for the power needed by RU l to perform processing tasks. The latter can be given by [32]

$$P^{\text{proc}}_{\text{RU},l} = P^{\text{proc}}_{\text{RU},0} + \Delta^{\text{proc}}_{\text{RU},l}\left(\frac{CC_{\text{RU},l}}{CC^{\max}_{\text{RU}}}\right), \quad (22)$$

where $P^{\text{proc}}_{\text{RU},0}$ is the power consumed by each digital signal processor (DSP) of RU l in idle mode; $\Delta^{\text{proc}}_{\text{RU},l}$ is the slope of power consumption due to processing in RU l, and $CC^{\max}_{\text{RU}}$ indicates the maximum GOPS capacity of the DSP in RU l.

The power consumed in each fronthaul link is calculated as $P_{\text{fh},l} = P_{0,l} + P_{\text{ft},l}, R_{\text{fh},l}$, where $P_{0,l}$ is the fixed power consumption of each fronthaul link, $P_{\text{ft},l}$ denotes the traffic-dependent power in Watt per bit/s, and $R_{\text{fh},l}$ is computed in (17). Similarly, for backhaul links, $P_{\text{bh}} = 0.5 \times J(J-1)P_{\text{bh},0} + P_{\text{bt}}R_{\text{bh}}$, where $P_{\text{bh},0}$ and P_{bt} represent the fixed and traffic-dependent power of each backhaul link, while R_{bh} is computed in (18). The term $0.5 \times J(J-1)$ refers to a fully connected topology, where each CPU has a direct connection to each other. Finally, $P^{\text{proc}}_{\text{CPUs}}$ can be expressed as

$$P^{\text{proc}}_{\text{CPUs}} = \frac{1}{\sigma_{\text{cool}}}\left(\Delta^{\text{proc}}_{\text{GPP}}\frac{CC_{\text{CPUs}}}{CC^{\max}_{\text{GPP}}} + \chi^{\text{proc}}_{\text{CPUs}}\right), \quad (23)$$

where $0 < \sigma_{\text{cool}} \leq 1$ denotes the cooling efficiency and CC_{CPUs} is calculated as $CC_{\text{CPUs}} = \sum_{j=1}^{J} CC_{\text{CPU},j}$. Moreover, $\Delta^{\text{proc}}_{\text{GPP}}$ stands for the slope of power consumption in a general purpose processor (GPP), and $CC^{\max}_{\text{GPP}}$ is the maximum processing capacity of each GPP in GOPS. The term $\chi^{\text{proc}}_{\text{CPUs}}$ is obtained as $\chi^{\text{proc}}_{\text{CPUs}} = P^{\text{proc}}_{\text{GPP},0}\sum_{j=1}^{J} W_j$, with W_j denoting the number of active GPPs in each CPU j, and $P^{\text{proc}}_{\text{GPP},0}$ representing the power consumed by each active GPP during idle mode. The term W_j can be given by $W_j = \lceil CC_{\text{CPU},j}/CC^{\max}_{\text{GPP}}\rceil$, with $\lceil . \rceil$ being the ceiling operation. It is worth mentioning that (23) extends the load dependent power consumption model proposed in [32] to a multi-CPU scenario.

IV. Scalable UC D-mMIMO Systems with Restricted Processing Capacity

In scalable D-mMIMO systems, the network complexity does not grow with the number of UEs since the number of UEs that each RU serve is limited, i.e., $K_l \leq \tau_p$, where $K_l = |\mathcal{D}_l|$. Therefore, the maximum number of UEs served by each RU remains finite even if the number of UEs K goes to infinity. However, the complexity of performing channel estimation and computing the precoding vectors can still grow with the number of RUs [9]. That is, as L increases, the number of RUs connected to the UE k (L_k) can also increase, resulting in more processing complexity from the network, where $L_k = |\mathcal{M}_k|$. To circumvent this issue, we rely on a strategy where each UE can be associated only with a finite number of RUs, denoted as C_{max}, with $L_k \leq \text{C}_{max}$ [19]. We refer to this strategy as maximum RU cluster size control. It is noteworthy that despite having a similar function, the C_{max} on this work is fundamentally different from the one presented in [19]. In this paper, C_{max} is a parameter that refers to the system processing capacity limitation that provides a new type of analysis for UC D-mMIMO systems.

A. RU Cluster Size Control with CPUs Cooperation

The maximum RU cluster size control procedure can be described as follows: when a new UE k enters the network, it measures the large-scale fading coefficients of the RUs in its vicinity, which is calculated according to $\beta_{kl} = \text{tr}\left(\mathbf{R}_{kl}\right)/N$

TABLE I: Number of complex multiplications required from CPUs and RUs to perform channel estimation and generate the combining vectors in each coherence block for different precoding schemes.

Scheme	Channel estimation		Combining vector computation	
P-RZF	$CC^{\text{est}}_{\text{CPU},j}$	$\sum_{k\in\mathcal{K}_j^{\text{all}}} \left(N\tau_p + N^2\right) L_{jk}$	$CC^{\text{comb}}_{\text{CPU},j}$	$\sum_{k\in\mathcal{K}_j^{\text{pri}}} \left[\frac{\lvert\mathcal{S}_k\rvert^2+\lvert\mathcal{S}_k\rvert}{2} NL_k + \lvert\mathcal{S}_k\rvert^2 + \lvert\mathcal{S}_k\rvert\, NL_k + \frac{\lvert\mathcal{S}_k\rvert^3-\lvert\mathcal{S}_k\rvert}{3}\right]$
P-MMSE	$CC^{\text{est}}_{\text{CPU},j}$	$\sum_{k\in\mathcal{K}_j^{\text{all}}} \left(N\tau_p + N^2\right) L_{jk}$	$CC^{\text{comb}}_{\text{CPU},j}$	$\sum_{k\in\mathcal{K}_j^{\text{pri}}} \left[\frac{(NL_k)^2+NL_k}{2} \lvert\mathcal{S}_k\rvert + (NL_k)^2 + \frac{(NL_k)^3-NL_k}{3}\right]$
LP-MMSE	$CC^{\text{est}}_{\text{RU},l}$	$\left(N\tau_p + N^2\right) K_l$	$CC^{\text{comb}}_{\text{RU},l}$	$\frac{1}{2}(N^2+N)K_l + N^2K_l + \frac{1}{3}(N^3-N)$
MR	$CC^{\text{est}}_{\text{RU},l}$	$\left(N\tau_p + N^2\right) K_l$	$CC^{\text{comb}}_{\text{RU},l}$	-

[9]. Then, it claims a master RU to ensure its connection with at least one RU. The master RU serves the UE even if it has a poor channel condition [8]. The UE k points the RU l with

$$\begin{aligned} l = \arg\max\nolimits_l \beta_{kl} \\ \text{s.t}\,\lvert\mathcal{A}_l\rvert < \tau_p, \end{aligned} \tag{24}$$

to be its master RU, where $\mathcal{A}_l \subset \mathcal{D}_l$ represents the subset of UEs the RU l serves as master RU. In order to solve (24), the UE k requests a connection to the available RUs. Posteriorly, the available RUs respond, and the UE k chooses the one with the strongest channel gain β_{kl} to be its master RU. The available RUs are the ones presenting $\lvert\mathcal{A}_l\rvert < \tau_p$, $\forall l \in \{1,\ldots,L\}$. Furthermore, $\lvert\mathcal{B}_l\rvert + \lvert\mathcal{A}_l\rvert \leq K_l$, where $\mathcal{B}_l \subset \mathcal{D}_l$ represents the subset of UEs the RU serves, but not as a master[4] (i.e., UEs that the RU may disconnect). It is worth mentioning that the CPU connected to the UE's master RU will be considered the UE's primary CPU [16].

After selecting the master RU, the UE k performs any UC RU selection scheme[5] in (6). In the following, the CPUs associated with the RU cluster of the UE k share the indexes of the RUs serving the UE ($\mathcal{M}_k$) with each other. Then, the CPUs serving the UE k compute the number of RUs serving the UE k, i.e., $L_k = \lvert\mathcal{M}_k\rvert$. If $L_k \leq \mathrm{C}_{max}$, no action is required. Otherwise, the CPUs will drop the connection of the UE k with the E_k RUs presenting the weakest channel gains, where E_k denotes the number of RUs that exceed C_{max}, which is calculated as $E_k = L_k - \mathrm{C}_{max}$. Let $\mathcal{J}_k$ denote the subset of CPUs associated with the RU cluster of the UE k. The maximum RU cluster size control is performed in J_k CPUs, where $J_k = \lvert\mathcal{J}_k\rvert$.

