Hybrid Receive Combining for Scalable Cell-Free Massive MIMO with Multiple CPUs and Latency-Constrained Midhaul Links

Leonard Paul Schulz, Gerhard Bauch
Institute of Communications
Hamburg University of Technology (TUHH)
Hamburg, Germany
{leonard.schulz, bauch}@tuhh.de

***Abstract*—We propose a new way of operating a cell-free massive multiple-input multiple-output (MIMO) system with multiple central processing units (CPUs). In scalable cell-free massive MIMO systems, each user equipment (UE) is served by a cluster of access points (APs). The required coordination of the APs is often modeled to be done at a single cloud CPU, but in practice this cloud CPU consists of many physical entities which can be distributed over a large geographical area. Contrary to other works, in this work we explicitly model the cloud CPU as multiple physical CPUs, where each AP is connected to one of these CPUs via a low-latency fronthaul link. The CPUs are connected via midhaul links which may have a much higher latency. Hence, extensive cooperation between APs connected to the same CPU is possible while cooperation between APs connected to different CPUs is limited. We outline a joint initial access, pilot assignment, and clustering procedure in such a network. Further, we propose a new hybrid receive combining approach that takes advantage of this structure and allows for a scalable implementation of the system. We compare the performance of our proposed hybrid cell-free system with multiple CPUs to both the idealized centralized cloud CPU version and to the fully distributed cell-free system. Our results show that the proposed system can achieve a performance close to the idealized cloud CPU system while being practically feasible. Furthermore, our hybrid approach substantially outperforms a cellular Distributed Antenna System (DAS) with no CPU cooperation.**

***Index Terms*—5G, 6G, cell-free massive MIMO, uplink operation, scalable implementation, multiple CPUs, receive combining**

I. Introduction

Cell-free massive multiple-input multiple-output (MIMO) is emerging as a key technology for next-generation wireless networks, offering significant improvements in coverage, spectral efficiency (SE), and user fairness [1]. Unlike conventional cellular architectures, where each user equipment (UE) is served by a single base station, cell-free massive MIMO deploys a large number of distributed access points (APs) that jointly serve the UEs. In the canonical cell-free massive MIMO system, all APs serve all UEs, and the transmission is coordinated by a single central processing unit (CPU) connected to all APs via fronthaul links. However, a practical cell-free massive MIMO network must be scalable, ensuring that fronthaul traffic and computational complexity remain bounded as the network expands. The concept of scalability was first introduced in [2]. In that work, each UE is served by a cluster of APs, and coordination is distributed across multiple CPUs, as shown in Fig. 1. A common simplification is to model the CPUs as a single virtual cloud CPU that is connected to all APs via fronthaul links (see e.g. [3]). The implicit assumption here is that this cloud CPU can be implemented in practice by deploying multiple physical CPUs that are interconnected via high-capacity and low-latency midhaul links. However, more recent works suggest that a real-world implementation of cell-free massive MIMO with technology at hand today, such as one leveraging the Open Radio Access Network (O-RAN), requires a special interface for inter-CPU coordination [4]. This interface may introduce non-negligible latency and limited capacity on the midhaul links, making the explicit introduction of multiple CPUs into the system model necessary and the classical centralized operation infeasible.

A. Related Work

The body of literature addressing cell-free massive MIMO with multiple CPUs remains rather limited. Recent research has begun to explore the virtualization of CPU pools, where multiple physical CPUs are interconnected via high-capacity, low-latency midhaul links to form a cloud-based CPU. For instance, [5] investigates load balancing in such architectures, while [6] examines their energy efficiency.

Several other works recognize the inherent challenges of inter-CPU coordination and propose strategies to minimize or avoid it. In [7], a non-cooperative architecture is analyzed, though it does not fully realize the benefits of a cell-free network due to persistent cell-edge effects at CPU boundaries - a limitation also observed in related non-cell-free literature, such as [8]. To reflect more realistic deployment scenarios, while keeping the core ideas of the cell-free approach, several works consider limited cooperation under constrained system models. In [9], a double-star topology with central and edge sites is introduced, aiming to restrict user service to edge sites and thus reduce coordination needs. The study in [10] focuses on interference suppression between CPUs rather than joint user service. Similarly, a power allocation method requiring only minimal inter-CPU information exchange is proposed

