Cell-Free Massive MIMO With Sequential Fronthaul Architecture and Limited Memory Access Points

Vida Ranjbar, *Graduate Student Member, IEEE*, Robbert Beerten, *Graduate Student Member, IEEE*, Marc Moonen, *Fellow, IEEE*, and Sofie Pollin, *Senior Member, IEEE*

***Abstract*— Cell-free massive multiple-input multiple-output (CFmMIMO) is a paradigm that can improve users' spectral efficiency (SE) far beyond traditional cellular networks. Increased spatial diversity in CFmMIMO is achieved by spreading the antennas into small access points (APs), which cooperate to serve the users. Sequential fronthaul topologies in CFmMIMO, such as the daisy chain and multi-branch tree topology, have gained considerable attention recently. In such a processing architecture, each AP must store its received signal vector in the memory until it receives the relevant information from the previous AP in the sequence to refine the estimate of the users' signal vector in the uplink. In this paper, we adopt vector-wise and element-wise compression on the raw or pre-processed received signal vectors to store them in the memory. We investigate the impact of the limited memory capacity in the APs on the optimal number of APs. We show that with no memory constraint, having single-antenna APs is optimal, especially as the number of users grows. However, a limited memory at the APs restricts the depth of the sequential processing pipeline. Furthermore, we investigate the relation between the memory capacity at the APs and the rate of the fronthaul link.**

***Index Terms*— Uplink cell-free massive MIMO, sequential fronthaul, sequential processing, limited memory capacity.**

I. Introduction

MASSIVE multiple-input multiple-output (mMIMO) technology is one of the critical enablers for ambitious future wireless communication networks. It enhances users' spectral efficiency (SE), system energy efficiency (EE), and reliability with low-cost hardware at both receiver and transmitter [2]. Massive MIMO makes it possible to separate the users spatially rather than using time/frequency orthogonalization, thanks to the spatial diversity it brings. Reusing the time/frequency resources increases the average throughput of users in the network. Cell-free massive multiple-input multiple-output (CFmMIMO) is a new paradigm that can further improve the SE of the users. In CFmMIMO, the antennas are spread into many small access points (APs) rather than being collocated in large base stations, as in traditional cellular networks. The APs cooperate to serve the nearby users, which mitigates the adverse effect of large-scale fading on the users' signal-to-interference-plus-noise-ratio (SINR) and provides uniform service to all the users [3], [4]. In other words, in a distributed network with the antennas distributed in small APs where multiple APs are serving each user, the probability of having a poor channel gain is lower than in a traditional cellular and small-cell system where only one AP serves each user [3], [4], a consequence of the so-called macro diversity phenomenon.

Initially, a CFmMIMO network was conceived as having multiple distributed APs, each with one/multiple antennas, randomly distributed in an area and cooperatively serving all the users [3], [4], [5]. Two main operational methods were introduced in the uplink. First, the APs can locally process their received signal vector and estimate the users' signal vector. Then, the local estimates are sent to a central processing unit (CPU) to estimate the users' signal vector globally. Second, the APs can act as relays, forwarding their received signal vector and local channel estimates to the CPU. The CPU then solves a network-wide optimization problem to estimate the users' signal vector.

Recently, sequential fronthaul has been introduced in CFmMIMO networks in which the estimate of the users' signal vector is refined through the sequential fronthaul [6], [7], [8], [9], [10], possibly without an intermediate CPU. In such a sequential fronthaul, the APs along the sequence need to store their raw/pre-processed received signal vector until they receive the estimate of users' signal vector from the previous APs in the sequence to be able to refine them. Thus, the delay of the final estimation of the users' signal vector and the memory available at the last AP becomes a bottleneck. On-chip cache memory can be a good candidate for the memory requirement at the APs. On-chip caches are fast,

Manuscript received 24 July 2023; revised 14 February 2024 and 8 June 2024; accepted 25 June 2024. Date of publication 2 July 2024; date of current version 19 December 2024. This work is supported by European Union's Horizon 2020 research and innovation program under grant agreements: 101013425 (REINDEER) and 101017171 (MARSAL) and from Research Foundation—Flanders (FWO) under project number G0C0623N. The resources and services used in this work were provided by the VSC (Flemish Supercomputer Center), funded by the Research Foundation—Flanders (FWO) and the Flemish Government. An earlier version of this paper was presented in part at the IEEE Global Communications Conference (Globecom), 2023 [DOI: 10.1109/GCWkshps58843.2023.10465118]. The associate editor coordinating the review of this article and approving it for publication was G. Alexandropoulos. *(Corresponding author: Vida Ranjbar.)*

The authors are with the Department of Electrical Engineering (ESAT), KU Leuven, 3001 Leuven, Belgium (e-mail: vida.ranjbar@kuleuven.be; robbert.beerten@kuleuven.be; marc.moonen@kuleuven.be; sofie.pollin@kuleuven.be).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCOMM.2024.3422184.

Digital Object Identifier 10.1109/TCOMM.2024.3422184

0090-6778 © 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: KU Leuven Libraries. Downloaded on January 29,2025 at 13:57:53 UTC from IEEE Xplore. Restrictions apply.

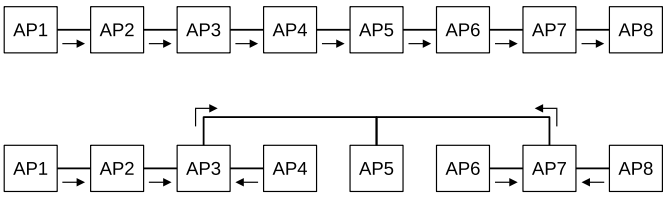


Fig. 1. Data flow in daisy chain (Top) vs multi-branch tree (Bottom) fronthaul topology.

energy-efficient, and usually of low capacity, making efficient usage of their capacity important. Storing the local received signal vector in a limited capacity memory may impose a compression noise on the stored received signal vector, especially at the APs toward the end of the sequence due to the large number of received signal vectors to be stored. As a result, the users' signal estimation quality will be adversely affected. This paper studies the impact of low-capacity memory on the average per-user SE in the uplink of a CFmMIMO, with the assumption of perfect channel state information (CSI), using either vector-wise or element-wise compression to store the received signal vectors in the memory. The fronthaul is a daisy chain, and each AP along the fronthaul refines users' signal estimates sequentially, starting from first AP to the last AP. The limited memory problem in a daisy chain fronthaul is discussed in subsection II-B. This paper considers sequential fronthaul similar to Fig. 1, which illustrates the daisy chain vs. multi-branch tree fronthaul topology for eight APs. Suppose that the processing level corresponds to the level of the tree in which a reference AP resides. The processing level in a given tree structure determines the number of the stored received signal vectors. In a daisy-chain topology, each AP has a distinct processing level. However, in the multi-branch tree in Fig. 1, multiple APs can be on the same processing level. For example, in Fig. 1, AP 1 is at the first processing level, AP 2, AP 4, AP 6 and AP 8 are at the second processing level, AP 3 and AP 7 are the third processing level, and finally AP 5 is at the forth (last) processing level.

A. Related Works

Aside from limited capacity memory, non-idealities such as limited bandwidth fronthaul links, hardware impairments, and low-resolution analog to digital converters (ADC) in (CF) mMIMO networks affect the users' signal vector estimation quality. The impact of such non-idealities on the users' SE and EE is discussed in [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21], and [22], among others. The authors in [12] show in which scenarios the correlation between the distortion vector element in a massive MIMO network with hardware impairment has a negligible impact on the users' SE. The ADC bit allocation among antennas in a massive MIMO network is discussed in [13], [14], [15], and [16]. In [14], the SE maximization is considered to find the optimal number of ADC bits of different antennas, subject to a constraint on the total number of bits or power consumption. The authors in [16] consider adaptive intra-AP and inter-AP bit allocation for ADCs at the APs. First, the authors consider channel estimation in the uplink and minimize the sum-weighted normalized mean square error (SWNMSE) to find optimal bit allocation among the antennas of one AP (intra-AP) and subsequently among the antennas of different APs (inter-AP). Furthermore, the ADC bits are allocated to the APs during data transmission based on the solution of a sum-SE maximization problem. In general, the solutions to the optimization problems show that in the case of intra-AP bit allocation, equal bit allocation to the antennas is optimal as all the antennas of one AP experience the same large-scale fading. However, when it comes to inter-AP bit allocation, different APs are assigned different numbers of bits based on their overall channel gain to the users. Intuitively, the APs with larger overall channel gains for users are allocated more bits. Furthermore, increasing the number of users reduces the variance of the allocated bits to different APs, as the variance of the received signal between antennas of different APs reduces with increasing users.

B. Contribution

To the best of our knowledge, this paper together with its short conference version [1] are the first to identify the problem of limited memory availability in sequential processing and fronthaul topologies. Our main contributions are then summarized as follows:

- We consider a CFmMIMO network with sequential fronthaul where the APs have limited memory along the sequence. The sequential fronthaul can be a daisy chain or organized in a multi-branch tree.
- We formulate a maximization problem on the upper bound of users' sum-SE to find the optimal per-AP compression noise covariance matrix for both element-wise/ vector-wise compression under a limited memory constraint. This covariance matrix could be used to design optimal compression methods that are not considered here.
- To use the limited memory capacity more efficiently, vector-wise compression is favored over element-wise compression. However, element-wise compression is used in practice due to its lower complexity. To avoid vector-wise compression without compromising performance, we propose pre-processing the local received signal vectors with the principal component analysis (PCA) method to decorrelate the dimension of the received signal vector and then compressing the elements of the pre-processed vector element-wisely.
- We consider two general memory models: 1) fixed per-AP and 2) fixed total memory constraints. We relate the memory capacity at the APs to the maximum rate of the fronthaul links.
- We provide simulation results on the impact of the limited memory at the APs on the optimal number of APs in the network with a daisy chain fronthaul topology. Having a fixed total memory budget, we compare equal or linearly increasing memory allocation schemes to the APs and compare the optimal processing length of the sequential fronthaul in both cases. In addition, we consider a multi-branch tree topology as an alternative fronthaul topology and study the impact of limited memory capacity in scenarios where processing along the branches can be parallelized.

Authorized licensed use limited to: KU Leuven Libraries. Downloaded on January 29,2025 at 13:57:53 UTC from IEEE Xplore. Restrictions apply.

- In contradiction to most existing state-of-the-art, we find that spatial diversity does not come free in a sequential processing topology. We must take into account that APs toward the end of this sequence need to store an exceedingly large number of received signal vectors and thus negate the effect of increased macro diversity by introducing significant compression noise on the stored received signal vectors.

C. Outline

The remainder of this paper is organized as follows. In Section II, the overall system model and the problem is defined. In Section III, the processing and compression of the received signal vector in the memory in each AP is elaborated on. Section IV introduces the memory capacity and allocation models in the considered fronthaul topologies. Furthermore, it elaborates on the maximum rate of the fronthaul links connecting two APs in a daisy chain fronthaul with different memory models. Section V analyses the simulation results, and Section VI concludes the paper.[1]

D. Notation

Vectors and matrices are denoted with boldface lower-case and upper-case letters, respectively. Transpose and conjugate transpose operations are denoted by superscripts $^{\mathrm{T}}$ and $^{\mathrm{H}}$, respectively. For two matrices $\mathbf{A}$ and $\mathbf{B}$, $\mathbf{A} \succeq \mathbf{B}$ means that $\mathbf{A}-\mathbf{B}$ is positive semi-definite. A zero-mean multivariate circularly symmetric complex Gaussian distribution with covariance matrix $\mathbf{X}$ is represented as $\mathcal{CN}(\mathbf{0},\mathbf{X})$. The mean of $\mathbf{x}$ is denoted by $\mathbb{E}\{\mathbf{x}\}$, $\mathcal{H}(\mathbf{x})$ is the differential entropy of $\mathbf{x}$, and $I(\mathbf{x};\hat{\mathbf{x}})$ is the mutual information between $\mathbf{x}$ and $\hat{\mathbf{x}}$. Euclidean norm of $\mathbf{x}$ is shown as $\|\mathbf{x}\|$. Furthermore, $\mathbf{X}=\mathrm{blkdiag}(\mathbf{X}_1,\ldots,\mathbf{X}_L)$ is a block-diagonal matrix with matrices $\mathbf{X}_i,\forall i$ as diagonal blocks. $\mathrm{diag}(\mathbf{x})$ denotes a diagonal matrix with the elements of $\mathbf{x}$ as its diagonal elements and $\mathrm{diag}(\mathbf{X})$ denotes a diagonal matrix with the same diagonal elements as $\mathbf{X}$. $\det(\mathbf{A})$ returns the determinant of the square matrix $\mathbf{A}$. $N\times N$ identity matrix is shown as $\mathbf{I}_N$. $\mathbf{X}^{1/2}$ refers to the square root of matrix $\mathbf{X}$ and superscript $^{\mathrm{H}/2}$ indicates the conjugate transpose of square root.

II. System Model and Problem Statement

Distributed processing is a necessity in CFmMIMO, as it avoids overloading a single AP/CPU with massive computations, and it enables truly scalable implementations and hence large-scale deployments [5], [23], [24]. Distributed processing has become even more attractive with the growing interest in the sequential fronthaul topologies [7], [8], [9], [10]. The multi-hop path between any AP and the CPU in sequential fronthaul enables serializing local processing in each AP.