In order to drop the RUs in excess, the J_k CPUs serving the UE k sort the channel gains (β_{kl}) of the RUs serving the UE k in ascending order, such that $\tilde{\beta}_{kl'} \leq \cdots \leq \tilde{\beta}_{k(L_k)}$, where $\tilde{\beta}_{kl'}$ denotes the sorted version of β_{kl}, $\forall l \in \mathcal{M}_k$. The indexes of the RUs before the sort operation are stored in the l'-th element of the subset $\bar{\mathcal{M}}_k$. Finally, the CPUs drop the connection of the first E_k RUs presenting the smallest channel gains after the sort operation. This procedure is given by

$$\mathrm{c}_{kl} = \begin{cases} 0 & \text{if } l' \leq E_k \\ 1 & \text{otherwise,} \end{cases} \tag{25}$$

where l' is mapped to the unsorted value of l in subset $\bar{\mathcal{M}}_k$. Hence, the final RU cluster of UE k will only be composed of the C_{max} RUs with the largest channel gains. Algorithm 1 summarizes the maximum RU cluster size control algorithm performed by the CPUs serving the UE.

The time complexity of the proposed method is computed as follows: the complexity for choosing a master RU by solving (24) is $\mathcal{O}(L)$. The time complexity to perform RU cluster size control in each CPU j is $\mathcal{O}\big(\lvert\mathcal{K}_j^{\text{all}}\rvert \log \lvert\mathcal{K}_j^{\text{all}}\rvert\big)$, since each CPU has to perform a sort operation before computing (25). Thus, the overall time complexity can be expressed as $\mathcal{O}\big(L + \sum_{j=1}^{J} \lvert\mathcal{K}_j^{\text{all}}\rvert \log \lvert\mathcal{K}_j^{\text{all}}\rvert\big)$.

Remark 2: *It is considered that controlling RU cluster size is done before reducing the effects of inter-communication between CPUs. In other words, we first limit the number of CPUs serving the UE so that $L_k \leq \mathrm{C}_{max}$. Then, we reduce the effects of inter-CPU communication in* (14)*. This sequence also applies to the other methods in Sections IV and V. That is, they are performed before computing* (14)*.*

B. RU Cluster Size Control without Using CPUs

This subsection presents a method that does not utilize the CPUs for controlling the maximum RU cluster sizes. Instead, it shows that RU selection methods themselves can incorporate C_{max}. Therefore, it is considered that the proposed approach is performed only between UEs and RUs. This one relies on a matched-decision (MD) strategy to make the RU clusters meet the restriction C_{max}. The MD strategy is a general RU selection framework that makes the RU clusters be composed of the more convenient connections for UEs and RUs. It is divided into two steps, where the UE first requests a connection to a subset of RUs following a decision criterion, e.g., largest-large-scale fading [5]. In the following, the RUs accept or reject the UE request using criteria such as least pilot contamination [9], effective channel gain [11], among others.

The subset of RUs selected by UE k is denoted by $\mathbf{e}_k = [\mathrm{e}_{k1}, ..., \mathrm{e}_{kL}] \in \mathbb{N}^{1\times L}$, where $\mathrm{e}_{kl} = 1$ if the RU is in accordance with the criterion adopted by the UE. Otherwise, $\mathrm{e}_{kl} = 0$. It is assumed that the UE k solves (24) and requests a connection to the C_{max} RUs presenting the strongest

[4]Subset $\mathcal{B}_l$ does not affect the master RU assignment in (24). For instance, if $K_l = \tau_p$ and $\lvert\mathcal{B}_l\rvert \geq 1$, the RU l could drop the UE with the weakest channel gain of $\mathcal{B}_l$ to serve the UE k in subset $\mathcal{A}_l$.

[5]The efficiency of the proposed solution is proportional to the effectiveness of the RU selection method. Thus, if the RU selection method does not provide a connection for a given UE, the proposed technique will not be activated.

channel gains β_{kl} in its vicinity. For this, the UE k sorts the channel gains of the RUs in descending order, such that $\tilde{\beta}_{kl'} \geq \cdots \geq \tilde{\beta}_{kL}$, where $\tilde{\beta}_{kl'}$ denotes the sorted version of β_{kl}, for $l = \{1, ..., L\}$. The indexes of the RUs before the sort operation are stored in the l'-th element of the subset $\bar{\mathcal{M}}_k = \{1, ..., L\}$. Then, the UE requests a connection to the first C_{max} RUs presenting the strongest channel gains after the sort operation. This procedure can be expressed as

$$\mathrm{e}_{kl} = \begin{cases} 1 & \text{if } (l' \leq \mathrm{C}_{max}) \vee (\mathrm{UE}\, k \in \mathcal{A}_l) \\ 0 & \text{otherwise,} \end{cases} \tag{26}$$

where l' is mapped to the unsorted value of l in subset $\bar{\mathcal{M}}_k$. The decision taken in the RU l is denoted by $\mathbf{f}_k = [\mathrm{f}_{k1}, ..., \mathrm{f}_{kL}] \in \mathbb{N}^{1\times L}$, where $\mathrm{f}_{kl} = 1$ if the RU l accepts the UE k. Otherwise, $\mathrm{f}_{kl} = 0$. It is considered that the RUs also rely on the channel gain to use similar decision criteria among the UEs and RUs. Therefore, f_{kl} is expressed as

$$\mathrm{f}_{kl} = \begin{cases} 1 & \text{if } (\mathrm{UE}\, k \in \mathcal{A}_l) \vee (K_l < \tau_p) \vee \left(\beta_{kl} > \beta_{il}^{\min}\right) \\ 0 & \text{otherwise,} \end{cases} \tag{27}$$

where $i \neq k$ denotes the UE with the smallest channel gain that the RU l serves in $\mathcal{B}_l$. It is worth noting that in (27), the RU l drops the UE i to serve the UE k if $K_l = \tau_p$. Finally, the RU cluster of the UE k is given by

$$\mathbf{c}_k = \mathbf{e}_k \wedge \mathbf{f}_k. \tag{28}$$

Therefore, the connections will only be established if both UEs and RUs agree. Algorithm 2 summarizes the maximum RU cluster size control without using CPUs.

Algorithm 1: RU cluster size control: CPUs

Input: $l = 1, ..., L$, C_{max}
The UE connects to a master RU by solving (24) and associates with a subset of RUs ($\mathcal{M}_k$) in (6);
Identify the J_k CPUs serving the UE; // $J_k = |\mathcal{J}_k|$
// The J_k CPUs perform RU cluster size control:
if $L_k > \mathrm{C}_{max}$ **then**
  $E_k = L_k - \mathrm{C}_{max}$; // where $L_k = |\mathcal{M}_k|$
  Sort the channel gains of the RUs serving the UE in ascending order, such that $\tilde{\beta}_{kl'} \leq \cdots \leq \tilde{\beta}_{k(L_k)}$;
  for $l' = 1$ **to** E_k **do**
    Map l' to the unsorted value of l in subset $\bar{\mathcal{M}}_k$;
    $\mathrm{c}_{kl} = 0$; // Computed in (25)
  end
end
Output: $\mathbf{c}_k = [\mathrm{c}_{k1}, \ldots, \mathrm{c}_{kL}]$.

V. RU Cluster Adjustment

In this section, a heuristic method that adjusts the RU clusters according to the network implementation is proposed. Such method holds for any UC RU selection scheme, i.e., with and without processing capacity limitation. Besides, it is a heuristic strategy because only heuristic solutions are scalable [9]. In a nutshell, the UEs are associated with a subset of RUs following any RU selection process. Then, the

Algorithm 2: RU cluster size control: RUs and UEs

Input: C_{max}, $\bar{\mathcal{M}}_k = \{1, ..., L\}$
The UE k connects to a master RU by solving (24) and sorts the channel gains (β_{kl}) of the RUs in descending order, such that $\tilde{\beta}_{kl'} \geq \cdots \geq \tilde{\beta}_{kL}$.
$\mathrm{e}_{kl} = 0$; $\mathrm{f}_{kl} = 0$, $\forall l \in \bar{\mathcal{M}}_k$.
for $l' = 1$ **to** C_{max} **do**
  Map l' to the unsorted value of l in subset $\bar{\mathcal{M}}_k$.
  $\mathrm{e}_{kl} = 1$; // Request a connection to nearby RUs
  // RUs accept or reject the UE request:
  if $k \in \mathcal{A}_l$ **or** $\beta_{kl} > \beta_{il}^{\min}$ **then**
    $\mathrm{f}_{kl} = 1$; $\mathrm{f}_{il} = 1$; // where $i \in \mathcal{B}_l$
    if $K_l = \tau_p$ **then**
      $\mathrm{f}_{il} = 0$;
    end
  end
  $\mathrm{c}_{kl} = (\mathrm{e}_{kl} \wedge \mathrm{f}_{kl})$ // Matched-decision
end
Output: $\mathbf{c}_k = [\mathrm{c}_{k1}, \ldots, \mathrm{c}_{kL}]$.

proposed method aims to simultaneously reduce the number of UEs served by each RU l (K_l) and the number of RUs connected to each UE k (L_k) while keeping the SE under minor degradation. In this context, it is a novel way to reduce the CC and increase EE in scalable UC D-mMIMO systems. Throughout the analysis, it is also assumed that each UE connects to a master RU.