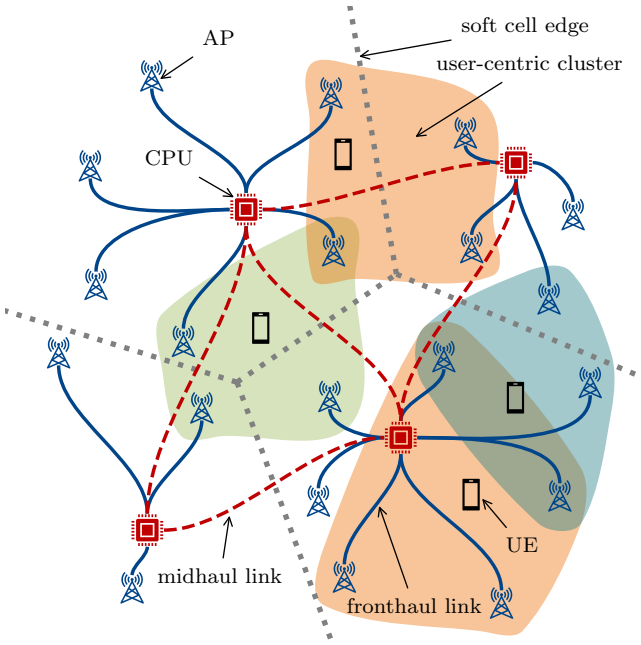


Figure 1: Schematic representation of a scalable cell-free massive MIMO system with multiple CPUs. The APs are grouped into soft cells by their CPU association, but user-centric clustering is still possible.

in [11]. The work in [12] complements these efforts by addressing handover procedures in multi-CPU cell-free setups. The authors of [13] assume low-latency midhaul links between CPUs, but they present a clustering approach that reduces the amount of information exchanged between CPUs with negligible performance degradation.

Closest to our work is the recent contribution in [14], which explores signal processing strategies for an O-RAN-based cell-free massive MIMO system, building on the initial concept introduced in [4]. However, they use a fully distributed combining scheme (as initially proposed in [15]) and their focus lies in channel modeling, clustering, and handover mechanisms. Our work differs by introducing a new hybrid combining scheme that achieves substantial performance gains without additional assumptions on the network topology.

B. Contributions

We present a realistic system model for cell-free massive MIMO networks with multiple CPUs in Sec. II. This includes an approach for joint initial access, pilot assignment, and clustering tailored for our specific system model. We propose a new hybrid combining scheme that makes use of the special structure of the system in Sec. III. To analyze the performance of the new scheme, we perform an extensive numerical evaluation in Sec. IV. Our results show that our hybrid combining scheme achieves substantial performance gains compared to the fully distributed combining scheme and comes very close to the performance of an idealized single-CPU cell-free system that employs the fully centralized scheme.

II. Realistic System Model for multi-CPU Cell-Free Massive MIMO

We consider a cell-free massive MIMO network topology as illustrated in Fig. 1. The sets $\mathcal{L}$, $\mathcal{K}$, and $\mathcal{J}$ denote the APs, UEs, and CPUs, respectively. Each AP is connected to exactly one CPU via a fronthaul link. The CPUs are interconnected via midhaul links.

The fronthaul links are assumed to be of high capacity and low latency (less than $1\,\mathrm{ms}$), while the midhaul links have higher latency (around $10\,\mathrm{ms}$) and potentially lower capacity. These assumptions align with the practical constraints of O-RAN-based deployments, where Open Radio Units (O-RUs) act as APs, Open Distributed Units (O-DUs) serve as CPUs, and midhaul links represent inter-O-DU communication [16].

Link characteristics impose constraints on the type of channel state information (CSI) that can be shared across the network. Instantaneous CSI, i.e., small-scale fading, changes on the order of a few milliseconds and can therefore only be shared over the fronthaul links, but they are unsuitable for transmission over the higher-latency midhaul links. In contrast, statistical CSI, which captures large-scale effects such as path loss and shadowing, evolves several orders of magnitude more slowly and can be shared across CPUs via the midhaul links.

Given these constraints, APs are grouped into static groups per CPU, that we call soft cells, as indicated by the gray dashed lines in Fig. 1. Still, in the cell-free paradigm, each UE is served by a dynamic cluster of nearby APs, which may span multiple CPUs. Clusters serving different UEs can overlap, maintaining the core idea of user-centric service despite the underlying CPU-based structure.

A. Transmission Model

For the wireless transmission channel, we consider the standard block fading model with a coherence block length of $\tau_c = \tau_p + \tau_u + \tau_d$ symbols, where τ_p symbols are used for pilot transmission, τ_u symbols for uplink data transmission, and τ_d symbols for downlink data transmission. The channel between the N-antenna AP l and the single-antenna UE k is denoted by $\mathbf{h}_{kl} \in \mathbb{C}^N$ and modeled as spatially correlated Rayleigh fading, i.e., $\mathbf{h}_{kl} \sim \mathcal{CN}(0, \beta_{kl}\mathbf{R}_{kl})$. The matrix $\mathbf{R}_{kl} \in \mathbb{C}^{N\times N}$ is normalized so that $\mathrm{tr}(\mathbf{R}_{kl}) = N$ and models the spatial correlation, while the coefficient β_{kl} represents the large-scale fading between AP l and UE k.