A. Recursive Least-Squares (RLS) for Uplink Signal Estimation

We consider distributed uplink signal estimation using the RLS method in a CFmMIMO network with daisy chain fronthaul topology, inspired by [8], [25], and [26]. There are L APs, each having N antennas and a limited memory to store the received signal vectors in the uplink. The APs are connected in a daisy chain topology, all jointly serving each of the K users in the uplink. The received signal vector at AP l is defined as follows:

$$\mathbf{y}_l = \mathbf{H}_l\mathbf{s}+\mathbf{n}_l, \tag{1}$$

where $\mathbf{s}\sim\mathcal{CN}(\mathbf{0},p\mathbf{I}_K)$ is the users' signal vector and $\mathbf{n}_l\sim\mathcal{CN}(\mathbf{0},\sigma^2\mathbf{I}_N)$ is the noise vector at AP l. Matrix $\mathbf{H}_l\in\mathbb{C}^{N\times K}$ is the local channel matrix between AP l and the users. Each column of $\mathbf{H}_l$ is a multi-variate circularly symmetric complex Gaussian random vector, distributed as follows:

$$\mathbf{H}_{l[:,k]}\sim\mathcal{CN}(\mathbf{0},\mathbf{R}_{kl}), \tag{2}$$

where subscript $[:,k]$ represent the k^{th} column of $\mathbf{H}_l$ and $\mathbf{R}_{kl}$ is the spatial correlation matrix [27] and $\beta_{kl}=\frac{\mathrm{trace}(\mathbf{R}_{kl})}{N}$. The network-wide channel matrix, received vector, and noise vector are given as $\mathbf{H}=\left[\mathbf{H}_1^T,\ldots,\mathbf{H}_L^T\right]^T$, $\mathbf{y}=\left[\mathbf{y}_1^T,\ldots,\mathbf{y}_L^T\right]^T$ and $\mathbf{n}=\left[\mathbf{n}_1^T,\ldots,\mathbf{n}_L^T\right]^T$, respectively. We assume a block fading model in which the channel matrix $\mathbf{H}$ remains constant in a coherence interval of $\tau_c=B_cT_c$ samples, with T_c and B_c the coherence time and coherence bandwidth of the channel, respectively [28]. Out of τ_c samples in one coherence block, τ_u samples are used for uplink transmission.

AP l stores a compressed version of the received signal vector in the memory. As a result of compression, a noise vector is added to the received signal vector. Section III elaborates on the compression model using rate-distortion theory. For now, assume that the compressed version of $\mathbf{y}_l$ is represented as $\hat{\mathbf{y}}_l$ and is formulated as follows:

$$\hat{\mathbf{y}}_l=\mathbf{y}_l+\mathbf{q}_l=\mathbf{H}_l\mathbf{s}+\mathbf{n}_l+\mathbf{q}_l, \tag{3}$$

where $\mathbf{z}_l=\mathbf{n}_l+\mathbf{q}_l$ is a spatially correlated noise vector with zero mean and covariance matrix $\mathbf{Z}_l=\mathbb{E}\{\mathbf{z}_l\mathbf{z}_l^{\mathrm{H}}\}$. Vector $\mathbf{q}_l$ is the compression noise vector. The network-wide compressed received signal and noise vector can be expressed as $\hat{\mathbf{y}}=\left[\hat{\mathbf{y}}_1^{\mathrm{T}},\ldots,\hat{\mathbf{y}}_L^{\mathrm{T}}\right]^{\mathrm{T}}$ and $\mathbf{z}=\left[\mathbf{z}_1^{\mathrm{T}},\ldots,\mathbf{z}_L^{\mathrm{T}}\right]^{\mathrm{T}}$, respectively. The noise vectors among the APs are assumed to be independent, i.e., $\mathbf{Z}=\mathbb{E}\{\mathbf{z}\mathbf{z}^{\mathrm{H}}\}=\mathrm{blkdiag}(\mathbf{Z}_1,\ldots,\mathbf{Z}_L)$.

The APs sequentially refine the estimate of the users' signal vector based on the local CSI using the RLS algorithm given in Algorithm (1). As stated in Algorithm 1, once per coherence block, AP l exchanges $\mathbf{\Gamma}_l$ with AP $l+1$ and once per each uplink sample, i.e., τ_u times per coherence block, AP l exchanges its local soft estimate of users' signal, i.e., $\hat{\mathbf{s}}_l$, with AP $l+1$ so AP $l+1$ can refine $\hat{\mathbf{s}}_l$ and update it to $\hat{\mathbf{s}}_{l+1}$.

By updating the estimate of the users' signal vector using the RLS method, the estimate of the users' signal vector in the last AP is given as follows:

$$\hat{\mathbf{s}}=\hat{\mathbf{s}}_L=(\mathbf{H}^{\mathrm{H}}\mathbf{Z}^{-1}\mathbf{H}+\frac{1}{p}\mathbf{I}_K)^{-1}\mathbf{H}^{\mathrm{H}}\mathbf{Z}^{-1}\hat{\mathbf{y}}. \tag{4}$$

Superscript n in Algorithm 1 is used to differentiate amongst the different uplink samples. However, for the sake of readability, we remove superscript n from the users' signal vector

[1]Matlab codes used to produce simulation results can be found at: https://gitlab.com/VidaRanjbar/limited-memory-cell-free

Authorized licensed use limited to: KU Leuven Libraries. Downloaded on January 29,2025 at 13:57:53 UTC from IEEE Xplore. Restrictions apply.

TABLE I
PARAMETERS

Parameter Description	Symbols	Parameter Description	Symbol
Number of users	K	Coherence time	T_c
Number of APs	L	Coherence bandwidth	B_c
Number of antennas per AP	N	Number of samples in one Coherence Block	τ_c
Total memory capacity	C_T	Number of subcarriers of an OFDM symbol	N_{sc}
Memory capacity per AP	C_{AP}	OFDM Symbol time	T_s
Memory capacity per subcarrier in each AP	C_{sc}	Total bandwidth	B
User's transmit power	p	Receiver noise power at the APs	σ^2

Algorithm 1 RLS Algorithm for Users' Signal Vector Estimation

1: **Initialize:**
2: $\quad \mathbf{\Gamma}_0 = p\mathbf{I}_K$
3: $\quad \hat{\mathbf{s}}_0^n = \mathbf{0}_{K\times 1}, \forall n \in [1:\tau_u]$
4: **for** $l = 1 \dots L$ **do**
5: $\quad \mathbf{\Gamma}_l = \mathbf{\Gamma}_{l-1} - \mathbf{\Gamma}_{l-1}\mathbf{H}_l^{\mathrm{H}}\mathbf{Z}_l^{-\mathrm{H}/2}(\mathbf{I}_N + \mathbf{Z}_l^{-1/2}\mathbf{H}_l\mathbf{\Gamma}_{l-1}\mathbf{H}_l^{\mathrm{H}}\mathbf{Z}_l^{-\mathrm{H}/2})^{-1}\mathbf{Z}_l^{-1/2}\mathbf{H}_l\mathbf{\Gamma}_{l-1}^{\mathrm{H}}$
6: $\quad$ **for** $n = 1 \dots \tau_u$ **do**
7: $\quad\quad \hat{\mathbf{s}}_l^n = \hat{\mathbf{s}}_{l-1}^n + \mathbf{\Gamma}_l\mathbf{H}_l^{\mathrm{H}}\mathbf{Z}_l^{-\mathrm{H}/2}(\hat{\mathbf{y}}_l^n - \mathbf{Z}_l^{-1/2}\mathbf{H}_l\hat{\mathbf{s}}_{l-1}^n)$.
8: $\quad$ **end for**
9: **end for**

anywhere else in the paper. Finally, in the last AP, users' signal estimation error, i.e., $\mathbf{e} = \mathbf{s} - \hat{\mathbf{s}}$, covariance matrix is as follows:

$$\mathbf{\Gamma}_L = \mathbb{E}\{\mathbf{e}\mathbf{e}^{\mathrm{H}}\} = (\mathbf{H}^{\mathrm{H}}\mathbf{Z}^{-1}\mathbf{H} + \frac{1}{p}\mathbf{I}_K)^{-1}. \tag{5}$$

Note that $\hat{\mathbf{s}}_l$ is the LS estimate of users' signal having the received signal vector and CSI from AP 1 to AP l.

we assume perfect CSI in this paper, which is realistic with a large enough transmit power per user during pilot transmission and in an indoor or semi-static environment where the coherence block is large enough to accommodate a unique pilot per user [28]. However, in the simulation section, we provide a subsection on the imperfect CSI.

The authors in [8], [9], and [10] assume an unlimited memory capacity in each AP. However, under the realistic assumption of limited memory capacity in each AP, two questions arise: how to optimally compress and store received signal vector in memory and what is the effect of this compression on the average per-user SE?

B. Limited Memory Capacity at the APs

In a sequential daisy chain fronthaul, each AP stores its raw/pre-processed received signal vector until it receives the estimate of the users' signal vector from the previous AP in the sequence. Then, it refines the estimate of the users' signal vector by co-processing it with their own local received signal vector, as shown in the line 7 of Algorithm 1. To make the problem more tangible, consider a CFmMIMO network with a daisy chain fronthaul topology. There are N_{sc} subcarriers for each OFDM (orthogonal frequency-division multiplexing) symbol. Suppose that $\mathbf{Y}_l^{t_0} \in \mathbb{C}^{N\times N_{sc}}$ is a symbol matrix with each column corresponding to the received signal vector at AP l for a particular subcarrier of symbol t_0. When AP 1 is processing $\mathbf{Y}_1^{t_0}$, the rest of the APs have their corresponding

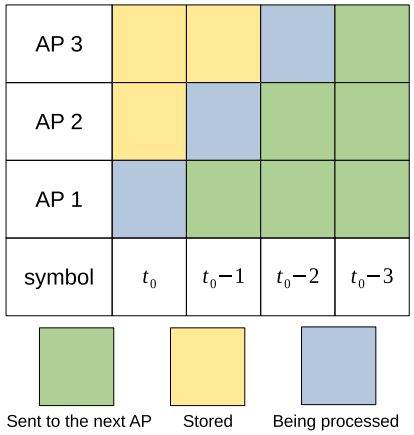


Fig. 2. Sequential processing and storage in a CFmMIMO network with daisy chain fronthaul topology.

matrices, i.e., $\mathbf{Y}_l^{t_0}, \forall l \in \{2, \dots, L\}$ in their memory. Similarly, when AP 2 is processing $\mathbf{Y}_2^{t_0-1}$, the corresponding matrix at AP l, i.e., $\mathbf{Y}_l^{t_0-1}, \forall l \in \{3, \dots, L\}$, is stored at the memory. Accordingly, the number of the symbol matrix stored at AP l is $l-1$, meaning that the number of received signal vectors stored at AP l is $(l-1)N_{sc}$, which increases linearly from one AP to the next by N_{sc}. Fig. 2 demonstrates the sequential processing and storage at the APs. It is worth mentioning that processors are designed to process at least one symbol during a symbol duration to have a stable system. In other words, the rate of the locally received signal vectors entering the memory should be lower or, in the worst case, the same as the rate at which the processor is processing them. Therefore, the number of vectors stored in the AP's memory increases by N_{sc} from one AP to the next.

The memory used to store the received signal vectors is usually a fast on-chip cache memory [29], close to the processing unit.[2] For an example of an AP processor implementation, the reader is referred to [30].

III. Local Received Signal Vector Compression at the APs

In this section, we consider three options for storing the local received signal vectors:

1) Vector-wise compression of the received signal vector, i.e., joint compression of the received signal vector elements.

[2]Note that local CSI is also stored in the memory, next to the received signal vectors. However, the size of the local CSI that needs to be stored in the memory is the same in all APs, and hence, we only focus on storing the received signal vector in this paper.

Authorized licensed use limited to: KU Leuven Libraries. Downloaded on January 29,2025 at 13:57:53 UTC from IEEE Xplore. Restrictions apply.

2) Element-wise compression of the received signal vector.
3) Element-wise compression of PCA pre-processed received signal vector.

To model the compression of the received signal vector at each AP, we use rate-distortion theory [31]. Based on rate-distortion theory, the rate-distortion function of a random variable gives the minimum number of bits needed to compress realizations of the random variable given a distortion constraint. The rate-distortion theory can also be extended to a random vector and covariance distortion constraint [32]. If there are d sources, each generating a unique element of the circularly symmetric complex Gaussian random vector $\mathbf{x}$ with dimension d, the extension to the rate-distortion function of vector $\mathbf{x}$ is given as below:

$$R(\mathbf{Q}) = \min_{f(\hat{\mathbf{x}}|\mathbf{x}),\mathbb{E}\{(\mathbf{x}-\hat{\mathbf{x}})(\mathbf{x}-\hat{\mathbf{x}})^{\mathrm{H}}\}\preceq\mathbf{Q}} I(\mathbf{x};\hat{\mathbf{x}}), \tag{6}$$

where $\hat{\mathbf{x}}$ is the compressed version of $\mathbf{x}$, $f(\hat{\mathbf{x}}|\mathbf{x})$ is the conditional probability distribution function (PDF) of vector $\hat{\mathbf{x}}$, $\mathbf{Q}$ is the compression noise covariance matrix (target distortion) and $R(\mathbf{Q})$ is the rate-distortion function. The mutual information between the vector and its compressed version, i.e., $I(\mathbf{x};\hat{\mathbf{x}})$ can be lower bounded as follows:

$$\begin{aligned} I(\mathbf{x};\hat{\mathbf{x}}) &= \mathcal{H}(\hat{\mathbf{x}}) - \mathcal{H}(\hat{\mathbf{x}}|\mathbf{x}) \\ &\overset{(a)}{=} \mathcal{H}(\hat{\mathbf{x}}) - \mathcal{H}(\hat{\mathbf{x}}-\mathbf{x}|\mathbf{x}) \\ &\overset{(b)}{\geq} \mathcal{H}(\hat{\mathbf{x}}) - \mathcal{H}(\hat{\mathbf{x}}-\mathbf{x}) \\ &\overset{(c)}{\geq} \mathcal{H}(\hat{\mathbf{x}}) - \mathcal{H}(\mathcal{CN}(\mathbf{0},\mathbb{E}\{(\hat{\mathbf{x}}-\mathbf{x})(\hat{\mathbf{x}}-\mathbf{x})^{\mathrm{H}}\}) \\ &= \mathcal{H}(\hat{\mathbf{x}}) - \log_2(\pi e)^d \det(\mathbf{Q}). \end{aligned} \tag{7}$$