A. RU Cluster Adjustment in the Distributed Implementation

In the distributed implementation, the proposed method exploits the local long-term CSI at each RU and intends to reduce K_l without causing significant SE degradation. When all RUs are involved, the average value of L_k is also reduced. It is noteworthy that L_k is not directly reduced in distributed implementation, and neither could it be since it would require global long-term CSI at each RU.

The adjustment of the RU cluster relies on two proposed metrics: (i) the partial channel strength indicator ($\bar{\beta}_{kl}$) and (ii) the total channel strength indicator ($\bar{\beta}_l$). We use these metrics to prevent the less fortunate UEs from being easily dropped by the RU. Thus, they do not directly represent the long-term CSI of the UEs that the RU serves. Instead, they are the long-term CSI raised to a normalization exponent, defined as λ_l, which provides a better balance between the channel gains of the most and less fortunate UEs served by the RU, such that $0 < \lambda_l < 1$. Without this normalization, the RU could easily drop a UE presenting a weaker channel gain if the RU was also serving UEs with stronger channel gains. However, these differences can be reduced when the channel gains are raised to a power lower than one and greater than zero, such as λ_l.

The partial channel strength indicator is given by $\bar{\beta}_{kl} = (\beta_{kl})^{\lambda_l}$, where $\lambda_l = \min_{k\in\mathcal{D}_l}(\beta_{kl}) / \max_{k\in\mathcal{D}_l}(\beta_{kl})$. The second metric, called total channel strength indicator, is calculated as $\bar{\beta}_l = \sum_{k\in\mathcal{D}_l} \bar{\beta}_{kl}$. In the proposed method, the two metrics are used by each RU l to calculate $\bar{\beta}_{l,-k} = \bar{\beta}_l - \bar{\beta}_{kl}$, $\forall k \in \mathcal{D}_l$. The purpose of calculating $\bar{\beta}_{l,-k}$ is to evaluate how much the

total channel strength indicator $\bar{\beta}_l$ is reduced by dropping the UE k from the RU l. After computing $\bar{\beta}_{l,-k}$, the RU keeps the connection of UE k, only if

$$\mathrm{c}_{kl} = \begin{cases} 1 & \text{if } (\text{UE } k \in \mathcal{A}_l) \vee \left(\bar{\beta}_{l,-k} \leq \bar{\beta}_l^{mean}\right) \\ 0 & \text{otherwise,} \end{cases} \tag{29}$$

where $\bar{\beta}_l^{mean} = \sum_{k \in \mathcal{D}_l} \bar{\beta}_{l,-k}/K_l$ is a threshold value and $\mathcal{A}_l \subset \mathcal{D}_l$ is the subset of UEs that RU l serves as a master. One can note that the term $\bar{\beta}_{l,-k}$ has to be smaller than $\bar{\beta}_l^{mean}$, because $\bar{\beta}_{l,-k}$ will be small if the UE k has a large partial channel strength indicator $\bar{\beta}_{kl}$, since $\bar{\beta}_{l,-k} = \bar{\beta}_l - \bar{\beta}_{kl}$. Meanwhile, $\bar{\beta}_{l,-k}$ will be large if the UE k adds only a marginal gain to the total channel strength indicator $\bar{\beta}_l$. That is, if $\bar{\beta}_{kl}$ represents a considerable percentage of $\bar{\beta}_l = \sum_{k \in \mathcal{D}_l} \bar{\beta}_{kl}$, the term $\bar{\beta}_l$ will be significantly reduced if the UE k is disconnected from RU l.

B. RU Cluster Adjustment in the Centralized Implementation

In the centralized implementation, the long-term CSI of RUs and UEs is available at the CPUs [7], [9]. Hence, the proposed method exploits the global long-term CSI to reduce L_k. At first, reducing L_k may appear counter-intuitive since the centralized implementation has a better interference suppression capability. However, since CC grows with the number of RUs serving the UE (recall that $L_k = |\mathcal{M}_k|$), the RU cluster expansion will not always be beneficial, and reducing L_k may be necessary even in this implementation. In the centralized implementation, the RU cluster adjustment is also performed by the J_k CPUs associated with the RU cluster of the UE k, which are denoted as $\mathcal{J}_k$, where $J_k = |\mathcal{J}_k|$. Moreover, the J_k CPUs need to share the indexes of the RUs serving the UE ($\mathcal{M}_k$) with each other, as in Section IV.

The partial channel strength indicator is now calculated in the CPUs as $\bar{\beta}_{kl} = (\beta_{kl})^{\lambda_k}$, where λ_k introduces a balance between the serving RUs presenting the smallest and highest channel gain to the UE k. The CPUs compute λ_k as $\lambda_k = \min_{l \in \mathcal{M}_k}(\beta_{kl}) / \max_{l \in \mathcal{M}_k}(\beta_{kl})$. The total channel strength indicator is computed as $\bar{\beta}_k = \sum_{l \in \mathcal{M}_k} \bar{\beta}_{kl}$. Then, the CPUs calculates the contribution that each RU brings to $\bar{\beta}_k$ as $\bar{\beta}_{k,-l} = \bar{\beta}_k - \bar{\beta}_{kl}$, $\forall l \in \mathcal{M}_k$. Therefore, the CPU connected to the RU l keeps the connection of RU l with the UE k only if

$$\mathrm{c}_{kl} = \begin{cases} 1 & \text{if } (\text{UE } k \in \mathcal{A}_l) \vee \left(\bar{\beta}_{k,-l} \leq \bar{\beta}_k^{mean}\right) \\ 0 & \text{otherwise,} \end{cases} \tag{30}$$

where $\bar{\beta}_k^{mean} = \sigma_{si}/2 + \sum_{l \in \mathcal{M}_k} \bar{\beta}_{k,-l}/L_k$ and σ_{si} denotes the standard deviation of $\bar{\beta}_{k,-l}$, $\forall l \in \mathcal{M}_k$. The term σ_{si} is utilized to make the CPUs drop fewer RUs from the RU cluster of UE k to exploit the centralized implementation's capacity in improving SE. It is worth noting that only the CPUs associated with the RU cluster of the UE run the proposed method.

C. Pros and Cons of the two RU Clusters Adjustments

The utilization of the proposed method in a distributed implementation enables a fronthaul signaling reduction since the number of data flows on the fronthaul is proportional to K_l in (17). Besides, it allows the RU to carry out fewer operations while attaining the same SE performance, increasing the system's EE. Utilizing the proposed method in a centralized implementation also allows significant savings in CC resources. Nonetheless, it does not directly reduce the number of data flows in the fronthaul links, since the required bit rate is not proportional to K_l in (17). It is worth noting that this paper has considered that the RU cluster adjustment is only activated when λ_l and λ_k are lesser than a threshold Γ to avoid excessive adjustments, where Γ is a project parameter. We have set $\Gamma = 10^{-2}$ and $\Gamma = 10^{-3}$ for the distributed and centralized implementations, respectively.

VI. Numerical Results

We consider a D-mMIMO network consisting of K single antenna UEs and L RUs, each equipped with N antennas. The values of L, N, and K vary and are specified throughout the results. The K UEs are uniformly distributed over a square area of 1×1 km, and the distribution of the RUs follows a hard core point process (HCPP)[6]. After the RUs positioning, the coverage area is divided into J rectangle regions of the same size, each consisting of a CPU coordinating approximately L/J RUs, where $J = 4$. Each CPU can serve up to $K^{\text{sec}} = 10$ UEs as a secondary CPU [16]. The simulations focus on DL channels and it is assumed that $\tau_c = 200$, $\tau_p = 10$, and $\tau_d = 190$. The total transmission powers of the UEs and RUs are $100\,\text{mW}$ and $200\,\text{mW}$, respectively. We perform Monte-Carlo simulations to account for different RU/UE locations and channel realizations. The wrap-around technique is also utilized to provide a better balance regarding the amount of interference affecting each RU.