B. Channel State Information Acquisition

In principle, each AP can measure the local CSI to all surrounding UEs. However, to ensure scalability, each AP only acquires CSI to a limited set of UEs. For statistical CSI each AP l considers UEs from the set $\mathcal{K}_l^{\mathrm{stat}}$, which is assigned by the network as described in Sec. II-C. Following standard assumptions, these quantities vary slowly and can be reliably estimated [17]. In summary, each AP has perfect knowledge of β_{kl} and $\mathbf{R}_{kl}$ for all $k \in \mathcal{K}_l^{\mathrm{stat}}$.

Instantaneous CSI is acquired for a smaller set $\mathcal{K}_l^{\mathrm{serv}} \subseteq \mathcal{K}_l^{\mathrm{stat}}$, containing only the users actively served by AP l. Since the instantaneous CSI changes quickly, it must be estimated

once per coherence block. For this, during the pilot phase, each UE transmits one of τ_p orthogonal pilot sequences. To estimate the channel AP l correlates its received signal with the pilot assigned to UE k and applies the standard minimum mean square error (MMSE) estimator, obtaining the channel estimate $\widehat{\mathbf{h}}_{kl} = \mathbf{h}_{kl} + \tilde{\mathbf{h}}_{kl}$. When using the unbiased MMSE estimator, the estimation error $\tilde{\mathbf{h}}_{kl}$ is Gaussian distributed with zero mean and covariance matrix

$$\mathbf{C}_{kl} = \mathbb{E}\left\{\tilde{\mathbf{h}}_{kl}\tilde{\mathbf{h}}_{kl}^{\mathsf{H}}\right\} \tag{1}$$

which can be analytically computed as described in [17, Sec. 4.2].

C. Initial Access, Pilot Assignment, and Clustering

In the following, we present a version of the joint procedure for initial access, pilot assignment, and clustering proposed in [3] that is adjusted to our multi-CPU system model. As it enters the network, each UE needs to be assigned a pilot index and a set of APs that will serve it. Assume a number of UEs are already connected to the network, and now a new UE k wants to connect. For initial access, it first searches for broadcast synchronization signals from nearby APs that are transmitted periodically. It then selects the AP l^* with the strongest channel as its *master* AP, i.e.,

$$l^* = \arg\max_{l \in \mathcal{L}} \left\{\beta_{kl}\right\}, \tag{2}$$

and performs a standard random access procedure to establish contact with it.

Once contact is established, the master AP l^* compares the large-scale fading coefficient of the new UE k to those of the UEs it is measuring the statistics of (the set $\mathcal{K}_l^{\text{stat}}$), and assigns it the pilot index t_k that minimizes the local pilot contamination, i.e.,

$$t_k = \arg\min_{t \in \{1,\ldots,\tau_p\}} \sum_{\substack{k' \in \mathcal{K}_l^{\text{stat}} \\ t_{k'} = t}} \beta_{k'l}. \tag{3}$$

The master AP l^* then informs its CPU j^* about the new UE k and its assigned pilot index t_k. The CPU j^* then forwards this information to all of its neighboring CPUs, i.e. the set of CPUs $\mathcal{J}_{j^*}$ that have a single-hop midhaul link to CPU j^*. Each of these CPUs then informs each of its APs to add the new UE to their measuring set of statistics, i.e.

$$\mathcal{K}_l^{\text{stat}} \leftarrow \mathcal{K}_l^{\text{stat}} \cup \{k\}, \quad \forall l \in \mathcal{L}_j, j \in \mathcal{J}_{j^*}. \tag{4}$$

Since all APs connected to the same CPU j measure the same set of UEs k, we also refer to this set as $\mathcal{K}_j^{\text{stat}}$.

Finally, each AP l adds UE k to its serving set $\mathcal{K}_l^{\text{serv}}$ if there is no other UE k' in its measuring set $\mathcal{K}_l^{\text{stat}}$ that uses the same pilot index t_k and has a stronger channel.

The entire procedure is repeated for each new UE attempting to connect. The key difference to the original version in [3] is that we explicitly restrict the set of UEs that each AP is aware of. Our restriction ensures that cooperation is confined to neighboring CPUs and that only a single midhaul hop is required for exchange of statistical CSI for each UE.

III. New Hybrid Receive Combining Scheme

In this section, we present our proposed hybrid receive combining scheme for the uplink transmission in a cell-free massive MIMO system with multiple CPUs. The scheme works in two steps: First, each CPU designs a local receive combining vector for each UE that maximizes the signal-to-interference-plus-noise ratio (SINR) using an approximate MMSE approach. Here, each CPU can make use of the locally available instantaneous CSI. Then all involved CPUs forward their signal estimates to the master CPU, which performs a fusion step, also known as large-scale fading decoding (LSFD), to obtain the final signal estimate. The final fusion step is based on the statistical CSI from all involved CPUs, but instantaneous CSI is not required. Our scheme is "hybrid" in the sense that it is neither fully distributed and nor fully centralized as compared in [15], but rather a combination of both.