Note that in this paper, for the vector-wise compression, we have $d = N$, and for the element-wise compression, we have $d = 1$. In (7), equality $\overset{(a)}{=}$ holds because $-\mathbf{x}$ is a pure translation in $\mathcal{H}(\hat{\mathbf{x}}-\mathbf{x}|\mathbf{x})$ and entropy is translation invariant [33]. Inequality $\overset{(b)}{\geq}$ comes from the fact that conditioning reduces entropy. Inequality $\overset{(c)}{\geq}$ results from the maximum differential entropy, which states that given a variance, the Gaussian distribution will maximize the entropy, which can be generalized to a multivariate Gaussian distribution [31]. To represent the relation between vector $\mathbf{x}$ and its compressed version, i.e., vector $\hat{\mathbf{x}}$, the concept of test channel is used. The test channel that achieves the lower bound $\overset{(c)}{\geq}$ in (7) is as follows [21], [34]:

$$\hat{\mathbf{x}} = \mathbf{x} + \mathbf{q}, \tag{8}$$

where compression noise $\mathbf{q} \sim \mathcal{CN}(\mathbf{0},\mathbf{Q})$ is independent of $\mathbf{x}$. For the rest of the paper, we use the aforementioned test channel in (8) due to mathematical simplicity. However, it is worth mentioning that the optimal test channel to lower bound the mutual information in (7) is as follows:

$$\mathbf{x} = \hat{\mathbf{x}} + \mathbf{q}, \tag{9}$$

where $\mathbf{q}$ and $\hat{\mathbf{x}}$ are independent and $\mathbf{q} \sim \mathcal{CN}(\mathbf{0},\mathbf{Q})$. The test channel in (9) results in a smaller value for $I(\mathbf{x};\hat{\mathbf{x}})$ which means that for a given number of bits, it results in a smaller compression noise power compared to the test channel in (8). However, using the optimal test channel in (9), the solutions to the maximization problems in the following sections are not mathematically straightforward. Hence, the test channel in (8) is used in this paper. In what follows, we show that, besides simplicity, the test channel in (8) is close in performance to the optimal test channel in (9) for the number of compression bits that we consider in this paper. For clarity and illustrative purposes, consider the following example. Suppose that we have a complex circularly symmetric Gaussian random variable $r \sim \mathcal{CN}(0,p_r)$.[3] In this example and without loss of generality, we assume $p_r = 1$. By compressing the random variable using test channel in (8), $I(r;\hat{r})$ is given as:

$$I(r;\hat{r}) = \log_2(\frac{1}{Q}+1). \tag{10}$$

With the optimal test channel in (9), $I(r;\hat{r})$ is given as:

$$I(r;\hat{r}) = \log_2(\frac{1}{Q}), \tag{11}$$

where Q is the compression noise variance. In Fig. 3, we plot and compare the mutual information functions in (10) and (11) (corresponding to the number of compression bits) versus the compression noise power for the two test channels in (8) and (9). It is clear that the lower the number of bits, the higher the compression noise power. The optimal test channel in (9) has a compression noise power that is bounded by the signal power, here p_r, and in case 0 bits are allocated to compress r, the compression noise is, in fact, the signal itself. The test channel in (8) performs worse for a very low number of bits ($<$ 2 bit per signal), where $Q > 0.33$. In our simulations, however, we usually allocate more than 2 bits per signal. Hence, we are usually working in the part of the curve where both test channels have very similar performance, i.e., to the left of the dashed line in Fig. 3.

Therefore, using the test channel in (8) not only gives tractable solutions to the problems maximizing the upper bound on users' sum-SE, but also the simulation results can be generalized to the practical implementations.

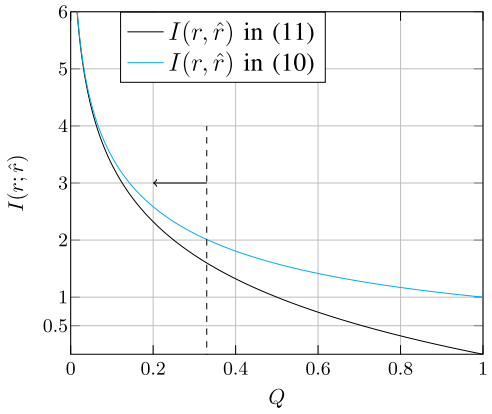


Fig. 3. Mutual information of r and $\hat{r}$ using two test channels in (8) and (9).

[3]To make the comparison between the two test channels in (8) and (9) simple, we considered compressing a complex circularly symmetric Gaussian **scalar** random variable.

Authorized licensed use limited to: KU Leuven Libraries. Downloaded on January 29,2025 at 13:57:53 UTC from IEEE Xplore. Restrictions apply.

A. Option 1: Vector-Wise Compression of the Received Signal Vector

In this option, each AP compresses its received signal vector vector-wisely. Following the rate-distortion argument at the beginning of this section, for AP l, we replace $\mathbf{x}$ with $\mathbf{y}_l$, and then the compressed vector is represented as $\hat{\mathbf{y}}_{vl}$. Using the test channel in (8), the relation between $\mathbf{y}_l$ and its compressed version $\hat{\mathbf{y}}_{vl}$ is given as follows:

$$\hat{\mathbf{y}}_{vl} = \mathbf{y}_l + \mathbf{q}_{vl} = \mathbf{H}_l\mathbf{s} + \mathbf{n}_l + \mathbf{q}_{vl}, \tag{12}$$

where $\mathbf{y}_l$ is defined in (1), $\mathbf{z}_{vl} = \mathbf{n}_l + \mathbf{q}_{vl}$ and $\mathbf{q}_{vl} \sim \mathcal{CN}(\mathbf{0}, \mathbf{Q}_{vl})$ represents the compression noise introduced by the joint compression of the vector elements.

Loosely speaking, the differential entropy of a random variable such as $\hat{\mathbf{y}}_{vl}$ measures the uncertainty of the random variable and the number of bits required to represent it. The mutual information between $\mathbf{y}_l$ and $\hat{\mathbf{y}}_{vl}$ shows how much of the uncertainty of one of them is reduced due to knowledge of the other one. Hence, the mutual information can also measure how many bits are required to represent $\hat{\mathbf{y}}_{vl}$ when $\mathbf{y}_l$ is known [31]. The relation between the number of compression bits per received signal vector, i.e., C_{sc}, and the compression noise covariance matrix $\mathbf{Q}_{vl}$ at AP l, conditioned on the local CSI is as follows [31]:

$$\begin{aligned} C_{sc} &= I(\mathbf{y}_l; \hat{\mathbf{y}}_{vl}|\mathbf{H}_l) \\ &= \mathcal{H}(\hat{\mathbf{y}}_{vl}|\mathbf{H}_l) - \mathcal{H}(\hat{\mathbf{y}}_{vl}|\mathbf{y}_l, \mathbf{H}_l) \\ &= \log_2 \det(\mathbf{Q}_{vl}^{-1}(p\mathbf{H}_l\mathbf{H}_l^{\mathrm{H}} + \sigma^2\mathbf{I}_N) + \mathbf{I}_N), \end{aligned} \tag{13}$$

where the vectors $\{\hat{\mathbf{y}}_{vl}|\mathbf{H}_l\}$ and $\{\hat{\mathbf{y}}_{vl}|\mathbf{y}_l, \mathbf{H}_l\}$ are multi-variate circularly symmetric Gaussian random vectors, as $\mathbf{s}$, $\mathbf{n}_l$ and $\mathbf{q}_{vl}$ are independent multi-variate circularly symmetric Gaussian random vectors. The network-wide compressed received signal vector is as follows:

$$\hat{\mathbf{y}}_v = \mathbf{y} + \mathbf{q}_v = \mathbf{H}\mathbf{s} + \underbrace{\mathbf{n} + \mathbf{q}_v}_{\mathbf{z}_v}, \tag{14}$$

where $\hat{\mathbf{y}}_v = \left[\hat{\mathbf{y}}_{v1}^{\mathrm{T}}, \ldots, \hat{\mathbf{y}}_{vL}^{\mathrm{T}}\right]^{\mathrm{T}}$, $\mathbf{q}_v = \left[\mathbf{q}_{v1}^{\mathrm{T}}, \ldots, \mathbf{q}_{vL}^{\mathrm{T}}\right]^{\mathrm{T}}$, and $\mathbf{z}_v = \left[\mathbf{z}_{v1}^{\mathrm{T}}, \ldots, \mathbf{z}_{vL}^{\mathrm{T}}\right]^{\mathrm{T}}$ is the receiver plus compression noise vector with covariance matrix defined as $\mathbf{Z}_v = \mathbb{E}\{\mathbf{z}_v\mathbf{z}_v^{\mathrm{H}}\} = \mathrm{blkdiag}(\mathbf{Z}_{v1}, \ldots, \mathbf{Z}_{vL}) = \mathrm{blkdiag}(\mathbf{Q}_{v1} + \sigma^2\mathbf{I}_N, \ldots, \mathbf{Q}_{vL} + \sigma^2\mathbf{I}_N)$. Following the discussion in Section II-A, the LS estimate of the users' signal vector in the last AP can be formulated as follows:

$$\hat{\mathbf{s}}_v = \mathbf{C}_v\hat{\mathbf{y}}_v, \tag{15}$$

where the combining matrix $\mathbf{C}_v \in \mathbb{C}^{K\times N}$ given $\mathbf{H}$ is formulated as follows:

$$\mathbf{C}_v = (\mathbf{H}^{\mathrm{H}}\mathbf{Z}_v^{-1}\mathbf{H} + \frac{1}{p}\mathbf{I}_K)^{-1}\mathbf{H}^{\mathrm{H}}\mathbf{Z}_v^{-1}. \tag{16}$$

Having an estimate of the users' signal vector as in (15), the sum-SE of users is formulated as follows:

$$\begin{aligned} R_v &= \frac{\tau_u}{\tau_c}\mathcal{I}(\hat{\mathbf{s}}_v; \mathbf{s}) \\ &= \frac{\tau_u}{\tau_c}\big(\mathcal{H}(\mathbf{s}) - \mathcal{H}(\mathbf{s}|\hat{\mathbf{s}}_v)\big) \\ &\overset{\text{(a)}}{=} \frac{\tau_u}{\tau_c}\log_2\det(p\mathbf{H}\mathbf{H}^{\mathrm{H}}\mathbf{Z}_v^{-1} + \mathbf{I}_{NL}) \\ &\overset{\text{(b)}}{\leq} \frac{\tau_u}{\tau_c}\log_2\prod_{l=1}^{L}\det(p\mathbf{H}_l\mathbf{H}_l^{\mathrm{H}}\mathbf{Z}_{vl}^{-1} + \mathbf{I}_N) \\ &= \frac{\tau_u}{\tau_c}\sum_{l=1}^{L}\log_2\det(p\mathbf{H}_l\mathbf{H}_l^{\mathrm{H}}\mathbf{Z}_{vl}^{-1} + \mathbf{I}_N) \\ &= \frac{\tau_u}{\tau_c}\sum_{l=1}^{L}\log_2\det(p\mathbf{H}_l\mathbf{H}_l^{\mathrm{H}}(\mathbf{Q}_{vl} + \sigma^2\mathbf{I}_N)^{-1} + \mathbf{I}_N), \end{aligned} \tag{17}$$

where R_v can be achieved if sequential interference cancellation (SIC) is used to estimate/detect users' signal in the last AP [35].[4] Furthermore, $\overset{\text{(a)}}{=}$ and $\overset{\text{(b)}}{\leq}$ are proved in Appendix A. The upper bound in equation (17) is for the instantaneous sum-SE in a particular coherence block. It is the summation of L functions, each dependent only on the compression noise covariance matrix and local channel matrix of a single AP. Additionally, each AP compresses its local received signal vector in isolation from other APs. Therefore, the maximization of the upper bound function can be decomposed into L smaller optimization problems to be solved in L APs. Therefore, the maximization sub-problem at AP l is defined as follows:

$$\begin{aligned} \arg\max_{\mathbf{Q}_{vl}^{-1}\succeq\mathbf{0}} \quad & \log_2\det(p\mathbf{H}_l\mathbf{H}_l^{\mathrm{H}}(\mathbf{Q}_{vl} + \sigma^2\mathbf{I}_N)^{-1} + \mathbf{I}_N) \\ & = \log_2\det(p\mathbf{H}_l\mathbf{H}_l^{\mathrm{H}} + \mathbf{Q}_{vl} + \sigma^2\mathbf{I}_N) \\ & \quad - \log_2\det(\mathbf{Q}_{vl} + \sigma^2\mathbf{I}_N) \\ & = \log_2\det(\mathbf{Q}_{vl}^{-1}(p\mathbf{H}_l\mathbf{H}_l^{\mathrm{H}} + \sigma^2\mathbf{I}_N) + \mathbf{I}_N) \\ & \quad - \log_2\det(\sigma^2\mathbf{Q}_{vl}^{-1} + \mathbf{I}_N) \\ \text{s.t.} \quad & C_{sc} = \log_2\det(\mathbf{Q}_{vl}^{-1}(p\mathbf{H}_l\mathbf{H}_l^{\mathrm{H}} + \sigma^2\mathbf{I}_N) + \mathbf{I}_N). \end{aligned} \tag{18}$$

The problem defined in (18) is an NLP (non-linear programming) problem. The constraint does not define a convex set, so the problem is not convex. However, the closed-form globally optimal matrix is derived in (21).

To present the optimal solution, we need the singular value decomposition (SVD) of matrix $\mathbf{H}_l$, which is formulated as follows:

$$\mathbf{H}_l = \mathbf{U}_l\mathbf{\Sigma}_l\mathbf{V}_l^{\mathrm{H}}. \tag{19}$$

Accordingly, the eigenvalue decomposition of matrix $\mathbf{H}_l\mathbf{H}_l^{\mathrm{H}}$ is as follows:

$$\mathbf{H}_l\mathbf{H}_l^{\mathrm{H}} = \mathbf{U}_l\mathbf{\Sigma}_l\mathbf{\Sigma}_l^{\mathrm{H}}\mathbf{U}_l^{\mathrm{H}}. \tag{20}$$

The columns of matrices $\mathbf{U}_l \in \mathbb{C}^{N\times N}$ and $\mathbf{V}_l \in \mathbb{C}^{K\times K}$ are the left and right singular vectors of $\mathbf{H}_l$, respectively, and so the columns of $\mathbf{U}_l$ are also the eigenvectors of matrix $\mathbf{H}_l\mathbf{H}_l^{\mathrm{H}}$. Furthermore, $\mathbf{\Sigma}_l \in \mathbb{C}^{N\times K}$ is a rectangular matrix with the sorted singular values of the $\mathbf{H}_l$ on its diagonal elements. The optimal matrix $\mathbf{Q}_{vl}^{-1}$ is given as:

$$(\mathbf{Q}_{vl}^{-1})^o = \mathbf{U}_l(\mathbf{\Sigma}_{vlq}^{-1})^o\mathbf{U}_l^{\mathrm{H}}, \tag{21}$$

[4] Sum-SE of the users when the users are detected jointly is more than the case in which the users are detected individually (i.e., $\sum_k \mathcal{I}(\hat{\mathbf{s}}_k; \mathbf{s}_k) \leq \mathcal{I}(\hat{\mathbf{s}}; \mathbf{s})$ [31]).