We utilize an RU clustering scheme that jointly performs the pilot assignment and RU selection [8]. In this one, the UEs can connect to master and non-masters RUs. The non-masters serve only the UEs with the greatest channel gain in each pilot. The first τ_p UEs are assigned to mutually orthogonal pilots, and the remaining ones to the pilot causing the lowest pilot contamination. Hereafter, we name it as scalable cell-free (SCF) scheme. Furthermore, we utilize two other RU selection strategies to assess the proposed approach's performance in UC D-mMIMO systems presenting distinct RU-UE association strategies. Both are non-scalable solutions but are utilized to demonstrate that the proposed approach can be used in any RU selection scheme when RU cluster size control is performed at the CPUs. The key features of these RU selection methods are described below:

- Largest-large-scale fading (LSFB) [5]: the UE measures the large-scale fading gains of the RUs in its surroundings and sums these channel gains. Posteriorly, it connects to the subset of RUs that contribute the most to the sum of its total channel gain in percentage $\delta\%$, with $\delta\% = 99.9$. The LSFB is a non-scalable scheme because it does not limit the number of UEs that each RU can serve.

[6]This method states that the distance between any two RUs cannot be smaller than $d_{\min} = \sqrt{A/L}$, where A is the coverage area in square meters. The first step is to randomly drop the RUs based on a homogeneous Poisson point process with mean rate $1/d_{\min}$, then randomly update the location of RUs that do not meet the spacing requirement until it is fulfilled.

- User-centric clustering (UCC) [6]: the RU serves the $U_{\max}$ UEs with the largest estimated channel in each coherence block, where $U_{\max}$ is the maximum number of UEs the RU can serve. We have adjusted this strategy to consider only the large-scale fading to avoid performing RU selection in each coherence block. Thus, the RU serves the $U_{\max}$ UEs presenting the largest large-scale fading in their vicinity, with $U_{\max} = \tau_p$.

The 3GPP Urban Micro (UMi) path loss model is adopted for modeling the propagation channel, with LOS/NLOS conditions defined in the Technical Report (TR) 38.901 [33]. It is considered that the shadowing terms of an RU to different UEs are correlated, and the computation of correlation matrices $\mathbf{R}_{kl}$ follows the local scattering spatial correlation model [8]. Table II exhibits the parameters used in the UMi and $\mathbf{R}_{kl}$ models [8], [34].

TABLE II: Parameters assumed for the UMi path loss and local scattering spatial correlation model.

Parameter	Value
Shadow fading standard deviation, σ_{SF}	4 dB
RU/UE antenna height, $h_{\text{RU}}, h_{\text{UE}}$	11.65 m, 1.65 m
RX noise figure (NF)	8 dB
Carrier frequency, bandwidth (B)	3.5GHz, 100MHz
Angular standard deviations (ASDs)	$\sigma_\varphi = \sigma_\theta = 15°$
Antenna spacing	1/2 wavelength distance

The power coefficients at RU l in the distributed implementation are set as $\rho_{kl} = \rho_d\sqrt{\beta_{kl}}/\sum_{k' \in \mathcal{D}_l}\sqrt{\beta_{k'l}}$, where ρ_d is the maximum transmit power per RU. For the centralized one, scalable fractional power control is used with the following parameters: $v = -0.5$ and $\kappa = 0.5$ [8]. The EE parameters related to the power consumption of the hardware of the RUs, fronthaul, and backhaul links are summarized in Table III, which follows [5], [29]. However, $P^{\text{proc}}_{\text{RU},0}$, $\Delta^{\text{proc}}_{\text{RU},l}$, and $CC^{\max}_{\text{RU}}$ are in accordance with a Texas Instruments TMS320C6678 DSP. Moreover, conventional 5G new radio (NR) parameters are assumed to compute the CC in GOPS, where $N_{DFT} = N_{sc} = 3300$, $f_s = 122.88$MHz, and $T_s = 35.38\mu$s. These values correspond to 30kHz of subcarrier spacing.

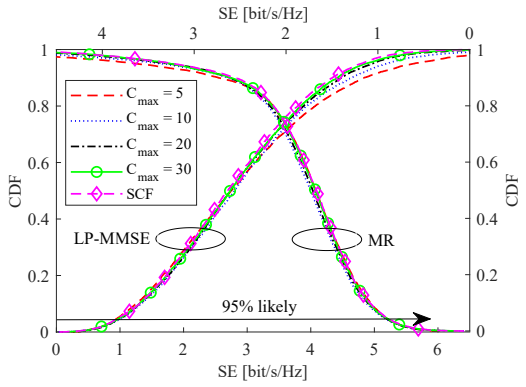
TABLE III: Parameters assumed for calculating the power consumption in CPUs, backhaul/fronthaul links, and EE.

Parameter	Value	Parameter	Value
$P^{\text{proc}}_{\text{RU},0}$, $P^{\text{proc}}_{\text{GPP},0}$	7.3 W, 20.8 W	$P_{0,l}$, $P_{\text{bh},0}$	0.825 W
$\Delta^{\text{proc}}_{\text{RU},l}$, $\Delta^{\text{proc}}_{\text{GPP}}$	73 mW, 74 W	$P_{\text{ft},l}$, $P_{\text{bt},l}$	0.25 W/(Gbit/s)
$CC^{\max}_{\text{RU}}$, $CC^{\max}_{\text{GPP}}$	180 GOPS	σ_{cool}, γ_l	0.9, 0.4
$R^{\max}_{\text{fh},l}$	10 Gbps	$b^{\max}$	12 bits

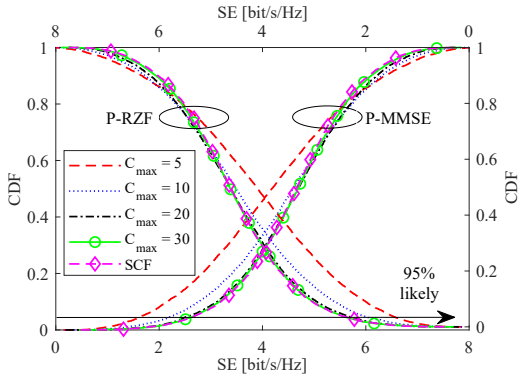
A. Impacts of Limiting the Processing Capacity

We start by evaluating a network composed of $K = 25$ UEs and $L = 100$ RUs equipped with $N = 1$ antenna. Fig. 2 presents the cumulative distribution functions (CDFs) of the SE of UC systems with and without processing capacity limitation. It considers different processing capacity limitations, i.e., several values of C_{max}, and the system is compared with a traditional UC scheme (i.e., L_k and K^{sec} are not restricted), which we have denoted as SCF. These results are presented for the case where CPUs impose that $L_k \leq \text{C}_{max}$.

In Fig. 2a, the SE is not significantly reduced by the variations of C_{max}. The SE even increases slightly for $5 \leq \text{C}_{max} \leq 10$. This is because decreasing L_k also reduces K_l, helping precoding techniques such as LP-MMSE (of local processing) to mitigate interference. In Fig. 2b, the SE can suffer significant losses when C_{max} is small. For instance, it decreases by 32% when C_{max} goes from 30 to 5. Hence, reducing the RU cluster sizes (L_k) may lead the centralized implementation to not exploit its full potential in mitigating interference and improving SE. Therefore, this implementation needs to utilize more processing capacity, such as $\text{C}_{max} \geq 20$.



(a) Distributed implementation.

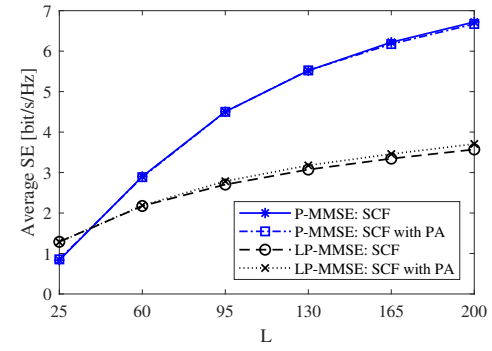


(b) Centralized implementation.