A. Local Receive Combining at each involved CPU

As a result of the procedure described in Sec. II-C, each UE k is assigned a set of APs that it is served by. We denote this set by $\mathcal{L}_k$. We denote the set of APs that are connected to CPU j and serve UE k as $\mathcal{L}_{jk}$, i.e.,

$$\mathcal{L}_{jk} = \mathcal{L}_k \cap \mathcal{L}_j. \tag{5}$$

These APs simply forward their received signal vectors to the CPU via the fronthaul links, and we can interpret the concatenation of their channels as a MIMO channel with receive signal given by

$$\mathbf{y}_{jk} = \sum_{k' \in \mathcal{K}} \mathbf{h}_{k'\mathcal{L}_{jk}} s_{k'} + \mathbf{n}_{\mathcal{L}_{jk}} \in \mathbb{C}^{N|\mathcal{L}_{jk}|}, \tag{6}$$

where $\mathbf{h}_{k\mathcal{L}_{jk}} = [\mathbf{h}_{kl}]_{l \in \mathcal{L}_{jk}}$ is the concatenation of the channels from all APs in $\mathcal{L}_{jk}$ to UE k, s_k is the transmit signal of UE k, and $\mathbf{n}_{\mathcal{L}_{jk}} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}_{|\mathcal{L}_{jk}|})$ is the noise vector at CPU j experienced by UE k. The target now is to compute a linear receive combining vector $\mathbf{v}_{jk}$ to obtain an estimate $\widehat{s}_k$ of the transmitted signal s_k so that

$$s_k \approx \widehat{s}_{jk} = \mathbf{v}_{jk}^{\mathsf{H}} \mathbf{y}_{jk} = \mathbf{v}_{jk}^{\mathsf{H}} \sum_{k' \in \mathcal{K}} \mathbf{h}_{k'\mathcal{L}_{jk}} s_{k'} + \mathbf{v}_{jk}^{\mathsf{H}} \mathbf{n}_{\mathcal{L}_{jk}}. \tag{7}$$

A full MMSE approach to compute the linear receive combining vector $\mathbf{v}_{jk}$ would require the CPU j to have access to the CSI of all interfering UEs in the network, but due to the procedure described in Sec. II-C the CPU j is only aware of the UEs in the set $\mathcal{K}_j^{\text{stat}}$. In this case, the MMSE receive combining vector can be approximated (as proposed in [3]) as

$$\mathbf{v}_{jk} = \arg\min_{\mathbf{v}_{jk}} \mathbb{E}\left\{\left\|s_k - \widehat{s}_{jk}\right\|^2\right\} \approx$$

$$p_k \left(\sum_{k' \in \mathcal{K}_j^{\text{stat}}} \left(p_{k'} \widehat{\mathbf{h}}_{k'\mathcal{L}_{jk}} \widehat{\mathbf{h}}_{k'\mathcal{L}_{jk}}^{\mathsf{H}} + \mathbf{C}_{k'\mathcal{L}_{jk'}} \right) + \sigma^2 \mathbf{I}_{|\mathcal{L}_{jk}|} \right)^{-1} \widehat{\mathbf{h}}_{k\mathcal{L}_{jk}}, \tag{8}$$

where p_k is the transmit power of UE k and $\mathbf{C}_{k'\mathcal{L}_{jk'}}$ is the block-diagonal matrix obtained by concatenating the individual estimation error covariance matrices $\mathbf{C}_{k'l}$ from each of the

APs in $\mathcal{L}_{jk}$. The set $\mathcal{L}_{jk}$ is the set of APs that are connected to CPU j and serve UE k, however this does not mean that each of these APs also serves all the considered interfering UEs. Hence, it is possible that a UE $k' \in \mathcal{K}_j^{\text{stat}}$ exists whose channel is not estimated by AP l, i.e., $k' \notin \mathcal{K}_l^{\text{serv}}$. In this case, we set the corresponding channel estimate to its expectation

$$\widehat{\mathbf{h}}_{k'l} = \mathbb{E}\{\mathbf{h}_{k'l}\} = \mathbf{0}. \tag{9}$$

Then, the covariance of the estimation error is given by

$$\mathbf{C}_{k'l} = \mathbb{E}\left\{\tilde{\mathbf{h}}_{k'l}\tilde{\mathbf{h}}_{k'l}^{\mathsf{H}}\right\} = \mathbb{E}\{\mathbf{h}_{k'l}\mathbf{h}_{k'l}^{\mathsf{H}}\} = \beta_{k'l}\mathbf{R}_{k'l}, \tag{10}$$

and is known to the CPU j since it is measured as part of the statistical CSI. In principle, it would be possible to also estimate the instantaneous CSI of the interfering UEs at the cost of some more computational effort and fronthaul traffic. However, the clustering algorithm from Sec. II-C forces all UEs whose instantaneous CSI can be estimated well to be actively served. Hence, the performance gains from canceling the interference from unserved UEs would be very small.