Authorized licensed use limited to: KU Leuven Libraries. Downloaded on January 29,2025 at 13:57:53 UTC from IEEE Xplore. Restrictions apply.

where the i^{th} diagonal element of $(\mathbf{\Sigma}_{vlq}^{-1})^{o}$ is calculated as:

$$\lambda_{vlqi}^{o} = \max(0, \frac{1}{\mu_{vl}^{o}}(\frac{1}{\sigma^2} - \frac{1}{p\lambda_{li}^2 + \sigma^2}) - \frac{1}{\sigma^2}), \forall i. \quad (22)$$

where λ_{li} is the i^{th} singular value of $\mathbf{H}_l$. The solution in (22) is reverse water filling on the eigenvalues of received signal vector covariance matrix [31]. The proof is elaborated on in Appendix B.

Note that the Lagrange multiplier μ_{vl}^{o} is selected to meet the equality constraint in (18).

B. Option 2: Element-Wise Compression of the Received Signal Vector

Each element of the received signal vector is compressed individually in this option. The number of bits allocated to compression of the i^{th} element is denoted by b_i and $C_{sc} = \sum_{i=1}^{N} b_i$. The compressed i^{th} element of the received signal vector at AP l is as follows:

$$\hat{y}_{eli} = y_{li} + q_{eli} = \mathbf{H}_{l[i,:]}\mathbf{s} + n_{li} + q_{eli}, \quad (23)$$

where n_{li} is the i^{th} element of noise vector $\mathbf{n}_l$, $z_{eli} = n_{li} + q_{eli}$ with $q_{eli} \sim \mathcal{CN}(0, \sigma_{eli}^2), \forall i \in \{1, \ldots, N\}$ and $\mathbf{q}_{el} = \left[q_{el1}, \cdots, q_{elN}\right]^{\mathrm{T}}$ is the compression noise vector with covariance matrix $\mathbf{Q}_{el} = \mathbb{E}\{\mathbf{q}_{el}\mathbf{q}_{el}^{\mathrm{H}}\}$. Vector $\hat{\mathbf{y}}_{el} = \left[\hat{y}_{el1}, \ldots, \hat{y}_{elN}\right]^{\mathrm{T}}$ is the compressed received signal vector at AP l. The receiver plus compression noise vector is $\mathbf{z}_{el} = \left[z_{el1}, \cdots, z_{elN}\right]^{\mathrm{T}}$.

The relation between b_i and the compression noise of the i^{th} element is as follows [31]:

$$\begin{aligned} b_i &= \mathcal{I}(y_{li}; \hat{y}_{eli}|\mathbf{H}_{l[i,:]}) = \mathcal{H}(\hat{y}_{eli}|\mathbf{H}_{l[i,:]}) - \mathcal{H}(\hat{y}_{eli}|y_{li}, \mathbf{H}_{l[i,:]}) \\ &= \log_2(1 + \frac{p\|\mathbf{H}_{l[i,:]}\|^2 + \sigma^2}{\sigma_{eli}^2}), \end{aligned} \quad (24)$$

where subscript $[i,:]$ represent the i^{th} row of $\mathbf{H}_l$. The diagonal elements of the covariance matrix $\mathbf{Q}_{el}$ are known and equal to $\sigma_{eli}^2, \forall i \in \{1, \ldots, N\}$. The off-diagonal elements of matrix $\mathbf{Q}_{el}$ are unknown.

We define diagonal matrix $\mathbf{P}_l$ with the variance of the elements of the received signal vector $\{\mathbf{y}_l|\mathbf{H}_l\}$ as its diagonal elements,

$$\mathbf{P}_l = \text{diag}\left(p\|\mathbf{H}_{l[1,:]}\|^2 + \sigma^2, \cdots, p\|\mathbf{H}_{l[N,:]}\|^2 + \sigma^2\right), \quad (25)$$

and the diagonal matrix $\mathbf{Q}_{el}^{d}$ with variance of the elements of the compression noise vector $\mathbf{q}_{el}$ as its diagonal elements, as follows:

$$\mathbf{Q}_{el}^{d} = \text{diag}\left(\sigma_{el1}^2, \ldots, \sigma_{elN}^2\right). \quad (26)$$

We can relate C_{sc} to the variance of the compression noise vector elements as follows:

$$\begin{aligned} C_{sc} &= \sum_{i=1}^{N} b_i = \log_2 \prod_{i=1}^{N}(1 + \frac{p\|\mathbf{H}_{l[i,:]}\|^2 + \sigma^2}{\sigma_{eli}^2}) \\ &= \log_2 \det(\mathbf{I}_N + \mathbf{P}_l{\mathbf{Q}_{el}^{d}}^{-1}). \end{aligned} \quad (27)$$

The values for $b_i, \forall i$ (and consequently σ_{eli}^2) can be selected heuristically. One simple approach is compressing the vector elements with the same number of bits. A better approach is to select the number of bits based on the variance of each element of the received signal vector.

The APs refine the users' signal vector based on the element-wise compressed received signal vector. The network-wide compressed received signal vector, in this case, is as follows:

$$\hat{\mathbf{y}}_e = \mathbf{y} + \mathbf{q}_e = \mathbf{H}\mathbf{s} + \underbrace{\mathbf{n} + \mathbf{q}_e}_{\mathbf{z}_e}, \quad (28)$$

where $\hat{\mathbf{y}}_e = \left[\hat{\mathbf{y}}_{e1}^{\mathrm{T}}, \ldots, \hat{\mathbf{y}}_{eL}^{\mathrm{T}}\right]^{\mathrm{T}}$, $\mathbf{q}_e = \left[\mathbf{q}_{e1}^{\mathrm{T}}, \ldots, \mathbf{q}_{eL}^{\mathrm{T}}\right]^{\mathrm{T}}$ and $\mathbf{z}_e = \left[\mathbf{z}_{e1}^{\mathrm{T}}, \ldots, \mathbf{z}_{eL}^{\mathrm{T}}\right]^{\mathrm{T}}$ with covariance matrix $\mathbf{Z}_e = \mathbb{E}\{\mathbf{z}_e\mathbf{z}_e^{\mathrm{H}}\} = \text{blkdiag}(\mathbf{Z}_{e1}, \ldots, \mathbf{Z}_{eL}) = \text{blkdiag}(\mathbf{Q}_{e1} + \sigma^2\mathbf{I}_N, \ldots, \mathbf{Q}_{eL} + \sigma^2\mathbf{I}_N)$. In element-wise compression, the correlation between compression noise elements in one AP, i.e., the off-diagonal elements of $\mathbf{Q}_{el}, \forall l$, are unknown. Therefore, while computing the combining matrix to estimate users' signal vector, $\mathbf{Q}_{el}, \forall l$ is assumed to be a diagonal matrix, which adversely affects the estimation quality. Following the discussion in Section II-A, the estimation of the users' signal vector results in equations (29) and (30), as follows:

$$\hat{\mathbf{s}}_e = \mathbf{C}_e\hat{\mathbf{y}}_e, \quad (29)$$

where the combining matrix $\mathbf{C}_e$ is formulated as follows:

$$\mathbf{C}_e = (\mathbf{H}^{\mathrm{H}}(\mathbf{Z}_e^{d})^{-1}\mathbf{H} + \frac{1}{p}\mathbf{I}_K)^{-1}\mathbf{H}^{\mathrm{H}}(\mathbf{Z}_e^{d})^{-1}, \quad (30)$$

where $\mathbf{Z}_e^{d} = \text{blkdiag}(\mathbf{Z}_{e1}^{d}, \ldots, \mathbf{Z}_{eL}^{d}) = \text{blkdiag}(\mathbf{Q}_{e1}^{d} + \sigma^2\mathbf{I}_N, \ldots, \mathbf{Q}_{eL}^{d} + \sigma^2\mathbf{I}_N)$ and $\mathbf{Q}_{el}^{d}$ is the diagonal matrix with the same diagonal elements as $\mathbf{Q}_{el}$. The solution to the optimization problem in option 2 follows the same steps in section III-A and Appendix B. However, deriving an upper bound on users' sum-SE similar to (17) is not straightforward due to the lack of knowledge of the off-diagonal elements of $\mathbf{Z}_e$ and thus can be simplified as follows:

$$\begin{aligned} R_e &= \frac{\tau_u}{\tau_c}\log_2\det(p\mathbf{H}\mathbf{H}^{\mathrm{H}}(\mathbf{Z}_e^{d})^{-1} + \mathbf{I}_{NL}) \\ &\overset{(a)}{\leq} \frac{\tau_u}{\tau_c}\sum_{l=1}^{L}\log_2\det(p\mathbf{H}_l\mathbf{H}_l^{\mathrm{H}}(\mathbf{Z}_{el}^{d})^{-1} + \mathbf{I}_N) \\ &\overset{(b)}{\leq} \frac{\tau_u}{\tau_c}\sum_{l=1}^{L}\log_2\det(p\text{diag}(\mathbf{H}_l\mathbf{H}_l^{\mathrm{H}})(\mathbf{Z}_{el}^{d})^{-1} + \mathbf{I}_N) \\ &= \frac{\tau_u}{\tau_c}\sum_{l=1}^{L}\log_2\det(p\mathbf{W}_l(\mathbf{Q}_{el}^{d} + \sigma^2\mathbf{I}_N)^{-1} + \mathbf{I}_N), \end{aligned} \quad (31)$$

where $\mathbf{W}_l$ is defined as $\mathbf{W}_l = \text{diag}(\|\mathbf{H}_{l[1,:]}\|^2, \ldots, \|\mathbf{H}_{l[N,:]}\|^2)$. In (31), the upper bounds $\overset{(a)}{\leq}$ and $\overset{(b)}{\leq}$ are proved similar to $\overset{(b)}{\leq}$ in (17).

Similar to Section III-A, the sub-problem at AP l to find the diagonal element of $\mathbf{Q}_{el}^{d}$ and, subsequently, the number of bits to compress each of the elements of the received signal vector $\mathbf{y}_l$ is formulated as follows:

$$\arg \max_{\mathbf{Q}_{el}^{d}{}^{-1} \succeq \mathbf{0}} \quad \log_2\det(p\mathbf{W}_l(\mathbf{Q}_{el}^{d} + \sigma^2\mathbf{I}_N)^{-1} + \mathbf{I}_N)$$

Authorized licensed use limited to: KU Leuven Libraries. Downloaded on January 29,2025 at 13:57:53 UTC from IEEE Xplore. Restrictions apply.

$$
\begin{aligned}
&= \log_2 \det(p\mathbf{W}_l + \mathbf{Q}_{el}^d + \sigma^2\mathbf{I}_N) \\
&- \log_2 \det(\mathbf{Q}_{el}^d + \sigma^2\mathbf{I}_N) \\
&- = \log_2 \det(\mathbf{P}_l {\mathbf{Q}_{el}^d}^{-1} + \mathbf{I}_N) \\
&- \log_2 \det(\sigma^2 {\mathbf{Q}_{el}^d}^{-1} + \mathbf{I}_N) \\
\text{s.t.} \quad & C_{sc} = \log_2 \det(\mathbf{P}_l {\mathbf{Q}_{el}^d}^{-1} + \mathbf{I}_N), \qquad (32)
\end{aligned}
$$

With $\mathbf{P}_l$ defined in (25). Note that $\mathbf{P}_l = p\mathbf{W}_l + \sigma^2\mathbf{I}_N$. Following the same steps as in Appendix B, the i^{th} diagonal element of matrix $({\mathbf{Q}_{el}^d}^{-1})^o$, shown as λ^o_{elqi}, is given as follows:

$$
\lambda^o_{elqi} = \max(0, \frac{1}{\mu^o_{el}}(\frac{1}{\sigma^2} - \frac{1}{\mathbf{P}_{l[i,i]}}) - \frac{1}{\sigma^2}), \forall i, \qquad (33)
$$

where $\mathbf{P}_{l[i,i]}$ denotes the i^{th} diagonal element of $\mathbf{P}_l$. Finally, $({\mathbf{Q}_{el}^d}^{-1})^o = \text{diag}(\lambda^o_{elq1}, \ldots, \lambda^o_{elqN})$. The parameter μ^o_{el} in (33) is also calculated similarly to μ^o_{vl} in Section III-A.