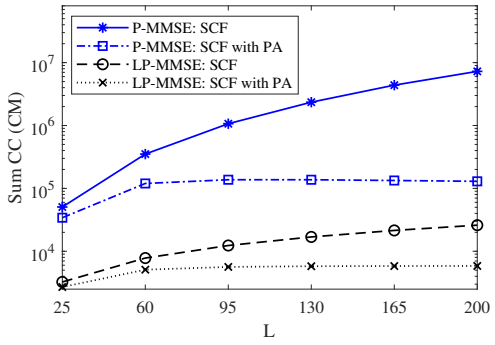
Fig. 2: CDF of SE by varying C_{max} from 5 to 30. Parameters setting: $J = 4$, $L = 100$, $K = 25$, $N = 1$, and $K^{\text{sec}} = 10$.

Fig. 3 presents the SE and sum of CC to perform channel estimation and generate the combining vectors when the number of RUs L varies. The CC is given in terms of number of complex multiplications (CM), i.e., only considering the terms $CC^{\text{cecb}}_{\text{CPU},j}$ and $CC^{\text{cecb}}_{\text{RU},j}$ in (19) and (20). In Fig. 3a, the average SE grows with L for UC systems with and without the proposed approach for processing capacity limitation. Despite this, limited systems have a significant advantage, as their CC does not always increase with L, starting to be constant from $L = 60$. This behavior occurs because K_l and L_k does not increase with L, as Table IV demonstrates. Additionally, it is possible to observe that the CC decreases by about 98.22% when the proposed approach for processing capacity limitation

is employed together with the P-MMSE scheme for $L = 200$. It is noteworthy that L_k does not equal C_{max} in Table IV. This happens because after applying (28), the proposed approach (PA) also reduces the effects of inter-CPU communication in (14). Thus, L_k is lowered both in (28) and (14), making $L_k < \mathrm{C}_{max}$.



(a) SE



(b) CC

Fig. 3: Average DL SE (a) and CC in terms of number of complex multiplications (CM) (b) achieved by varying the number of RUs L. Parameters setting: $J = 4$, $K = 25$, $N = 1$, $\mathrm{C}_{max} = 20$, and $K^{\mathrm{sec}} = 10$.

Fig. 4 shows the total CC in GOPS when the number of RUs L varies, i.e., considering all the terms of (19) and (20). One can note that the overall CC still increases with the number of RUs. This is because some network functions, such as higher-layer control, do not rely on C_{max}. Therefore, their CC (denoted as $CC^{\mathrm{basic}}_{\mathrm{CPU},j}$) still grows with the number of RUs L. However, the proposed approach still effectively reduces the CC. For instance, it decreases the CC by about 85% and 77.5% in centralized and distributed implementations, respectively.

TABLE IV: Average number of RUs per UE (L_k) and UEs per RU (K_l) without and with RU cluster control. Parameters setting: $J = 4$, $K = 25$, $N = 1$, $\mathrm{C}_{max} = 20$, and $K^{\mathrm{sec}} = 10$.

Method	$L = 95$		$L = 200$	
	K_l	L_k	K_l	L_k
SCF	10	38	10	80
With PA	4.56	17.35	2.25	18

Fig. 5 presents the backhaul traffic for UC D-mMIMO systems, with and without the proposed approach, when the number of UEs K grows. It can be noted that the proposed approach (i.e., using C_{max} and K^{sec}) decreases the backhaul traffic significantly. For instance, it reduces the backhaul traffic by about 77% and 80% in centralized and distributed implementations, respectively, for $K = 100$. These results indicate that the proposed strategy allows UC D-mMIMO systems to reduce their CC and signaling demands while keeping the SE under minor degradation.

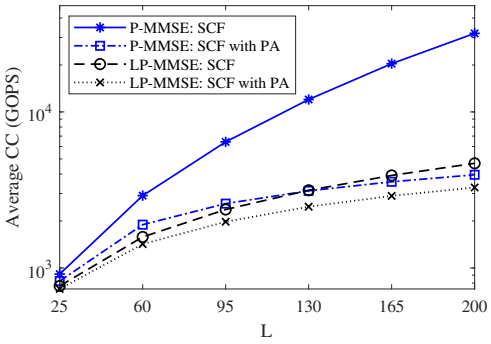


Fig. 4: Average CC in GOPS achieved by varying the number of RUs L. Parameters setting: $J = 4$, $K = 25$, $N = 1$, $\mathrm{C}_{max} = 20$, and $K^{\mathrm{sec}} = 10$.

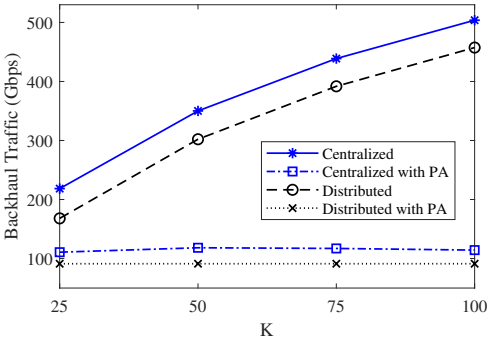


Fig. 5: Backhaul traffic in each network implementation by varying the number of UEs K. Parameters setting: $J = 4$, $L = 100$, $N = 1$, $\mathrm{C}_{max} = 20$, and $K^{\mathrm{sec}} = 10$.

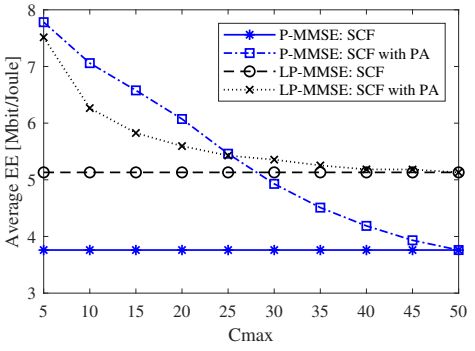


Fig. 6: Average EE achieved by varying C_{max}. Parameters setting: $J = 4$, $L = 100$, $K = 25$, $N = 1$, and $K^{\mathrm{sec}} = 10$.

Fig. 6 illustrates the EE achieved by a UC system with and

without the proposed approach for processing capacity limitations with different values of C_{max}. Note that the processing capacity limitation can provide a considerable improvement in the EE, especially for small values of C_{max}. For instance, the EE grows by about 47% in the LP-MMSE and 106% in the P-MMSE, when C_{max} decreases from 50 to 5. This happens because reducing C_{max} also decreases the number of UEs the RUs serve (i.e., K_l), as indicated in Table IV. For instance, the proposed approach reduces the average value of K_l from 10 to 2.25 for $L = 200$. Consequently, the power consumption in each fronthaul link $P_{\text{fh},l}$ reduces since $P_{\text{fh},l}$ is proportional to the number of UEs served by the RU. Thus, even though the system experiences SE losses when $C_{max} = 5$, the reduction in power consumption in the fronthaul links compensates for these losses, thereby increasing the EE. In other words, the denominator of (21), which contains $P_{\text{fh},l}$, decreases more than the numerator, which contains the SE.

Fig. 7 compares the RU cluster size control performed by the CPUs with the MD scheme, which operates between UEs and RUs. One can note that both methods maintain the SE with minor degradation compared to the SCF scheme. It is noteworthy that CPUs provide adaptable control for any RU selection method, while the MD strategy has limited flexibility as it is performed only between UEs and RUs. However, the MD scheme demonstrates that the benefits of limiting CC can be achieved whether the procedure of computing and applying $C_{\max}$ is centralized in CPUs or occurs locally among the UEs and RUs.