B. Large Scale Fading Decoding at the Master CPU

As a result of the local receive combining step described above, each CPU j produces a signal estimate $\widehat{s}_{jk}$ that is then forwarded to the master CPU j^* of UE k. Futhermore, the CPUs forward their known statistical CSI of UE k to the master CPU j^*. There, a final fusion step, also known as LSFD [17, Sec. 5.2], is performed to obtain the final signal estimate $\widehat{s}_k$ according to the weighted sum

$$s_k = \sum_{j \in \mathcal{J}_{j^*}} a_{jk}\widehat{s}_{jk}, \tag{11}$$

where $\mathcal{J}_{j^*}$ is the set of all CPUs that are involved in the transmission of UE k. Here it is useful to write the effective channel of signal s_k from the perspective of the master CPU j^* as

$$\mathbf{g}_{kk'} = \left[\mathbf{v}_{jk}^{\mathsf{H}}\mathbf{h}_{k'\mathcal{L}_{jk}}\right]_{j \in \mathcal{J}_{j^*}} \tag{12}$$

Then, the LSFD vector $\mathbf{a}_k = [a_{jk}]_{j \in \mathcal{J}_{j^*}}$ is computed as

$$\mathbf{a}_k = p_k \left(\sum_{k' \in \mathcal{K}_j^{\text{stat}}} p_{k'} \mathbb{E}\{\mathbf{g}_{kk'}\mathbf{g}_{kk'}^{\mathsf{H}}\} + \sigma^2 \mathbf{V}_k \right)^{-1} \mathbb{E}\{\mathbf{g}_{kk}\} \tag{13}$$

where σ^2 is the noise power and $\mathbf{V}_k$ is a diagonal matrix of the norms of the combining vectors of the CPUs, i.e.,

$$\mathbf{V}_k = \operatorname{diag}\left(\left[\|\mathbf{v}_{jk}\| \right]_{j \in \mathcal{J}_{j^*}} \right) \tag{14}$$

C. Achievable Spectral Efficiency

Adapting the Use-and-then-Forget (UatF) bound from [17, Th. 5.4] to our notation, an achievable SE for UE k is given by

$$\text{SE}_k = \frac{\tau_u}{\tau_c} \log_2\left(1 + \text{SINR}_k\right), \tag{15}$$

where the SINR is given by $\text{SINR}_k =$

$$\frac{p_k \left| \mathbf{a}_k^{\mathsf{H}} \mathbb{E}\{\mathbf{g}_{kk}\} \right|^2}{\mathbf{a}_k^{\mathsf{H}} \left(\sum_{k' \in \mathcal{K}} p_{k'} \mathbb{E}\left\{\mathbf{g}_{kk'}\mathbf{g}_{kk'}^{\mathsf{H}}\right\} - p_k \mathbb{E}\{\mathbf{g}_{kk}\}\mathbb{E}\left\{\mathbf{g}_{kk}^{\mathsf{H}}\right\} + \sigma^2 \mathbf{V}_k \right) \mathbf{a}_k}. \tag{16}$$

IV. Numerical Evaluation

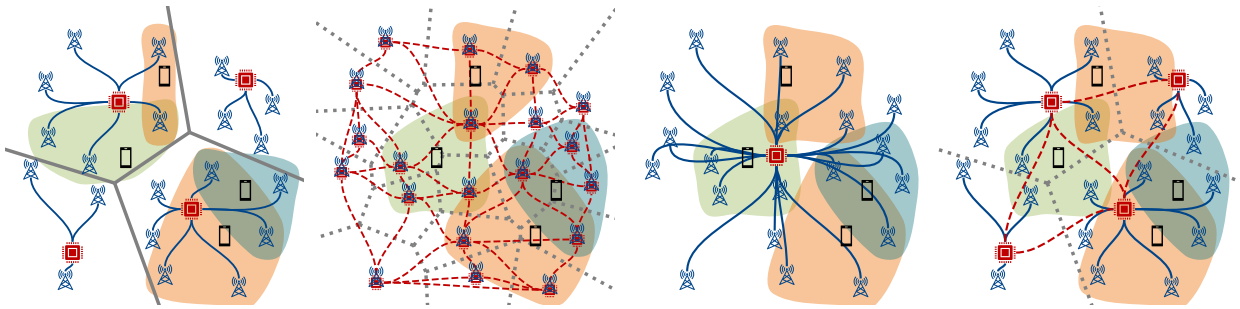
To evaluate the performance of our proposed hybrid receive combining scheme, we conduct numerical simulations of the four different systems shown in Fig. 2. The system shown in Fig. 2a resembles a cellular Distributed Antenna System (DAS) such as in [7], [8], where there is no cooperation between the different CPUs. Therefore, each UE can only be served by a cluster of APs that are connected to the same CPU. The second system, depicted in Fig. 2b, is a fully distributed cell-free system, meaning each AP has an onboard CPU which performs the first step of the receive combining scheme locally. For this system, no long fronthaul links are required, but the APs are only able to cooperate with each other via the midhaul links. The fully centralized cell-free system, shown in Fig. 2c, is an idealized system where all APs are connected to a single CPU and can cooperate with each other via the fronthaul links, which is not practically feasible [4]. Here, no LSFD is needed, as there is only one signal estimate for a given UE k. Finally, the system shown in Fig. 2d is our proposed hybrid system which can be seen as a compromise between the fully distributed and the fully centralized system. Some of the APs are connected to the same CPU and can cooperate with each other via the fronthaul links, but only higher-latency midhaul links are used to connect the CPUs to each other. In contrast to the cellular DAS, the clusters are formed dynamically around the UEs, so only soft cell edges remain.