C. Option 3: Element-Wise Compression of the PCA-Pre-Processed Received Signal Vector

Despite being efficient in bit usage, vector compression can be costly in practice. For example, using well-known quantization algorithms such as the generalized Lloyd algorithm to quantize a given set of N-dimensional vectors with n_q bits have a complexity order of $\mathcal{O}(\text{number of vectors} \times N \times 2^{n_q} \times I)$ if the algorithm stops after I iterations. On the other hand, element-wise quantization (sharing the total number of bits among the dimensions and then quantizing each dimension individually) is of lower complexity, i.e., $\mathcal{O}(\text{number of vectors} \times 2^{n_q\delta_i} \times I)$ for quantizing dimension i and $\sum_{j=1}^N \delta_j = 1$, at the cost of inefficient bit usage. To bridge this gap, it is proposed to de-correlate the vectors' dimensions before element-wise quantization to use the total bits efficiently. With the above introduction in mind, suppose each AP uses PCA to map its local received signal vector into another subspace. At AP l, the covariance matrix of the received signal vector, using the SVD of $\mathbf{H}_l$ in (19), can be formulated as follows:

$$
\begin{aligned}
\mathbf{R}_{yl} &= \mathbb{E}\{(\mathbf{y}_l - \mathbb{E}\{\mathbf{y}_l\})(\mathbf{y}_l - \mathbb{E}\{\mathbf{y}_l\})^{\text{H}} | \mathbf{H}_l\} \\
&\overset{(a)}{=} \mathbb{E}\{\mathbf{y}_l\mathbf{y}_l^{\text{H}} | \mathbf{H}_l\} = \mathbf{U}_l \underbrace{(p\mathbf{\Sigma}_l\mathbf{\Sigma}_l^{\text{H}} + \sigma^2\mathbf{I}_N)}_{\tilde{\mathbf{\Sigma}}_l} \mathbf{U}_l^{\text{H}}, \qquad (34)
\end{aligned}
$$

where $\overset{(a)}{=}$ is due to the fact that $\mathbb{E}\{\mathbf{y}_l|\mathbf{H}_l\} = \mathbf{0}$ (as $\mathbb{E}\{\mathbf{s}\} = 0$ and $\mathbb{E}\{\mathbf{n}_l\} = 0$). The mapped received signal vector is as follows:

$$
\tilde{\mathbf{y}}_l = \mathbf{A}_l^{\text{H}}\mathbf{y}_l = \underbrace{\mathbf{A}_l^{\text{H}}\mathbf{H}_l}_{\tilde{\mathbf{H}}_l}\mathbf{s} + \underbrace{\mathbf{A}_l^{\text{H}}\mathbf{n}_l}_{\tilde{\mathbf{n}}_l}, \qquad (35)
$$

where $\mathbf{A}_l = \mathbf{U}_{l[:,1:x]}$, in which x can be in the range $[1 : min(N, K)]$. We select $x = min(N, K)$. Note that $\tilde{\mathbf{H}}_l$ and $\tilde{\mathbf{n}}_l$ are the local effective channel and receiver noise at AP l, respectively. After the mapping, the APs must store the pre-processed received signal vector. The vector $\{\tilde{\mathbf{y}}_l|\tilde{\mathbf{H}}_l\}$ is a circularly symmetric Gaussian random vector with zero mean and uncorrelated elements (with diagonal covariance matrix $\mathbb{E}\{\tilde{\mathbf{y}}_l\tilde{\mathbf{y}}_l^{\text{H}}|\tilde{\mathbf{H}}_l\} = \tilde{\mathbf{\Sigma}}_l$), hence the elements are also mutually independent as any two jointly Gaussian and uncorrelated random variables are independent as well. The compressed pre-processed received signal vector at AP l is as follows:

$$
\hat{\tilde{\mathbf{y}}}_{el} = \tilde{\mathbf{y}}_l + \tilde{\mathbf{q}}_{el} = \tilde{\mathbf{H}}_l\mathbf{s} + \tilde{\mathbf{n}}_l + \tilde{\mathbf{q}}_{el}, \qquad (36)
$$

where $\tilde{\mathbf{z}}_{el} = \tilde{\mathbf{n}}_l + \tilde{\mathbf{q}}_{el}$ and $\tilde{\mathbf{q}}_{el} \sim \mathcal{CN}(0, \tilde{\mathbf{Q}}_{el})$. Note that $\tilde{\mathbf{Q}}_{el}$ is assumed to be diagonal as the elements of the $\{\tilde{\mathbf{y}}_l|\tilde{\mathbf{H}}_l\}$ are independent. Hence, they can be optimally compressed individually, conditioned on the optimal allocation of a total number of bits. Matrix $\tilde{\mathbf{Q}}_{el}$ can be related to C_{sc} as follows:

$$
\begin{aligned}
C_{sc} &= I(\tilde{\mathbf{y}}_l; \hat{\tilde{\mathbf{y}}}_{el}|\tilde{\mathbf{H}}_l) \\
&= \mathcal{H}_l(\hat{\tilde{\mathbf{y}}}_{el}|\tilde{\mathbf{H}}_l) - \mathcal{H}_l(\hat{\tilde{\mathbf{y}}}_{el}|\tilde{\mathbf{y}}_l, \tilde{\mathbf{H}}_l) \\
&= \log_2 \det(\tilde{\mathbf{Q}}_{el}^{-1}(p\tilde{\mathbf{H}}_l\tilde{\mathbf{H}}_l^{\text{H}} + \sigma^2\mathbf{I}_x) + \mathbf{I}_x) \\
&= \sum_{i=1}^{x} \log_2(\tilde{\lambda}_{elqi}(p\lambda_{il}^2 + \sigma^2) + 1), \qquad (37)
\end{aligned}
$$

where $\tilde{\lambda}_{elqi}$ is the i^{th} diagonal element of $\tilde{\mathbf{Q}}_{el}^{-1}$. After pre-processing and compressing the local received signal vector, the network-wide compressed received signal vector is as follows:

$$
\hat{\tilde{\mathbf{y}}}_e = \tilde{\mathbf{y}} + \tilde{\mathbf{q}}_e = \tilde{\mathbf{H}}\mathbf{s} + \underbrace{\tilde{\mathbf{n}} + \tilde{\mathbf{q}}_e}_{\tilde{\mathbf{z}}_e}, \qquad (38)
$$

where $\hat{\tilde{\mathbf{y}}}_e = \left[\hat{\tilde{\mathbf{y}}}_{e1}^{\text{T}}, \ldots, \hat{\tilde{\mathbf{y}}}_{eL}^{\text{T}}\right]^{\text{T}}$, $\tilde{\mathbf{y}} = \left[\tilde{\mathbf{y}}_1^{\text{T}}, \ldots, \tilde{\mathbf{y}}_L^{\text{T}}\right]^{\text{T}}$, $\tilde{\mathbf{H}} = \left[\tilde{\mathbf{H}}_1^{\text{T}}, \ldots, \tilde{\mathbf{H}}_L^{\text{T}}\right]^{\text{T}}$. Furthermore, $\tilde{\mathbf{q}}_e = \left[\tilde{\mathbf{q}}_{e1}^{\text{T}}, \ldots, \tilde{\mathbf{q}}_{eL}^{\text{T}}\right]^{\text{T}}$ and $\tilde{\mathbf{z}}_e = \left[\tilde{\mathbf{z}}_{e1}^{\text{T}}, \ldots, \tilde{\mathbf{z}}_{eL}^{\text{T}}\right]^{\text{T}}$ with covariance matrix $\tilde{\mathbf{Z}}_e = \mathbb{E}\{\tilde{\mathbf{z}}_e\tilde{\mathbf{z}}_e^{\text{H}}\} = \text{blkdiag}(\tilde{\mathbf{Z}}_{e1}, \ldots, \tilde{\mathbf{Z}}_{eL}) = \text{blkdiag}(\tilde{\mathbf{Q}}_{e1} + \sigma^2\mathbf{I}_x, \ldots, \tilde{\mathbf{Q}}_{eL} + \sigma^2\mathbf{I}_x)$. Finally, after RLS processing of the compressed PCA pre-processed signal vectors at the APs sequentially, the estimate of users' signal vector in the last AP is as follows:

$$
\hat{\tilde{\mathbf{s}}} = \tilde{\mathbf{C}}_e\hat{\tilde{\mathbf{y}}}_e, \qquad (39)
$$

where the combining matrix given $\tilde{\mathbf{H}}$ is formulated as follows:

$$
\tilde{\mathbf{C}}_e = (\tilde{\mathbf{H}}^{\text{H}}\tilde{\mathbf{Z}}_e^{-1}\tilde{\mathbf{H}} + \frac{1}{p}\mathbf{I}_{xL})^{-1}\tilde{\mathbf{H}}^{\text{H}}\tilde{\mathbf{Z}}_e^{-1}. \qquad (40)
$$

By defining the signal estimation error as $\tilde{\mathbf{e}} = \mathbf{s} - \hat{\tilde{\mathbf{s}}}$, and with the same reasoning as subsection III-A the users' sum-SE is given as:

$$
\begin{aligned}
R_e^{PCA} &= \frac{\tau_u}{\tau_c}\mathcal{I}(\mathbf{s}; \hat{\tilde{\mathbf{s}}}) \\
&= \frac{\tau_u}{\tau_c}\big(\mathcal{H}(\mathbf{s}) - \mathcal{H}(\mathbf{s}|\hat{\tilde{\mathbf{s}}})\big) \\
&\overset{(a)}{=} \frac{\tau_u}{\tau_c}\log_2 \det(p\tilde{\mathbf{H}}\tilde{\mathbf{H}}^{\text{H}}\tilde{\mathbf{Z}}_e^{-1} + \mathbf{I}_{xL}) \\
&\overset{(b)}{\leq} \frac{\tau_u}{\tau_c}\sum_{l=1}^{L}\log_2 \det(p\tilde{\mathbf{H}}_l\tilde{\mathbf{H}}_l^{\text{H}}\tilde{\mathbf{Z}}_{el}^{-1} + \mathbf{I}_x), \qquad (41)
\end{aligned}
$$

where $\overset{(a)}{=}$ and $\overset{(b)}{\leq}$ are proved similarly to $\overset{(a)}{=}$ and $\overset{(b)}{\leq}$ in (17), respectively. The upper bound function defined in (41) is a summation of L functions, each dependent on a unique matrix

Authorized licensed use limited to: KU Leuven Libraries. Downloaded on January 29,2025 at 13:57:53 UTC from IEEE Xplore. Restrictions apply.

$\tilde{\mathbf{Z}}_{el}$ and effective channel $\tilde{\mathbf{H}}_l$. The maximization sub-problem at AP l is formulated as:

$$\begin{aligned}\arg\max_{\tilde{\mathbf{Q}}_{el}^{-1}\succeq\mathbf{0}} \quad & \log_2\det(p\tilde{\mathbf{H}}_l\tilde{\mathbf{H}}_l^{\mathrm{H}}\tilde{\mathbf{Z}}_{el}^{-1}+\mathbf{I}_x)\\ &=\log_2\det(\tilde{\mathbf{Q}}_{el}^{-1}(p\tilde{\mathbf{H}}_l\tilde{\mathbf{H}}_l^{\mathrm{H}}+\sigma^2\mathbf{I}_x)+\mathbf{I}_x)\\ &-\log_2\det(\sigma^2\tilde{\mathbf{Q}}_{el}^{-1}+\mathbf{I}_x)\\ &=\sum_{i=1}^{x}\log_2(\tilde{\lambda}_{elqi}(p\lambda_{il}^2+\sigma^2)+1)\\ &-\sum_{i=1}^{x}\log_2(\tilde{\lambda}_{elqi}\sigma^2+1)\\ \text{s.t.}\quad & C_{sc}=\sum_{i=1}^{x}\log_2(\tilde{\lambda}_{elqi}(p\lambda_{il}^2+\sigma^2)+1). \end{aligned}\tag{42}$$

Based on the discussion at the beginning of this section, we know that $\tilde{\mathbf{Q}}_{el},\forall l$ is a diagonal matrix. The optimization problem follows the similar steps as Appendix B. Matrix $(\tilde{\mathbf{Q}}_{el}^{-1})^o$ with the i^{th} diagonal element defined in (43) maximizes the objective function in (42).

$$\tilde{\lambda}_{elqi}^o=\max(0,\frac{1}{\tilde{\mu}_{el}^o}(\frac{1}{\sigma^2}-\frac{1}{p\lambda_{li}^2+\sigma^2})-\frac{1}{\sigma^2}),\forall i.\tag{43}$$

Finally, matrix $(\tilde{\mathbf{Q}}_{el}^{-1})^o=\mathrm{diag}(\tilde{\lambda}_{elq1}^o,\ldots,\tilde{\lambda}_{elqx}^o)$. The parameter $\tilde{\mu}_{el}^o$ in (43) is also calculated similarly to μ_{vl}^o in Section III-A.

IV. Memory Capacity Allocation Models and Their Impact on the Rate on the Fronthaul Links

The simulation section considers two specific fronthaul topologies, a daisy chain topology, and a multi-branch tree topology, as shown in Fig. 1. The following discussion is based on the daisy chain topology and can be easily extended to the tree topology.

A. Memory Capacity Allocation Models

Regarding the memory capacity at the APs, we consider two reference scenarios. In the following, we assume a fixed number of antennas are distributed among the APs.

- **Fixed per AP (FAP) memory capacity**: Regardless of the number of antennas per AP, APs have the same processing chips and consequently the same processing power and fixed memory capacity C_{AP}. Therefore, the total memory capacity depends on the number of APs, i.e., $C_T=LC_{AP}$.
- **Fixed Total (FT) memory capacity:** There is a fixed total memory capacity C_T that is divided between APs. This total memory capacity can be allocated to APs in two ways:
  - **Equal Allocation (EA):** In this memory model, the total memory capacity is split equally among APs, so $C_{AP}=\frac{C_T}{L}$.
  - **Linear Allocation (LA):** The memory capacity is distributed according to the number of signal vectors stored in the APs. Therefore, the first APs in the sequence receive less memory capacity, and as we move along the sequence, the APs receive a larger share of the total memory capacity. Based on the discussion in Section II, the total number of the received signal vectors, i.e., N_v, in the whole network, is as follows:

$$\begin{aligned}N_{rsv}&=N_{sc}+2N_{sc}+\ldots+(L-1)N_{sc}\\&=\frac{L(L-1)N_{sc}}{2}.\end{aligned}\tag{44}$$

In the case of FAP and FT-EA, the number of bits allocated to each received signal vector in AP l is as follows:

$$C_{sc}=\frac{C_{AP}}{(l-1)N_{sc}}.\tag{45}$$

Furthermore, the number of bits allocated to each received signal vector in FT-LA is as follows:

$$C_{sc}=\frac{C_T}{N_{rsv}}=\frac{2C_T}{L(L-1)N_{sc}}.\tag{46}$$

In reality, local CSI should also be stored in the local memory, which makes efficient memory usage more critical. However, as the amount of data related to CSI is similar in each AP, we ignore the low precision storage of the local CSI.

B. The Rate of the Fronthaul Links

Compression in a CFmMIMO network can also happen in the fronthaul link connecting two APs, which occurs when the fronthaul capacity is the limiting factor. In what follows, we compute the data rate of the links connecting two subsequent APs using the memory model introduced earlier in this section. Each AP needs to send K scalars to the next AP in RLS, as explained in Algorithm 1. To compute the estimated signal for a reference user, we need to multiply a combining vector with ρ bits per element with a received signal vector with γ_i bits for element i. To measure the rate of the fronthaul links, it is worth knowing that:

- The number of bits to represent the multiplication product of an $\rho-$bit multiplicand and an γ_i-bit multiplier (and ignoring the sign bit) is at most $\rho+\gamma_i$ bits [36].
- Furthermore, the number of bits to represent the summation of an $(\rho+\gamma_i)-$bit summand and an $(\rho+\gamma_j)-$bit summand can not exceed $\max(\rho+\gamma_i,\rho+\gamma_j)+1$ [36].[5]

Hence, the number of bits to represent each user's signal estimate, which is the result of N multiplications and then N summations (inner product of the combining vector with the compressed received signal vector), is at most as follows[6]:

$$\alpha=\max_i(\rho+\gamma_i)+1,\tag{47}$$

Scaling with the number of subcarriers and users, the rate of the fronthaul link connecting AP l to AP $l+1$ is as follows:

$$\begin{aligned}C_{fl}=\frac{KN_{sc}\alpha}{T_s}&=\frac{KN_{sc}(\max_i(\rho+\gamma_i)+1)}{T_s}\\&\overset{(a)}{\leq}\frac{KN_{sc}(\rho+C_{sc})}{T_s},\end{aligned}\tag{48}$$

[5]We consider only the fixed point arithmetic in this section, due to its simplicity and practicality.