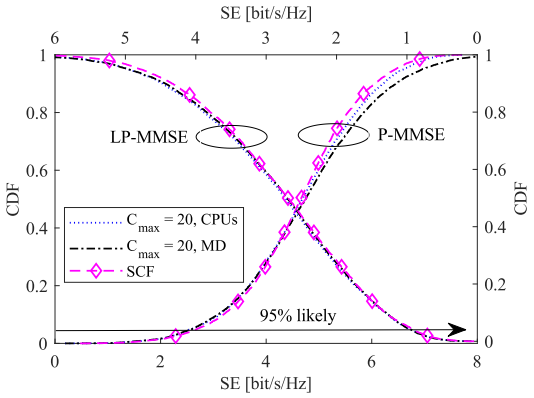
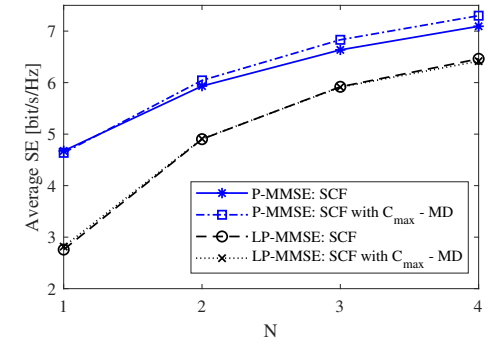
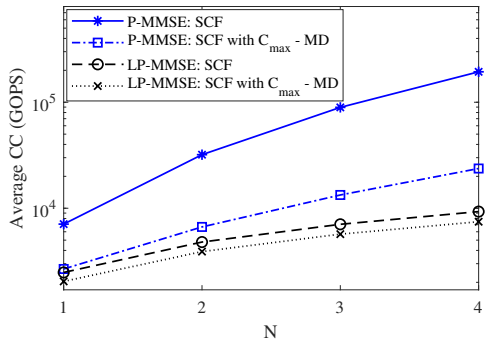


Fig. 7: CDF of SE for the proposed strategies to control the RU cluster size. Parameters setting: $J = 4$, $L = 100$, $K = 25$, $N = 1$, and $K^{\text{sec}} = 10$.

Fig. 8 presents the average SE and the total CC in GOPS as the number of antenna elements per RUs N varies. The MD scheme is used to control the maximum RU cluster size. As expected, the proposed approach reduces the overall CC in GOPS, as it mitigates the impact of N, as shown in Table I. Interestingly, the proposed approach also leads to some improvements in SE as N increases in the P-MMSE scheme. This happens because when the RU cluster size for certain UEs is large, these UEs may connect to RUs that contribute marginally to the desired signal while intensifying interference. Consequently, removing such RUs can be beneficial, particularly in sophisticated precoding techniques like P-MMSE, as these techniques can better manage interference and further exploit the advantages of using more antenna elements per RU when the UE is connected to fewer RUs.



(a) SE



(b) CC

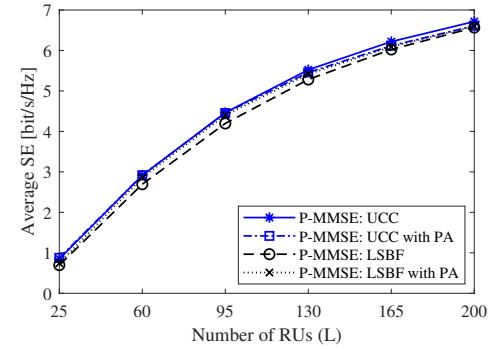
Fig. 8: Averages DL SE (a) and CC in GOPS (b) achieved by varying the number antennas per RU N, when the MD scheme is utilized to control the RU cluster size. Parameters setting: $J = 4$, $K = 25$, $L = 100$, $C_{max} = 20$, and $K^{\text{sec}} = 10$.

However, this same effect is not observed in the LP-MMSE scheme since its interference mitigation is performed locally at the RUs, which do not have access to global statistical and channel estimation information. Additionally, although not shown in the figures due to space constraints, our results demonstrate that the same behaviors are observed when the RU cluster size control is performed at the CPUs.

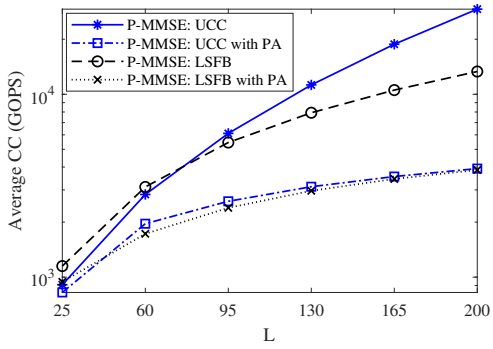
B. Comparison with other Baseline Solutions

Recall that the proposed approach can be applied to any RU selection strategy when RU cluster size control is performed at the CPUs, as described in Subsection IV-A. Therefore, we also evaluate the proposed approach for the UCC and LSFB RU selection schemes, which are non-scalable strategies but can be utilized to demonstrate the proposed approach's performance. To avoid redundancies, we present the results only for the P-MMSE precoding scheme.

In this regard, Fig. 9 shows the average SE and CC in GOPS as the number of RUs L, varies. It can be noted that the previous explanations hold even if the RU selection scheme is modified, i.e., the proposed approach can reduce the CC while yielding minor degradation in the SE. Besides, it is possible to observe that distinct RU selection schemes can lead to different



(a) SE



(b) CC

Fig. 9: Averages DL SE (a) and CC in GOPS (b) achieved by varying the number of RUs L, when the UCC and LSFB RU selection schemes are employed. Parameters setting: $J = 4$, $K = 25$, $N = 1$, $\mathrm{C}_{max} = 20$, and $K^{\text{sec}} = 10$.

CC and SE since they operate differently and thus affect the overall CC distinctly. For instance, the proposed approach decreases the average CC by about 86% and 54.5% for the UCC and LSFB schemes, respectively. It is worth mentioning that the higher CC of the UCC scheme compared to the LSFB is associated with the fact that the UCC scheme always makes the RUs operate at their maximum capacity, i.e., $K_l = \tau_p$. In contrast, K_l is usually lesser than τ_p in the LSFB strategy.

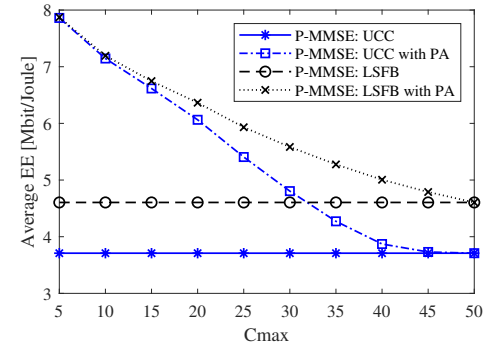


Fig. 10: Average EE achieved by varying C_{max}, when the UCC and LSFB RU selection schemes are employed. Parameters setting: $J = 4$, $L = 100$, $K = 25$, $N = 1$, and $K^{\text{sec}} = 10$.

The proposed approach can also improve the EE for both the UCC and LSFB schemes, as depicted in Fig. 10. The EE increases by about 56% and 94% in the LSFB and UCC schemes, respectively, when C_{max} decreases from 50 to 5. The higher EEs of the LSFB scheme is also related to the fact that it operates with RUs serving fewer UEs, which reduces not only the CC but also the power consumption in the fronthaul links, thereby improving the EE.

TABLE V: Average number of RUs per UE (L_k) and UEs per RU (K_l) without and with RU cluster adjustment. Parameters setting: $L = 100$, $N = 1$, and $K^{\text{sec}} = 10$.

Method	$K = 25$		$K = 50$	
	K_l	L_k	K_l	L_k
SCF	10	40	10	20
Distributed adjustment	4.32	17.3	4.38	8.75
Centralized adjustment	6.23	24.92	6.17	12.35

C. Impacts of RU Cluster Adjustment

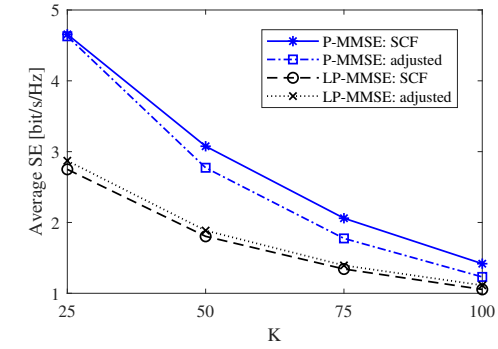
From now on, we will investigate the impacts of adjusting the RU clusters in UC D-mMIMO systems. We will focus on UC systems without processing capacity limitation to assess the full benefits of the RU cluster adjustment in reducing CC. Furthermore, we will consider only the P-MMSE and LP-MMSE schemes as they provide the best interference mitigation in centralized and distributed implementations.

Fig. 11 presents the average SE and total CC versus the number of UEs K in a network composed of $L = 100$ RUs equipped with $N = 1$ antenna. Note that the SE is significantly reduced in the P-MMSE with the proposed method (denoted as adjusted). Nonetheless, this reduction is related to the value of $K^{\text{sec}} = 10$, which is small for centralized implementation. For instance, although not shown in the figures due to space constraints, the average SE for $K^{\text{sec}} = 20$ has been analyzed, and the RU cluster adjustment revealed to affect the SE negligibly. One can also note that the proposed method causes a slight increase in the SE of LP-MMSE. Moreover, the RU cluster adjustment also reduces the CC of both network implementations, decreasing the CC by up to 58% in the P-MMSE scheme for $K = 25$. Finally, the proposed method decreases the values of K_l and L_k as illustrated in Table V, indicating that it can also increase the EE.