A. Simulation Parameters

To simulate the four systems, we consider a grid of size $2\,\text{km} \times 2\,\text{km}$ and we sample the positions of the UEs, APs, and CPUs according to a Poisson Point Process (PPP) with density λ_K, λ_L, and λ_J, respectively. In order to avoid boundary effects, we employ the wrap-around technique. Each AP is connected to its closest CPU via a fronthaul link. We determine neighboring CPUs links by computing the Delaunay triangulation of the CPU positions and connect the neighbors via midhaul links. For comparability, we take the commonly used parameters from [15]: The large-scale fading coefficients are computed as

$$\beta_{kl}[\text{dB}] = -30.5 - 36.7 \log_{10}\left(\frac{d_{kl}}{1\,\text{m}}\right) + \text{SF}[\text{dB}], \tag{17}$$

where d_{kl} is the Euclidean distance between UE k and AP l in meters, taking a height difference of $15\,\text{m}$ into account, and the shadow fading is modeled as a log-normal random variable $\text{SF} \sim \mathcal{N}(0, 7.82^2)$. The spatial correlation matrices $\mathbf{R}_{kl}$ are modeled as in [17, Sec. 2.5], assuming a randomly oriented, half-wavelength spaced, uniform linear array with N antennas at each AP and an angular delay spread of $15°$. The channel is modeled as constant over the coherence block length of $\tau_c = 200$ symbols out of which $\tau_p = 10$ are used for pilots. Since we only consider the uplink, we set $\tau_d = 0$ and $\tau_u = 190$. We let each UE transmit its data with the maximum uplink transmit power $p = 100\,\text{mW}$ and its pilots with the maximum pilot transmit power $\eta_k = 100\,\text{mW}$. The uplink noise power is set to $\sigma_{\text{ul}}^2 = -96\,\text{dBm}$.



(a) Cellular DAS. (b) Fully Distributed. (c) Fully Centralized. (d) Hybrid (proposed).

Figure 2: Schematic representation of network setups considered in our evaluation. (Dotted) gray lines represent (soft) cell edges, blue lines are fronthaul links and dashed red lines are midhaul links.

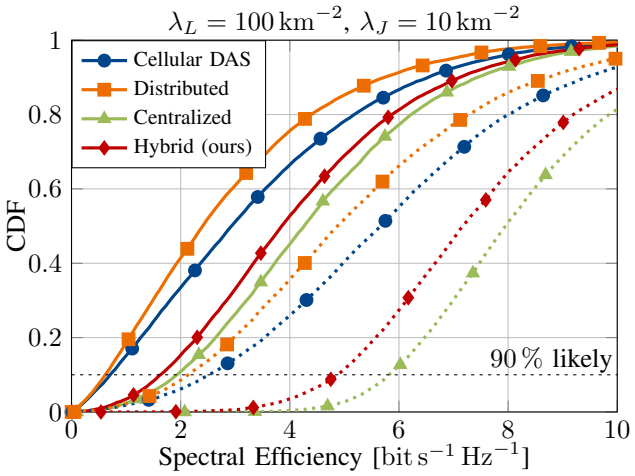


Figure 3: CDF of the SE for the four systems. The solid lines represent a high UE density of $\lambda_K = 80\,\mathrm{km}^{-2}$, while the dotted lines represent a low UE density of $\lambda_K = 20\,\mathrm{km}^{-2}$.

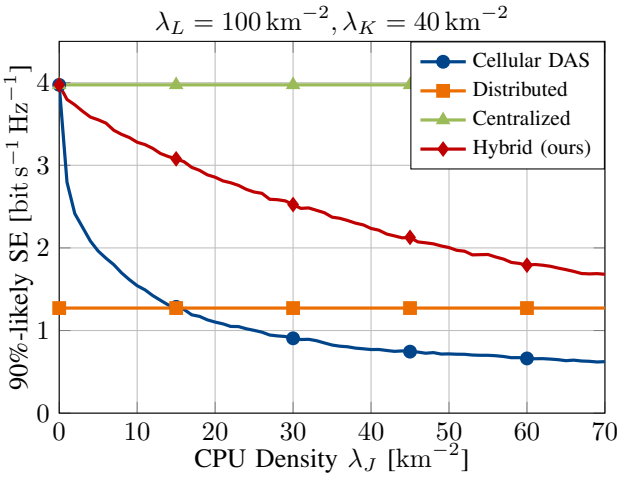


Figure 4: 90%-likely SE at different CPU densities λ_J. The distributed system (one CPU per AP) and centralized system (one CPU for all APs) are not affected by the CPU density parameter and are shown for reference.