[6]The upper bound in (47) is for the case of adding the numbers sequentially from the smallest number to the largest.

Authorized licensed use limited to: KU Leuven Libraries. Downloaded on January 29,2025 at 13:57:53 UTC from IEEE Xplore. Restrictions apply.

where T_s is OFDM symbol duration in second. Note that C_{sc} in FAP and FT-EA depends on the AP index, as seen in (45). Therefore, in these scenarios, the upper bound on the rate of the fronthaul link in (48) depends on the AP index, which is higher for the APs at the beginning of the sequence. However, the upper bound on the rate of the fronthaul link is the same between any two APs and independent of the AP index when FT-LA is used.

It is worth mentioning that exchanging the CSI-related matrix, e.g., $\mathbf{\Gamma}_l$ in Algorithm 1 from AP l to the next AP, also contributes to the rate of the fronthaul link. However, this exchange only happens once in one coherence block and is assumed to be scenario and AP index independent. Therefore, it is neglected in the (48). Note that in case a multi-branch tree topology is used for the fronthaul, then the C_{sc} depends on the level that a particular AP such as AP l resides, e.g., in (45) and also on the depth of the tree (total number of levels), e.g., in (46).

In conclusion, the rate of each fronthaul link scales with C_{sc}. Therefore, optimizing memory size to minimize the memory cost in the APs positively impacts the rate of the fronthaul links.

V. Simulation Results

This section presents simulation results, which give insight into how the limited memory capacity in each AP affects the average per-user SE. The average per-user SE i.e., R^u_{sub} is calculated as follows:

$$R^u_{sub} = \mathbb{E}_{\mathbf{H}}\{R_{sub}\}/K, \tag{49}$$

where $sub \in \{v, e, ePCA\}$ indicates one of the three local compression options and $R_{sub} \in \{R_v, R_e, R_{ePCA}\}$ is the sum-SE of the users corresponding to one of the three options which are defined in sectionIII. The expectation in 49 is on the different realization of channel matrix $\mathbf{H}$. The simulation area is square with a perimeter of $D = 500m$ [8]. The total number of antennas is $M = 128$ (unless otherwise stated) which are distributed in $L = \{2, 4, 8, 16, 32, 64, 128\}$ APs. The APs are distributed around the area in a daisy chain or a multi-branch tree with at most two branches per node. The path loss model of an urban microcell with 2GHz carrier frequency is considered [8], [37].

$$\beta_{kl} = -30.5 - 36.7 \log_{10}(\frac{d_{kl}}{1m}), \tag{50}$$

where d_{kl} is the distance (in meters) between user k and AP l. The communication bandwidth is $B = 100$MHz and an FFT size of $N_{sc} = 4096$ [38]. Noise variance is $\sigma^2 = -85$dBm. In addition, the transmitted power of the users is $p = 10$mWatt. Table II summarizes a list of abbreviations introduced before and used in this section.

A. Optimal Number of APs in the Presence of Limited Memory

In Fig. 4, the average per-user SE[7] versus the number of APs is plotted. Two scenarios of memory capacity allocation are considered. Furthermore, the sub-figures are associated with different numbers of users. It is worth mentioning that FAP and FT-EA with the selected C_{AP} and C_T represent the same scenario in the case of the full distribution of antennas. We also considered infinite memory size (no compression) as a benchmark. It is observed that:

- Having memory constraints always limits the performance of distributed antenna systems with sequential processing. While the average per-user SE improves when distributing the fixed number of antennas over multiple APs with infinite memory, this is no longer true when there are memory constraints.
- When limiting the memory capacity so that all APs have the same memory constraint, the performance is dominated by the memory requirements of the last AP. In all sub-figures in Fig. 4, the FAP model with $C_{AP} = 64KB$ performs the same as the FT-LA with $C_T = 8MB$ when $L = 128$. However, when the number of AP is halved, i.e., $L = 64$, the increase in the average per-user SE is about 20% in FT-EA and 6% in FAP, e.g., in the case of

TABLE II
Abbreviations

Parameter Description	Abbreviation
Vector-wise Compression	VC
Element-wise compression	EC
Fixed memory capacity per AP	FAP
Fixed total memory capacity	FT
Equal allocation	EA
Linear allocation	LA

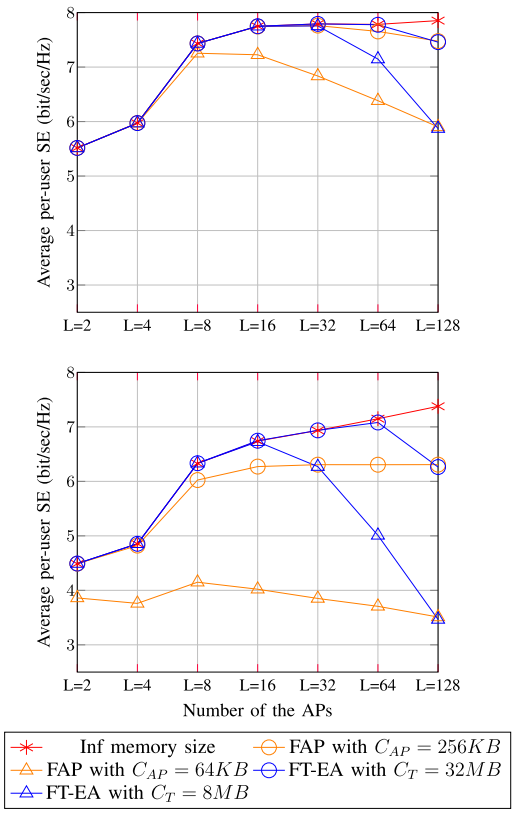


Fig. 4. Impact of limited memory capacity on average per-user SE in a daisy chain fronthaul with two different numbers of users: (Top) $K = 4$, (Bottom) $K = 64$. VC is used for compression.

[7]Note that for simulation results, as 1) we don't consider any specific values for τ_u and τ_c and 2) the factor $\frac{\tau_u}{\tau_c}$ appears sum-SE of all the compression options, we neglect it in plotting the figures. Also, the average per-user SE is the average users' sum-SE divided by the number of users.

Authorized licensed use limited to: KU Leuven Libraries. Downloaded on January 29,2025 at 13:57:53 UTC from IEEE Xplore. Restrictions apply.

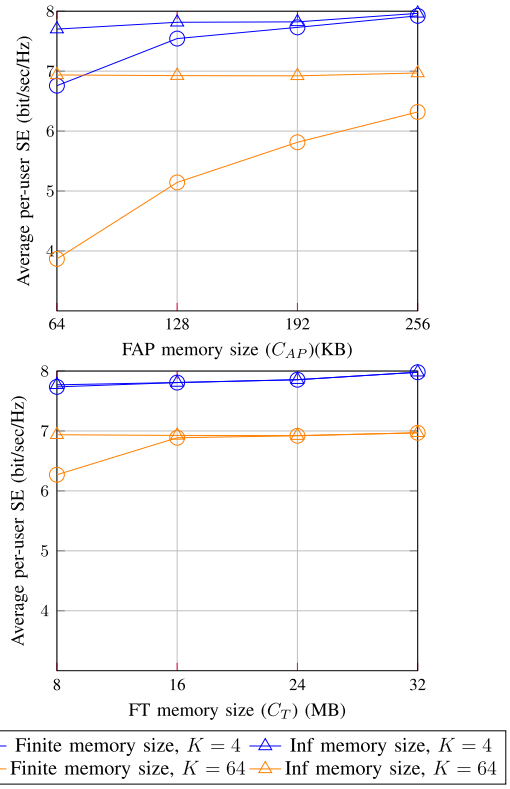


Fig. 5. Average per-user SE with (Top) varying the value of C_{AP} in FAP memory model and (Bottom) varying the value of C_T in FT memory model. The number of AP is fixed to $L = 32$. The compression option for all plots is VC.

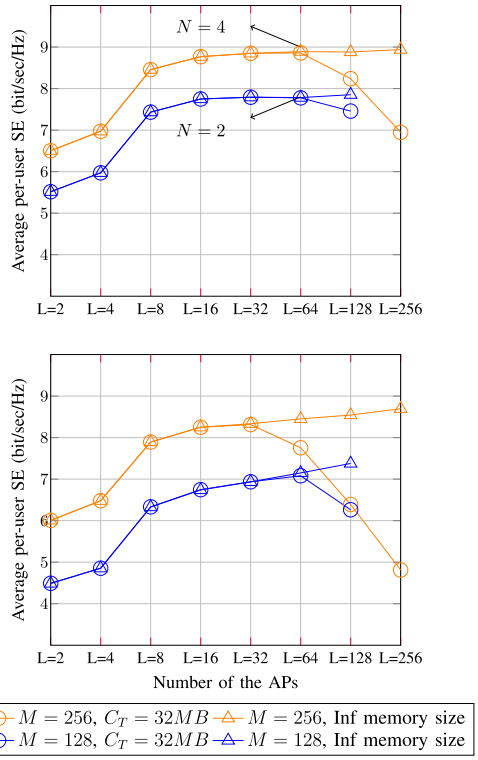


Fig. 6. Average per-user SE when having a different total number of antennas, i.e., $M = 128$ and $M = 256$ and two different numbers of users: (Top) $K = 4$ and (Bottom) $K = 64$. In all plots, the memory model is FT-EA with $C_T = 32MB$. The compression option for all plots is VC.

$K = 4$. This is because, in FT-EA, when the number of APs is halved, the allocated memory to each AP gets doubled as the same total memory is now distributed among 64 instead of 128 APs. While in FAP, the memory per AP remains the same in the case of $L = 64$ or $L = 128$.

- When comparing the different results in Fig. 4, it is clear that the impact of compression is much more severe when the number of users K is higher, as with increasing users, the entropy of the local received signal vector in each AP increases, which demands more bits to keep a certain distortion level. This is especially visible for the scenario of FAP, and the performance degradation is already visible for a low number of APs in the sequence. Even without severe memory requirements for sequential processing, limited AP memory degrades multiuser performance. Thus, when the number of users increases and the memory capacity is limited, it pays off to have collocated antennas.
- Figs. 5 testify to the aforementioned claim. It is observed that for a fixed number of APs L, the performance degradation of limited memory capacity cases compared to the infinite memory case gets more severe when the number of users increases.
- Furthermore, in Fig. 6, we plotted the average per-user SE for the case of $M = 256$ and compared it with the case of $M = 128$ when the FT-EA memory model with $C_T = 32MB$ is used. By increasing the total number of antennas from $M = 128$ to $M = 256$, we observe that the performance degradation compared to infinite memory gets worse. For the case of $K = 64$, the optimal number of APs decreases. This is because for a fixed number of APs, with an increasing total number of antennas M, we are increasing the number of antennas per AP (N), and hence, a higher dimensional received signal vector needs to be stored in each AP.

B. PCA-Pre-Processing Before Element-Wise Compression

In Fig. 7, the average per-user SE under three compression options and a FAP memory capacity scenario with an increasing number of users over sub-figures is studied. It is observed that:

- When the number of users is $K = 4$ and $C_{AP} = 64KB$, the performance improvement of VC compared to EC is around 1.2 bit/sec/Hz - at the optimal number of AP $L = 8$.
- In the two sub-figures, Element-wise compression of the PCA pre-processed vector results in the same performance as vector-wise compression of the received signal vector. This is because pre-processing the circularly symmetric Gaussian received signal vector with PCA de-correlates the elements of the vector, and for a circularly symmetric Gaussian vector, the de-correlation of the elements means their independence from each other. Our analysis shows that only in some particular scenarios, such as strict memory constraints and a low number of users, vector-wise compression improves over element-wise compression. Moreover, by pre-processing the data, there is no motivation for vector-wise compression.

Authorized licensed use limited to: KU Leuven Libraries. Downloaded on January 29,2025 at 13:57:53 UTC from IEEE Xplore. Restrictions apply.

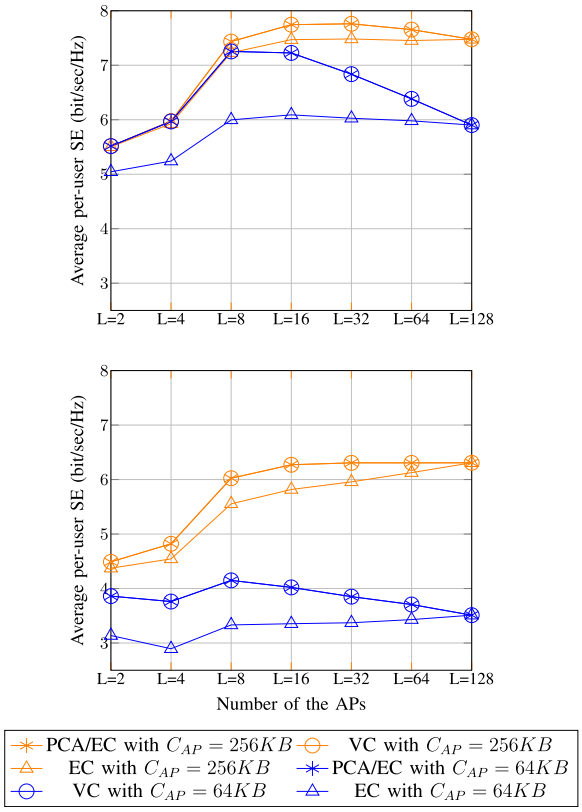


Fig. 7. Average per-user SE in a daisy chain fronthaul with three compression options and two different numbers of users: (Top) $K = 4$, (Bottom) $K = 64$. FAP memory model is used.

C. Impact of Imperfect CSI

Throughout the whole paper, we assumed perfect CSI for the mathematical tractibility. Specifically, assuming imperfect CSI in the presence of correlated Rayleigh fading, the transformations in problem (18) do not hold. However, we considered the uncorrelated Rayleigh fading scenario to show that the paper's conclusion also holds for imperfect CSI cases (due to pilot contamination, for example). We assumed that the pilot signal length is $\tau_p < K$. Then, the channel is estimated in the presence of pilot contamination, and the matrix $\mathbf{H}$ is replaced by its estimate. The impact of channel estimation error is reflected by adding the channel estimation error noise to σ^2. The simulation results in Fig. 8 validate the earlier discussion on the limited memory effects on optimal fronthaul architecture.