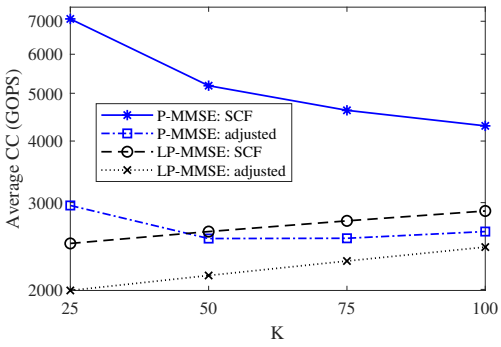
Fig. 12 presents the average SE and CC versus the number of RUs L and N for a fixed total number of antennas $M = LN = 100$, when $K = 25$. One can note that the same discussions regarding decreasing CC apply to this case. The difference is the SE behavior. When $L = 50$ and $N = 2$, the LP-MMSE scheme achieved the best balance regarding the amount of interference and desired signal, leading the average SE to its maximum value. Meanwhile, the P-MMSE presents better SE when the RU clusters are adjusted for $10 < L < 100$. This is because the fewer RUs in the coverage area, the further away the RUs will be from the UE. Therefore, disconnecting some of these RUs will not impact the UE's performance [12].

VII. Conclusions

This paper investigated the performance of scalable UC D-mMIMO systems with multiple CPUs. In this regard, we



(a) SE



(b) CC

Fig. 11: Average DL SE (a) and CC in GOPS (b) achieved by varying the number of UEs K, when the proposed RU cluster adjustment is employed. Parameters setting: $J = 4$, $L = 100$, $N = 1$, and $K^{\text{sec}} = 10$.

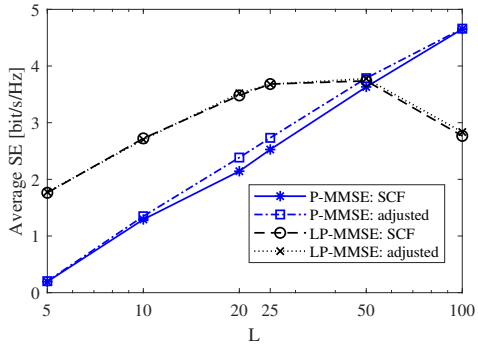


Fig. 12: Average DL SE by varying L and N, while keeping $M = LN = 100$, when the proposed RU cluster adjustment is employed. Parameters setting: $J = 4$, $K = 25$ and $K^{\text{sec}} = 10$.

proposed a framework that restricts the number of RUs serving each UE and considers that each CPU can only serve a limited number of UEs managed by other CPUs. The backhaul traffic was modeled, and the CC of performing channel estimation and generating the combining vectors was presented for multiple CPUs in four precoding schemes. The EE modeling was also improved by considering the power consumption in CPUs and backhaul links. Two strategies for controlling the size of the RU cluster were presented, where one was carried out by the CPUs and the other between the UEs and the RUs. Moreover, we proposed two methods that adjust the RU clusters to the network implementations, i.e., centralized and distributed. The results demonstrated that using the proposed strategies to restrict the processing capacity can improve the EE up to 106% in centralized implementation. However, it can degrade the SE of centralized implementation when the maximum number of RUs serving the UE is small. On the other hand, RU clusters comprising just a few RUs almost do not harm the SE of distributed implementation. The benefits of the proposed schemes were observed in both RU cluster size control methods presented in this paper.

Simulation results also reveal that the proposed framework allows UC D-mMIMO systems to decrease CC and signaling requirements while maintaining minor degradation in SE. For instance, it can reduce the CC to perform channel estimation and generate the combining vectors up to 98% while preventing it from growing with the number of RUs. The backhaul traffic due to inter-CPU communication is also controlled, i.e., it does not increase with the number of UEs. Nonetheless, the CC of certain network functions, such as higher-layer control, are not affected by the proposed methods. Thus, their CC continue to scale with the number of RUs. Finally, the proposed RU cluster adjustment can slightly improve the SE of distributed implementation while reducing the CC in both network implementations. These results open the way for future works to design practical UC systems with processing capacity and signaling constraints. Future works can expand our analyses to consider non-reciprocity and mobility.

REFERENCES

[1] M. M. M. Freitas, D. D. Souza, D. B. da Costa, A. M. Cavalcante, L. Valcarenghi, and J. C. W. A. Costa, "Scalable user-centric distributed massive MIMO systems with limited processing capacity," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2023, pp. 4298–4304.

[2] J. Zhang *et al.*, "Cell-free massive MIMO: A new next-generation paradigm," *IEEE Access*, vol. 7, pp. 99 878–99 888, Jul. 2019.

[3] I. F. Akyildiz *et al.*, "6G and beyond: The future of wireless communications systems," *IEEE Access*, vol. 8, pp. 133 995–134 030, Jul. 2020.

[4] H. Q. Ngo *et al.*, "Cell-free massive MIMO versus small cells," *IEEE Trans. Wireless Commun.*, vol. 16, no. 3, pp. 1834–1850, Mar. 2017.

[5] H. Q. Ngo, L. Tran, T. Q. Duong, M. Matthaiou, and E. G. Larsson, "On the total energy efficiency of cell-free massive MIMO," *IEEE Trans. Green Commun. Netw*, vol. 2, no. 1, pp. 25–39, Mar. 2018.

[6] S. Buzzi and C. D'Andrea, "Cell-free massive MIMO: User-centric approach," *IEEE Wireless Commun. Lett.*, vol. 6, no. 6, pp. 706–709, Dec. 2017.

[7] E. Björnson and L. Sanguinetti, "Making cell-free massive MIMO competitive with MMSE processing and centralized implementation," *IEEE Trans. Wireless Commun.*, vol. 19, no. 1, pp. 77–90, Jan. 2019.

[8] Ö. Demir *et al.*, *Foundations of User-Centric Cell-Free Massive MIMO*. Foundations and Trends® in Signal Processing, 2021, vol. 14, no. 3-4.

[9] E. Björnson and L. Sanguinetti, "Scalable cell-free massive MIMO systems," *IEEE Trans. Commun.*, vol. 68, no. 7, pp. 4247–4261, Jul. 2020.

[10] G. Interdonato *et al.*, "Scalability aspects of cell-free massive MIMO," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jul. 2019, pp. 1–6.

[11] H. T. Dao and S. Kim, "Effective channel gain-based access point selection in cell-free massive MIMO systems," *IEEE Access*, vol. 8, pp. 108 127–108 132, Jun. 2020.

[12] M. Freitas, D. Souza, G. Borges, A. M. Cavalcante, D. B. da Costa, M. Marquezini, I. Almeida, R. Rodrigues, and J. C. W. A. Costa, "Matched-decision AP selection for user-centric cell-free massive MIMO networks," *IEEE Trans. Veh. Technol*, vol. 72, no. 5, pp. 6375–6391, 2023.

[13] T. H. Nguyen, T. K. Nguyen, H. D. Han, and V. D. Nguyen, "Optimal power control and load balancing for uplink cell-free multi-user massive MIMO," *IEEE Access*, vol. 6, pp. 14 462–14 473, 2018.

[14] V. Ranasinghe, N. Rajatheva, and M. Latva-aho, "Graph neural network based access point selection for cell-free massive MIMO systems," in *Proc. IEEE Global Commun. Conf.*, Feb. 2021, pp. 01–06.

[15] V. Ranjbar *et al.*, "Cell-free mMIMO support in the O-RAN architecture: A PHY layer perspective for 5G and beyond networks," *IEEE Commun. Stand. Mag.*, vol. 6, no. 1, pp. 28–34, 2022.

[16] M. M. M. Freitas, D. D. Souza, D. B. d. Costa, A. M. Cavalcante, L. Valcarenghi, G. S. Borges, R. Rodrigues, and J. C. W. A. Costa, "Reducing inter-CPU coordination in user-centric distributed massive MIMO networks," *IEEE Wireless Commun. Lett.*, vol. 12, no. 6, pp. 957–961, Jun. 2023.