B. Performance Comparison of the Four Systems

As a first result, we present the cumulative density function (CDF) of the SE of the four systems in Fig. 3. We set the AP density to $\lambda_L = 100\,\mathrm{km}^{-2}$ and the CPU density to $\lambda_J = 10\,\mathrm{km}^{-2}$, so that each CPU is connected to 10 APs on average. The solid lines represent a challenging scenario with a high UE density of $\lambda_K = 80\,\mathrm{km}^{-2}$, while the dotted lines were generated with a moderate UE density of $\lambda_K = 20\,\mathrm{km}^{-2}$. The expectations in the SINR expression in Eq. 16 are approximated by averaging over 100 channel realizations. To obtain smooth curves, we simulated 40 drops for the high UE density and 160 drops for the low UE density.

The idealized fully centralized system is an upper benchmark for the other systems as it allows full cooperation between all APs, but it is not practically feasible. The three other systems are attempts to make the system more practical at the cost of some performance. The fully distributed system shows large degradations in performance, because the receive combining vectors for the UEs are only designed locally in each AP, so that the spatial diversity is not well exploited. This is a problem especially for the more challenging scenario with a $\lambda_K = 80\,\mathrm{km}^{-2}$, where good spatial separation of UEs is required to achieve a high SINR. The cellular DAS achieves good performance for the most fortunate UEs, because the UEs in the center of a cell are served by a large number of APs that design one joint receive combining vector together, which is similar to the fully centralized system. However, the performance drops significantly for the less fortunate UEs at the cell edges, which are only served by a small number of APs and suffer from a high interference from the neighboring cells. For those UEs, the cellular DAS is even outperformed by the fully distributed system. In contrast, the performance of our hybrid system is very close to the idealized fully centralized system, even for the most unfortunate users at the cell edges, while being practically feasible. This is also true for the challenging scenario with a high UE density. In particular, the hybrid system outperforms

1) the distributed system, because it allows for exchange of instantaneous CSI across multiple APs which enables a superior design of receive combining vectors.
2) the cellular DAS, because it allows for cooperation between CPUs which is crucial to overcome the cell-edge problem.

C. Impact of AP and CPU Densities

An inherent tradeoff in the design of a multi-CPU network is the density of CPUs: The higher the CPU density, the lower the computational load per CPU as it has to process fewer APs. Additionally, the average distance between an AP and its closest CPU decreases, meaning shorter fronthaul links are required. In the extreme case of the fully distributed system, each AP has its own integrated CPU and the fronthaul links are not needed at all. On the other hand, a higher CPU density means that the geographical area covered by each CPU becomes smaller, which can lead to more interference from neighboring CPUs and therefore a lower SE. Furthermore, the number of APs connected to each CPU decreases, so the receive combiner loses some of its degrees of freedom.

A central goal of cell-free massive MIMO is to improve the reliability of service, so we focus on the SE that is achieved by 90% of the UEs in the network, as also marked in Fig. 3. In Fig. 4, we investigate the impact of the deployment densities of the APs and CPUs on the performance of those UEs. The plot shows the 90%-likely SE over the density λ_J. The parameter λ_J only applies to the cellular DAS and the hybrid system, as the fully distributed system always has one CPU per AP and the fully centralized system has only one CPU for all APs. For the minimal CPU density (i.e., only a single CPU), the hybrid system and the cellular DAS are equivalent to the fully centralized system and therefore also achieve the same performance. As λ_J increases, the cellular DAS quickly loses performance and already for a density of $\lambda_J = 16\,\mathrm{km}^{-2}$ it starts delivering less reliable service than the fully distributed system. This shows that interference at the cell edges is the limiting factor for the less fortunate UEs: For $\lambda_J = 16\,\mathrm{km}^{-2}$, on average more than 6 APs are connected to each CPU, giving the cellular DAS a large advantage over the fully distributed system in the design of the receive combining vectors. However, the fully distributed system can form user-centric clusters that are not limited by strict cell edges.

In contrast, our hybrid approach achieves very reliable service even for higher CPU densities, making it superior to the cellular DAS. For the extreme case of very high CPU densities, effectively only one AP is connected to each CPU and then the hybrid system approaches the performance of the fully distributed system. In that way, our hybrid approach can be seen as a compromise between the centralized and the distributed approach, where the higher the CPU density λ_J, the more distributed the system becomes. Notably, even for a high CPU density of $\lambda_J = 33\,\mathrm{km}^{-2}$ (i.e., on average 3 APs per CPU), the hybrid system achieves a 90%-likely SE of $2.45\,\mathrm{bit\,s}^{-1}\,\mathrm{Hz}^{-1}$ compared to $1.27\,\mathrm{bit\,s}^{-1}\,\mathrm{Hz}^{-1}$ for the fully distributed system. This means by using our hybrid approach, an *almost* fully distributed system achieves 93% better service for the less fortunate UEs than the classical fully distributed cell-free system.