D. Equal Vs Linear Allocation of a Total Memory Budget

Intuitively, the memory capacity of the last AP becomes a bottleneck in the network with sequential topology. Therefore, it may seem like a solution that the memory capacity of the AP increases linearly (FT-LA), similar to the number of stored received signal vectors. Consequently, the APs at the end of the sequence get a larger share of the total memory. It is observed from Fig. 9 that FT-LA memory capacity allocation improves the average per-user SE only slightly around the optimal AP length where the memory constraint is not yet limiting. For a larger length of the sequential fronthaul, FT-LA even results in a lower performance compared to the FT-EA. With FT-LA, even though the received signal vectors of the APs at the end

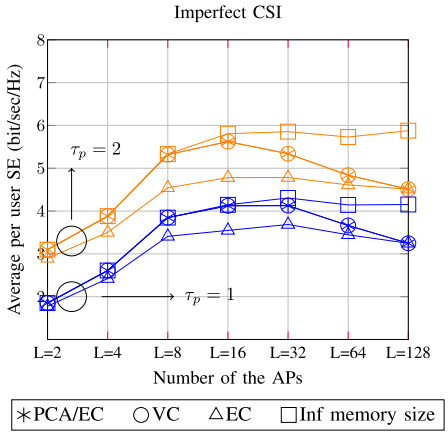


Fig. 8. Impact of imperfect CSI on average per-user SE when $K = 4$, $C_{AP} = 64KB$. The channel model is uncorrelated Rayleigh fading, and imperfect CSI is assumed with a pilot signal length of $\tau_p = 2$ or $\tau_p = 1$.

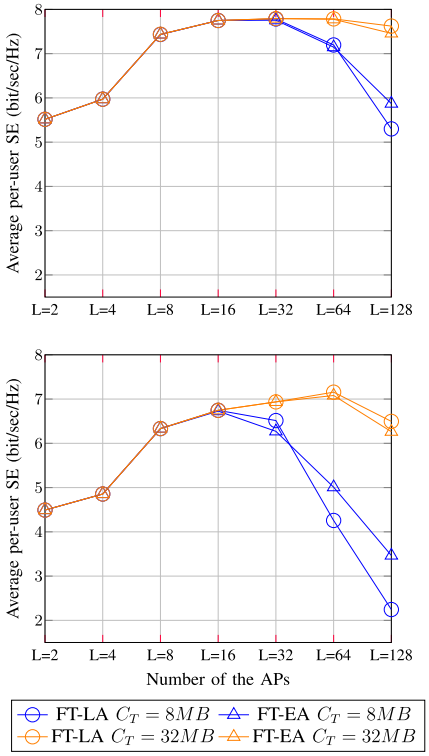


Fig. 9. Average per-user SE comparison between FT-EA and FT-LA memory model in a daisy chain fronthaul with two different numbers of users: (Top) $K = 4$, (Bottom) $K = 64$. VC is used for compression.

of the sequence can be compressed less, the compression noise power of the stored received signal vectors of the APs at the beginning of the sequence increases compared to the case of FT-EA. FT-LA will hence suffer from more noisy user signal estimation at first APs, upon which the subsequent APs should build their own estimation. Because of this error propagation in the sequence, giving more memory capacity to the last APs does not pay off. Thus, we conclude EA is highly preferred for longer sequences.

E. An Alternative Fronthaul Topology to Daisy Chain Topology: Multi-Branch Tree Topology

Finally, in Fig. 10, a multi-branch tree topology is compared with the daisy chain fronthaul topology. As in a multi-branch

Authorized licensed use limited to: KU Leuven Libraries. Downloaded on January 29,2025 at 13:57:53 UTC from IEEE Xplore. Restrictions apply.

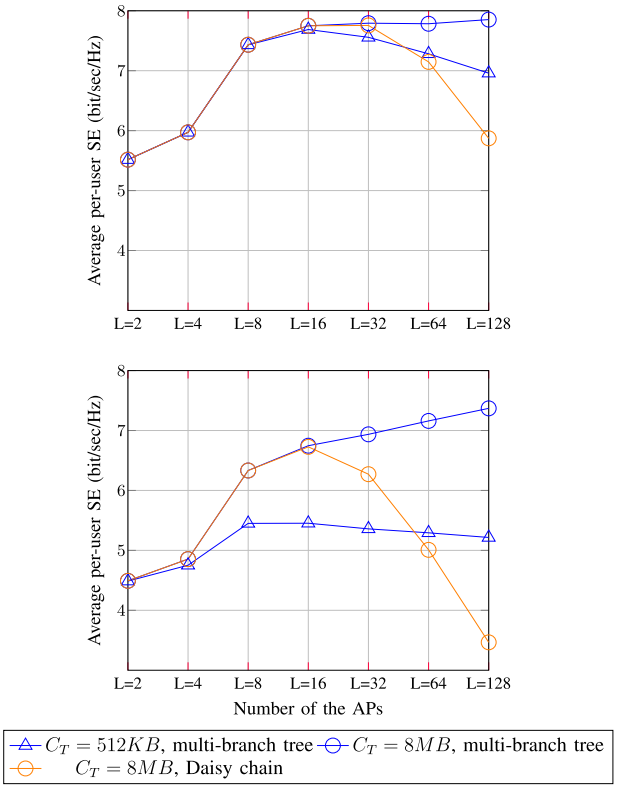


Fig. 10. Average per-user SE comparison between a daisy chain and multi-branch tree fronthaul topologies with two different numbers of users: (Top) $K = 4$, (Bottom) $K = 64$. FT-EA memory model is used, and the compression option is VC.

tree, the depth of the tree is less than in the case of the daisy chain topology (the depth of a daisy chain with $L = 128$ is 128 while the depth of a multi-branch tree with $L = 128$ and APs arranged as shown in Fig. 1 is 8 which is 16 times smaller.), we first consider a smaller total memory size, i.e., $C_T = 512KB$ (almost 16 times smaller than $C_T = 8MB$). As shown in Fig. 10, for $K = 4$, using a multi-branch tree topology with $C_T = 512KB$ have comparable peak performance to the daisy chain topology with $C_T = 8MB$, e.g., for $K = 4$, daisy chain with $C_T = 8MB$ has its peak performance of 7.8 bit/sec/Hz at $L = 32$, which is almost the same as peak performance of multi-branch tree topology with $C_T = 512KB$. Therefore, multi-branch topology can save around 0.94% memory cost compared to daisy chain topology. For a higher number of users, e.g., $K = 64$, multi-branch tree topology can improve the average per-user SE performance for around 9% (at $L = 128$) compared to the peak performance of the daisy chain topology (at $L = 16$) when the same memory capacity, i.e., $C_T = 8MB$, is used in both topologies. However, assuming the same total memory, i.e., $C_T = 8MB$, in the multi-branch tree topology, the average incoming fronthaul rate to the APs is larger than the daisy chain topology.

VI. Conclusion

In this paper, we considered the performance scaling with the number of APs in a CFmMIMO network under a realistic assumption of limited working memory in each AP. We considered two models for the memory capacity of the APs, the fixed per AP memory capacity model and the fixed total memory capacity model, and investigated the impact of using different memory models at the APs on the rate of the fronthaul link. Furthermore, the simulation result shows that the limited memory capacity at the APs limits the optimal number of APs in the sequence. This is in sharp contrast to the infinite memory case, which favors the maximal distribution of the antennas.

We further analyzed multiple compression options and showed that vector-wise compression, which is joint compression of all the antenna signals per AP, results in larger average per-user spectral efficiency gains. However, element-wise compression is used in practice for its simplicity. More importantly, by simply pre-processing the received signal vectors at each AP using PCA, element-wise compression achieves the same performance as rather expensive vector-wise compression in all considered scenarios.

Furthermore, we analyzed different memory allocation models. As the delay and number of samples to be stored at the APs increase linearly, a good option to allocate a total memory capacity may seem to be the linearly increasing memory allocation to the APs. However, the simulation results show that this memory allocation method doesn't bring much of a benefit as we allocate less memory capacity to the APs at the beginning of the sequence. Consequently, we start with a noisy compression of the received signal vector and a low-quality local estimate of the users' signal vector in the first AP, upon which the rest of the APs build their refined version of the users' signal vector estimate.

Finally, we show that when the number of users is relatively low, i.e., $K = 4$, using a multi-branch tree topology can save the memory cost more than 90%, compared to daisy chain topology, with a similar average per-user SE peak performance. When the number of users gets larger, i.e., $K = 64$, multi-branch tree topology can improve the average per-user SE for around 9% compared to the daisy chain fronthaul with the same total memory, i.e., $C_T = 8MB$, however, at the cost of increase average incoming fronthaul rate to the APs.

Appendix A

Before proving the equality $\overset{(a)}{=}$ and inequality $\overset{(b)}{\leq}$ in (17, Theorem 1 is provided.

Theorem 1: : For matrices $\mathbf{A}$ and $\mathbf{B}$ with size $n \times m$ and $m \times n$ respectively, the following equality holds:

$$\det(\mathbf{I}_n + \mathbf{AB}) = \det(\mathbf{I}_m + \mathbf{BA}). \tag{51}$$

Proof of Theorem 1: Theorem 1 is Sylvester's determinant theorem and can be proved easily [39].

A. proof of $\overset{(a)}{=}$ in (17)

We know that the vector $\{\hat{\mathbf{y}}_v|\mathbf{H}\}$ is a circularly symmetric Gaussian random vector because:

$$\{\hat{\mathbf{y}}_v|\mathbf{H}\} = \{\mathbf{y}|\mathbf{H}\} + \mathbf{q}_v \tag{52}$$

where $\{\mathbf{y}|\mathbf{H}\}$ and $\mathbf{q}_v$ are two independent circularly symmetric Gaussian random vectors and based on the definition of the circularly symmetric Gaussian random vectors in

Authorized licensed use limited to: KU Leuven Libraries. Downloaded on January 29,2025 at 13:57:53 UTC from IEEE Xplore. Restrictions apply.

[28, Appendix A], their summation is also circularly symmetric Gaussian random vector. Furthermore, any linear transformation of a circularly symmetric Gaussian vector results in a vector that is a circularly symmetric Gaussian vector [28, Appendix A], which concludes that $\hat{\mathbf{s}}_v = \hat{\mathbf{C}}_v \hat{\mathbf{y}}_v$, defined in (15) is a circularly symmetric Gaussian vector. Similarly, the signal estimation error vector $\mathbf{e}$ can also be proved to be a circularly symmetric Gaussian vector, independent of $\hat{\mathbf{s}}_v$. Therefore,

$$\begin{aligned}\mathcal{I}(\hat{\mathbf{s}}_v;\mathbf{s}) &= \mathcal{H}(\mathbf{s}) - \mathcal{H}(\mathbf{s}|\hat{\mathbf{s}}_v)\\ &= \mathcal{H}(\mathbf{s}) - \mathcal{H}(\underbrace{\mathbf{s}-\hat{\mathbf{s}}_v}_{\mathbf{e}})\\ &= \log_2\det(\pi e p\mathbf{I}_K) - \log_2\det(\pi e\boldsymbol{\Gamma}_L)\\ &= \log_2\det(p\boldsymbol{\Gamma}_L^{-1})\\ &= \log_2\det(p\mathbf{H}^{\mathrm{H}}\mathbf{Z}_v^{-1}\mathbf{H}+\mathbf{I}_K). \end{aligned} \tag{53}$$

where we substitute the $\boldsymbol{\Gamma}_L$ in (53) with its equivalent matrix as in (5) and used Theorem 1 to conclude the proof of $\overset{(a)}{=}$ in (17).

B. Proof of $\overset{(b)}{\leq}$ in (17)

To prove inequality $\overset{(b)}{\leq}$ in (17), Theorem 1 is used. Using Theorem 1:

$$\begin{aligned}&\log_2\det(p\mathbf{H}\mathbf{H}^{\mathrm{H}}\mathbf{Z}_v^{-1}+\mathbf{I}_{NL})\\ &\quad= \log_2\det(p\mathbf{H}\mathbf{H}^{\mathrm{H}}\mathbf{Z}_v^{-1/2}\mathbf{Z}_v^{-H/2}+\mathbf{I}_{NL})\\ &\quad= \log_2\det(p\mathbf{Z}_v^{-\mathrm{H}/2}\mathbf{H}\mathbf{H}^{\mathrm{H}}\mathbf{Z}_v^{-1/2}+\mathbf{I}_{NL}). \end{aligned} \tag{54}$$

where $\mathbf{Z}_v^{-1/2} = \mathrm{blkdiag}(\mathbf{Z}_{v1}^{-1/2},\ldots,\mathbf{Z}_{vL}^{-1/2})$ is the square root of $\mathbf{Z}_v^{-1}$. Matrix $\mathbf{Z}_v^{-\mathrm{H}/2}\mathbf{H}\mathbf{H}^{\mathrm{H}}\mathbf{Z}_v^{-1/2}$ is positive semi-definite and for positive semi-definite matrices, invoking the Hadamard's inequality [40] and subsequently Theorem 1 leads to :

$$\begin{aligned}&\log_2\det(p\mathbf{Z}_v^{-\mathrm{H}/2}\mathbf{H}\mathbf{H}^{\mathrm{H}}\mathbf{Z}_v^{-1/2}+\mathbf{I}_{NL})\\ &\quad\leq \log_2\prod_{l=1}^{L}\det(p\mathbf{Z}_{vl}^{-\mathrm{H}/2}\mathbf{H}_l\mathbf{H}_l^{\mathrm{H}}\mathbf{Z}_{vl}^{-1/2}+\mathbf{I}_N)\\ &\quad= \sum_{l=1}^{L}\log_2\det(p\mathbf{H}_l\mathbf{H}_l^{\mathrm{H}}\mathbf{Z}_{vl}^{-1}+\mathbf{I}_N). \end{aligned} \tag{55}$$

APPENDIX B

To prove that the solution derived in (22) is globally optimal, Theorem 2 is used, which states under which conditions the proposed solution is optimal. In the following, we first prove why satisfying the conditions in Theorem 2 makes a solution optimal. Then, we prove that the solution provided in (22) satisfies the condition in Theorem 2 and hence, is optimal.