[17] H. A. Ammar *et al.*, "Distributed resource allocation optimization for user-centric cell-free MIMO networks," *IEEE Trans. Wireless Commun.*, vol. 21, no. 5, pp. 3099–3115, May 2022.

[18] F. Li, Q. Sun, X. Ji, and X. Chen, "Scalable cell-free massive MIMO with multiple CPUs," *Mathematics*, vol. 10, no. 11, Jun. 2022.

[19] D. D. Souza, M. M. M. Freitas, D. B. da Costa, G. S. Borges, A. M. Cavalcante, L. Valcarenghi, and J. C. Weyl Albuquerque Costa, "Effective channel DL pilot-based estimation in user-centric cell-free massive MIMO networks," in *Proc. IEEE Global Commun. Conf.*, Dec. 2022, pp. 705–710.

[20] O. Özdogan, E. Björnson, and E. G. Larsson, "Massive MIMO with spatially correlated Rician fading channels," *IEEE Trans. Commun.*, vol. 67, no. 5, pp. 3234–3250, May 2019.

[21] O. Özdogan, E. Björnson, and J. Zhang, "Performance of cell-free massive MIMO with Rician fading and phase shifts," *IEEE Trans. Wireless Commun.*, vol. 18, no. 11, pp. 5299–5315, Aug. 2019.

[22] A. A. Polegre *et al.*, "Channel hardening in cell-free and user-centric massive MIMO networks with spatially correlated Ricean fading," *IEEE Access*, vol. 8, pp. 139 827–139 845, Jul. 2020.

[23] E. Björnson, J. Hoydis, and L. Sanguinetti, "Massive MIMO networks: Spectral, energy, and hardware efficiency," *Foundations and Trends® in Signal Processing*, vol. 11, no. 3-4, pp. 154–655, Nov. 2017.

[24] A. H. Jafari *et al.*, "Study on scheduling techniques for ultra dense small cell networks," in *Proc. IEEE VTC-Fall*, Sep. 2015, pp. 1–6.

[25] G. Femenias and F. Riera-Palou, "Fronthaul-constrained cell-free massive MIMO with low resolution ADCs," *IEEE Access*, vol. 8, pp. 116 195–116 215, 2020.

[26] J. Max, "Quantizing for minimum distortion," *IEEE Trans. Inf. Theory.*, vol. 6, no. 1, pp. 7–12, Mar. 1960.

[27] A. Gersho and R. M. Gray, *Vector quantization and signal compression*. Springer Science & Business Media, 2012, vol. 159.

[28] B. Debaillie *et al.*, "A flexible and future-proof power model for cellular base stations," in *IEEE VTC Spring*, May 2015, pp. 1–7.

[29] Ö. T. Demir *et al.*, "Cell-free massive MIMO in O-RAN: Energy-aware joint orchestration of cloud, fronthaul, and radio resources," *IEEE J. Sel. Areas Commun.*, pp. 1–1, 2024.

[30] S. Malkowsky *et al.*, "The world's first real-time testbed for massive MIMO: Design, implementation, and validation," *IEEE Access*, vol. 5, pp. 9073–9088, 2017.

[31] C. Desset and B. Debaillie, "Massive MIMO for energy-efficient communications," in *2016 46th European Microwave Conference (EuMC)*, Oct. 2016, pp. 138–141.

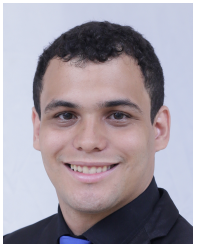
[32] T. Sigwele, A. S. Alam, P. Pillai, and Y. F. Hu, "Energy-efficient cloud radio access networks by cloud based workload consolidation for 5G," *J. Netw. Comput. Appl.*, vol. 78, pp. 1–8, 2017.

[33] 3GPP , *Study on channel model for frequencies from 0.5 to 100 GHz*, 2019, 3GPP TR 38.901 (Release 16).

[34] 3GPP, *NR; User Equipment (UE) radio transmission and reception; Part 1: Range 1 Standalone*, 2021, 3GPP TR 38.101-1 (Release 17).

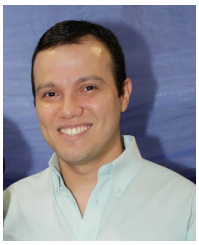
Marx M. M. Freitas received the B.Sc., M.Sc., and Ph.D. degrees in electrical engineering from the Federal University of Pará (UFPA), Belém, Brazil, in 2018, 2019 and 2024, respectively. From 2022 to 2023, he was a visiting Ph.D. student with the Scuola Superiore Sant'Anna, Pisa, Italy. He has experience in broadband communication systems. His research interests include signal processing, resource allocation, massive MIMO and mobile transport networks for future wireless communication systems.

Daynara Dias Souza received a B.Sc. in electrical engineering, in 2018, and a M.Sc. in electrical engineering with emphasis on telecommunication, in 2020, from Federal University of Pará (UFPA), Belém, Brazil, where she is currently working toward the Ph.D. degree in partnership with the School of Energy Systems, LUT University, Finland. Her research interests include signal processing, resource allocation, and the development of emergency communication networks utilizing cell-free massive MIMO systems.



André L. P. Fernandes received the B.S., M.S., and Ph.D., degrees in electrical engineering from the Federal University of Pará, Belém, Brazil, in 2018, 2019, and 2024, respectively. He was a visiting researcher with the Chalmers University of Technology, Gothenburg, Sweden, from 2022 to 2023. He has experience in techno-economic modeling for fixed and mobile broadband communication systems, as well as reliability analysis for communication systems. His research interests include enabling technologies for future wireless communication systems, such as massive MIMO, mobile transport infrastructure, and network deployment analysis and optimization.

Prof. Daniel Benevides da Costa received the B.Sc. degree in Telecommunications from the Military Institute of Engineering (IME), Rio de Janeiro, Brazil, in 2003, and the M.Sc. and Ph.D. degrees in Electrical Engineering, Area: Telecommunications, from the University of Campinas, SP, Brazil, in 2006 and 2008, respectively. His Ph.D. thesis was awarded the Best Ph.D. Thesis in Electrical Engineering by the Brazilian Ministry of Education (CAPES) at the 2009 CAPES Thesis Contest. Prof. Daniel B. da Costa is currently Distinguished University Professor at the Department of Electrical Engineering, King Fahd University of Petroleum & Minerals (KFUPM), Saudi Arabia. He is also Editor-in-Chief of the IEEE Communications Letters and Specialty Chief Editor of the Frontiers in Communications and Networks - Wireless Communications Section. He has been recognized as World's Top 2% Scientist by Stanford University (2021-2024) and has been ranked among 1% Top Scientists in the world in the broad field of Electronics and Electrical Engineering (2022, 2023). He is the recipient of six conference paper awards, and he is a Distinguished Speaker of the IEEE Vehicular Technology Society.



André Mendes Cavalcante received the Ph.D. degree in electrical engineering from the Federal University of Pará (UFPA), Brazil, in 2007. From 2007 to 2016, he was a coworker with the Nokia Institute of Technology (INDT) in Brazil, where he was involved in several research and development projects related to wireless communications. He is currently a senior researcher at Ericsson Research, Brazil. His research interest includes distributed massive MIMO for future wireless communication systems.

Luca Valcarenghi is a Full Professor at the Scuola Superiore Sant'Anna of Pisa, Italy, since 2014. He received the Laurea in Electrical Engineering in 1997 from Politecnico di Torino and the M.S.E.E. and Ph.D. in Electrical Engineering Major Telecommunications from UTD in 1999 and 2001, respectively. He published more than three hundred papers in International Journals and Conference Proceedings. Dr. Valcarenghi received a Fulbright Research Scholar Fellowship in 2009 and a JSPS "Invitation Fellowship Program for Research in Japan (Long Term)" in 2013. His main research interests are optical networks design, analysis, and optimization; communication networks reliability; energy efficiency in communications networks; optical access networks; zero touch network and service management; 5G technologies and beyond.

João C. Weyl Albuquerque Costa received a B.Sc. in electrical engineering from the Federal University of Pará (UFPA), Belém, Brazil, in 1981, an M.Sc. in electrical engineering from the Pontifical Catholic University of Rio de Janeiro, Rio de Janeiro, Brazil, in 1989, and a Ph.D. in electrical engineering from the State University of Campinas, Campinas, Brazil, in 1994. He is currently a professor with the Institute of Technology, UFPA, and a researcher with the Brazilian Research Funding Agency National Council for Scientific and Technological Development, Brasília, Brazil. His current research interests include broadband systems and optical sensors.