V. Conclusion

In this work we investigated a practically feasible cell-free massive MIMO system with multiple CPUs and latency-constrained midhaul links. Tailored to this system model, we proposed a new hybrid receive combining scheme that strikes a balance between the well-known distributed and centralized schemes. Our hybrid scheme outperforms other practically feasible schemes, such as fully distributed cell-free massive MIMO and cellular DAS, making it a promising candidate for real-world deployments.

References

[1] H. Q. Ngo, A. Ashikhmin, H. Yang, E. G. Larsson, and T. L. Marzetta, "Cell-free massive MIMO versus small cells," *IEEE Transactions on Wireless Communications*, vol. 16, no. 3, pp. 1834–1850, Mar. 2017.

[2] G. Interdonato, P. Frenger, and E. G. Larsson, "Scalability aspects of cell-free massive MIMO," in *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, May 2019, pp. 1–6.

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

[4] V. Ranjbar, A. Girycki, M. A. Rahman, S. Pollin, M. Moonen, and E. Vinogradov, "Cell-free mMIMO support in the o-RAN architecture: A PHY layer perspective for 5g and beyond networks," *IEEE Communications Standards Magazine*, vol. 6, no. 1, pp. 28–34, Mar. 2022.

[5] F. Göttsch, G. Caire, W. Xu, and M. Schubert, *User-centric cell-free wireless networks for 6g: Communication theoretic models and research challenges*, Jan. 12, 2024.

[6] Ö. T. Demir, M. Masoudi, E. Björnson, and C. Cavdar, "Cell-free massive MIMO in o-RAN: Energy-aware joint orchestration of cloud, fronthaul, and radio resources," *IEEE Journal on Selected Areas in Communications*, vol. 42, no. 2, pp. 356–372, Feb. 2024.

[7] F. Riera-Palou and G. Femenias, "Decentralization issues in cell-free massive MIMO networks with zero-forcing precoding," in *2019 57th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, Sep. 2019, pp. 521–527.

[8] S. Schwarz, "Dynamic distributed antenna systems: A transitional solution for CRAN implementation," in *2018 IEEE Globecom Workshops (GC Wkshps)*, Dec. 2018, pp. 1–7.

[9] T. Murakami, N. Aihara, A. Ikami, Y. Tsukamoto, and H. Shinbo, "Analysis of CPU placement of cell-free massive MIMO for user-centric RAN," in *NOMS 2022-2022 IEEE/IFIP Network Operations and Management Symposium*, Apr. 2022, pp. 1–7.

[10] A. Ikami, N. Aihara, Y. Tsukamoto, T. Murakami, and H. Shinbo, "Cooperation method between CPUs in large-scale cell-free massive MIMO for user-centric RAN," *IEEE Access*, vol. 11, pp. 95 267–95 277, 2023.

[11] S. Kim, S. Ahn, J. Park, J. Youn, Y. Kwon, and S. Cho, "CPU-cooperative power control scheme for scalable cell-free massive MIMO systems," *IEEE Transactions on Wireless Communications*, pp. 1–1, 2024.

[12] S. Kim and S. Cho, "Performance analysis of handover algorithms in scalable cell-free massive MIMO systems," in *2024 15th International Conference on Information and Communication Technology Convergence (ICTC)*, Oct. 2024, pp. 163–168.

[13] M. M. M. Freitas, D. D. Souza, D. B. d. Costa, *et al.*, "Reducing inter-CPU coordination in user-centric distributed massive MIMO networks," *IEEE Wireless Communications Letters*, vol. 12, no. 6, pp. 957–961, Jun. 2023.

[14] R. Beerten, V. Ranjbar, K. A. P. Guevara, and S. Pollin, "Mobile cell-free massive MIMO: A practical o-RAN-based approach," *IEEE Open Journal of the Communications Society*, vol. 6, pp. 593–610, 2025.

[15] E. Björnson and L. Sanguinetti, "Making cell-free massive MIMO competitive with MMSE processing and centralized implementation," *IEEE Transactions on Wireless Communications*, vol. 19, no. 1, pp. 77–90, Jan. 2020.

[16] R. Beerten, A. Girycki, and S. Pollin, "User centric cell-free massive MIMO in the o-RAN architecture: Signalling and algorithm integration," in *2022 IEEE Conference on Standards for Communications and Networking (CSCN)*, Nov. 2022, pp. 181–187.

[17] Ö. T. Demir, E. Björnson, and L. Sanguinetti, *Foundations of User-Centric Cell-Free Massive MIMO*. Feb. 22, 2022.