Theorem 2: Consider the problem defined in (18). $(\mathbf{Q}_{vl}^{-1})^o$ and μ_{vl}^o are optimal and strong duality holds, if and only if

(Primal feasibility)

$$C_{sc} = \log_2\det\big((\mathbf{Q}_{vl}^{-1})^o(p\mathbf{H}_l\mathbf{H}_l^{\mathrm{H}}+\sigma^2\mathbf{I}_N)+\mathbf{I}_N\big),\ (\mathbf{Q}_{vl}^{-1})^o \succeq \mathbf{0}$$

(Lagrange optimality)

$$(\mathbf{Q}_{vl}^{-1})^o = \arg\max_{\mathbf{Q}_{vl}^{-1}\succeq\mathbf{0}} \mathcal{L}(\mathbf{Q}_{vl}^{-1},\mu_{vl}^o), \tag{56}$$

where the Lagrange function is defined as follows:

$$\begin{aligned}&\mathcal{L}(\mathbf{Q}_{vl}^{-1},\mu_{vl})\\ &\quad= (1-\mu_{vl})\log_2\det(\mathbf{Q}_{vl}^{-1}(p\mathbf{H}_l\mathbf{H}_l^{\mathrm{H}}+\sigma^2\mathbf{I}_N)+\mathbf{I}_N)\\ &\quad- \log_2\det(\sigma^2\mathbf{Q}_{vl}^{-1}+\mathbf{I}_N)+\mu_{vl}C_{sc}. \end{aligned} \tag{57}$$

Proof of Theorem 2: Theorem 2 can be proved based on Proposition 6.2.5 in [41]. Before starting the proof, we define the objective function as:

$$\begin{aligned}F(\mathbf{Q}_{vl}^{-1}) &= \log_2\det(\mathbf{Q}_{vl}^{-1}(p\mathbf{H}_l\mathbf{H}_l^{\mathrm{H}}+\sigma^2\mathbf{I}_N)+\mathbf{I}_N)\\ &\quad- \log_2\det(\sigma^2\mathbf{Q}_{vl}^{-1}+\mathbf{I}_N) \end{aligned} \tag{58}$$

In the following, we prove that Theorem 2 provides necessary and sufficient conditions for the optimality of our solution.

- If $(\mathbf{Q}_{vl}^{-1})^o$ and μ_{vl}^o are optimal primal and dual variables for which strong duality holds, they must satisfy the following conditions:
  - For any $(\mathbf{Q}_{vl}^{-1})^o$ to be optimal, it must be in the feasible set obviously, so the primal feasibility of Theorem 2 is achieved.
  - For the Lagrange optimality condition, as $(\mathbf{Q}_{vl}^{-1})^o$ and μ_{vl}^o are primal and dual optimal and based on the Lagrange function definition in (57),

$$\begin{aligned}F^o &= F((\mathbf{Q}_{vl}^{-1})^o)\\ &\overset{(a)}{=} \mathcal{L}((\mathbf{Q}_{vl}^{-1})^o,\mu_{vl}^o)\\ &= \max_{\mathbf{Q}_{vl}^{-1}\succeq\mathbf{0},h(\mathbf{Q}_{vl}^{-1})=0}\mathcal{L}(\mathbf{Q}_{vl}^{-1},\mu_{vl}^o)\\ &\leq \max_{\mathbf{Q}_{vl}^{-1}\succeq\mathbf{0}}\mathcal{L}((\mathbf{Q}_{vl}^{-1}),\mu_{vl}^o)\overset{(b)}{=}F^o \end{aligned} \tag{59}$$

    where $\overset{(a)}{=}$ holds due to the primal and dual optimality of $(\mathbf{Q}_{vl}^{-1})^o$ and μ_{vl}^o respectively and $\overset{(b)}{=}$ holds due to strong duality. Based on (59), equality holds throughout (59) and we have:

$$(\mathbf{Q}_{vl}^{-1})^o = \arg\max_{\mathbf{Q}_{vl}^{-1}\succeq\mathbf{0}}\mathcal{L}((\mathbf{Q}_{vl}^{-1}),\mu_{vl}^o). \tag{60}$$

    which proves the Lagrange optimality condition in Theorem 2.
- Conversely, If the primal feasibility and Lagrange optimality conditions in Theorem 2 hold, then we can prove that $(\mathbf{Q}_{vl}^{-1})^o$ and μ_{vl}^o are optimal primal and dual variables for which strong duality holds. This can be easily proved with the help of the fact that weak duality always holds. The proof is omitted due to its simplicity, and interested readers are referred to 6.2.5 in [41].

Based on Theorem 2 and its proof, to prove that the matrix in (21) is globally optimal (over the range of all positive semi-definite matrices), it should satisfy the optimality conditions in (56). To prove that the matrix in (21) optimizes the Lagrange function, Lemma 1 is used.

Authorized licensed use limited to: KU Leuven Libraries. Downloaded on January 29,2025 at 13:57:53 UTC from IEEE Xplore. Restrictions apply.

Lemma 1*:* : [11, Appendix B] If matrix $\mathbf{A}$ and $\mathbf{B}$ are positive (semi-)definite matrix and $\mathbf{\Gamma_A}$ and $\mathbf{\Gamma_B}$ have the ordered eigenvalues of $\mathbf{A}$ and $\mathbf{B}$ on their diagonal respectively, then:

$$\det(\mathbf{AB}+\mathbf{I}_n) \leq \det(\mathbf{\Gamma_A}\mathbf{\Gamma_B}+\mathbf{I}_n), \tag{61}$$

with equality if the eigenvectors of $\mathbf{A}$ is conjugate transpose of eigen vectors of $\mathbf{B}$.

Using Lemma 1, the Lagrange function in (57) can be upper bounded as follows:

$$\begin{aligned}\mathcal{L}(\mathbf{Q}_{vl}^{-1},\mu_{vl}^{o}) &\leq (1-\mu_{vl}^{o})\log_2\det(\mathbf{\Sigma}_{vlq}^{-1}(p\mathbf{\Sigma}_l\mathbf{\Sigma}_l^{\mathrm{H}}+\sigma^2\mathbf{I}_N)+\mathbf{I}_N)\\ &\quad-\log_2\det(\sigma^2\mathbf{\Sigma}_{vlq}^{-1}+\mathbf{I}_N)+\mu_{vl}^{o}C_{sc},\end{aligned} \tag{62}$$

where $\mathbf{\Sigma}_{vlq}^{-1}$ is the matrix whose i^{th} diagonal element is the i^{th} eigenvalues of $\mathbf{Q}_{vl}^{-1}$, i^{th} diagonal elements of $\mathbf{\Sigma}_l\mathbf{\Sigma}_l^{\mathrm{H}}$ is given as λ_{il}^2. Therefore (62) can be re-written as follows [40]:

$$\begin{aligned}\mathcal{L}(\mathbf{Q}_{vl}^{-1},\mu_{vl}^{o}) &\leq (1-\mu_{vl}^{o})\sum_{i=1}^{N}\log_2(\lambda_{vlqi}(p\lambda_{li}^2+\sigma^2)+1)\\ &\quad-\log_2(\lambda_{vlqi}\sigma^2+1)+\mu_{vl}^{o}C_{sc}\\ &=\sum_{i=1}^{N}f_i(\lambda_{vlqi})+\mu_{vl}^{o}C_{sc},\end{aligned} \tag{63}$$

where each of the functions $f_i, \forall i \in 1,\ldots,N$ depends only on one variable $\lambda_{vlqi}, \forall i \in 1,\ldots,N$,

$$\begin{aligned}f_i(\lambda_{vlqi}) &= (1-\mu_{vl}^{o})\log_2(\lambda_{vlqi}(p\lambda_{li}^2+\sigma^2)+1)\\ &\quad-\log_2(\lambda_{vlqi}\sigma^2+1).\end{aligned} \tag{64}$$

The derivative of $f_i(\lambda_{vlqi})$ with respect to λ_{vlqi} is zero at λ_{vlqi}^{o}, defined as follows:

$$\lambda_{vlqi}^{o}=\frac{1}{\mu_{vl}^{o}}\left(\frac{1}{\sigma^2}-\frac{1}{p\lambda_{li}^2+\sigma^2}\right)-\frac{1}{\sigma^2}, \forall i. \tag{65}$$

Following the discussion on [11, Appendix. B], λ_{vlqi}^{o} in (65) globally maximizes $f_i, \forall i \in 1,\ldots,N$. However, restricting $\lambda_{vlqi}^{o} \geq 0$, the maximizer of $f_i, \forall i \in 1,\ldots,N$ is as follows:

$$\lambda_{vlqi}^{o}=\max\left(0,\frac{1}{\mu_{vl}^{o}}\left(\frac{1}{\sigma^2}-\frac{1}{p\lambda_{li}^2+\sigma^2}\right)-\frac{1}{\sigma^2}\right), \forall i, \tag{66}$$

where μ_{vl}^{o} is derived by inserting (66) into equality constraint of (18). Inserting the optimal values given in (66) into the upper bound in (63), the Lagrange function will be upper bounded as follows:

$$\begin{aligned}\mathcal{L}(\mathbf{Q}_{vl}^{-1},\mu_{vl}^{o}) &\leq (1-\mu_{vl}^{o})\sum_{i=1}^{N}\log_2(\lambda_{vlqi}^{o}(p\lambda_{li}^2+\sigma^2)+1)\\ &\quad-\log_2(\lambda_{vlqi}^{o}\sigma^2+1)+\mu_{vl}^{o}C_{sc}\\ &=\mathcal{L}((\mathbf{Q}_{vl}^{-1})^{o},\mu_{vl}^{o}).\end{aligned} \tag{67}$$

Hence, we have $(\mathbf{Q}_{vl}^{-1})^{o} = \arg\max_{\mathbf{Q}_{vl}^{-1}\succeq\mathbf{0}}\mathcal{L}((\mathbf{Q}_{vl}^{-1}),\mu_{vl}^{o})$. Consequently, both optimality conditions in (56) are satisfied for $(\mathbf{Q}_{vl}^{-1})^{o},\mu_{vl}^{o}$, which means that $(\mathbf{Q}_{vl}^{-1})^{o},\mu_{vl}^{o}$ are the global maximizer of the maximization problem defined in (18).

ACKNOWLEDGMENT

The authors would like to thank Panagiotis (Panos) Patrinos for improving and simplifying the article's appendix.

REFERENCES

[1] V. Ranjbar, R. Beerten, M. Moonen, and S. Pollin, "Sequential processing in cell-free massive MIMO uplink with limited memory access points," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Dec. 2023, pp. 20–25.

[2] E. G. Larsson, O. Edfors, F. Tufvesson, and T. L. Marzetta, "Massive MIMO for next generation wireless systems," *IEEE Commun. Mag.*, vol. 52, no. 2, pp. 186–195, Feb. 2014.

[3] E. Nayebi, A. Ashikhmin, T. L. Marzetta, and H. Yang, "Cell-free massive MIMO systems," in *Proc. 49th Asilomar Conf. Signals, Syst. Comput.*, Nov. 2015, pp. 695–699.

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

[5] 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. 2020.

[6] G. Interdonato, E. Björnson, H. Q. Ngo, P. Frenger, and E. G. Larsson, "Ubiquitous cell-free massive MIMO communications," *EURASIP J. Wireless Commun. Netw.*, vol. 2019, no. 1, p. 197, Dec. 2019.

[7] Z. H. Shaik, E. Björnson, and E. G. Larsson, "Cell-free massive MIMO with radio stripes and sequential uplink processing," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, Jun. 2020, pp. 1–6.

[8] Z. H. Shaik, E. Björnson, and E. G. Larsson, "MMSE-optimal sequential processing for cell-free massive MIMO with radio stripes," *IEEE Trans. Commun.*, vol. 69, no. 11, pp. 7775–7789, Nov. 2021.

[9] K. W. Helmersson, P. Frenger, and A. Helmersson, "Uplink D-MIMO processing using Kalman filter combining," in *Proc. GLOBECOM IEEE Global Commun. Conf.*, Dec. 2022, pp. 1703–1708.

[10] V. Ranjbar, S. Pollin, and M. Moonen, "Finite precision implementation of recursive algorithms for uplink detection in cell-free networks," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Dec. 2022, pp. 25–30.

[11] A. del Coso and S. Simoens, "Distributed compression for MIMO coordinated networks with a backhaul constraint," *IEEE Trans. Wireless Commun.*, vol. 8, no. 9, pp. 4698–4709, Sep. 2009.

[12] E. Björnson, L. Sanguinetti, and J. Hoydis, "Hardware distortion correlation has negligible impact on UL massive MIMO spectral efficiency," *IEEE Trans. Commun.*, vol. 67, no. 2, pp. 1085–1098, Feb. 2019.

[13] X. Hu, C. Zhong, X. Chen, W. Xu, H. Lin, and Z. Zhang, "Cell-free massive MIMO systems with low resolution ADCs," *IEEE Trans. Commun.*, vol. 67, no. 10, pp. 6844–6857, Oct. 2019.

[14] D. Verenzuela, E. Björnson, and M. Matthaiou, "Optimal per-antenna ADC bit allocation in correlated and cell-free massive MIMO," *IEEE Trans. Commun.*, vol. 69, no. 7, pp. 4767–4780, Jul. 2021.

[15] Y. Xiong et al., "Asymptotic analysis for cell-free massive MIMO with MMSE combining and low-resolution ADCs," *IEEE Commun. Lett.*, vol. 25, no. 10, pp. 3219–3223, Oct. 2021.

[16] Y. Xiong, S. Sun, L. Liu, Z. Zhang, and N. Wei, "Performance analysis and bit allocation of cell-free massive MIMO network with variable-resolution ADCs," *IEEE Trans. Commun.*, vol. 71, no. 1, pp. 67–82, Jan. 2023.

[17] M. Bashar, K. Cumanan, A. G. Burr, H. Q. Ngo, M. Debbah, and P. Xiao, "Max-min rate of cell-free massive MIMO uplink with optimal uniform quantization," *IEEE Trans. Commun.*, vol. 67, no. 10, pp. 6796–6815, Oct. 2019.

[18] M. Bashar, K. Cumanan, A. G. Burr, H. Q. Ngo, E. G. Larsson, and P. Xiao, "On the energy efficiency of limited-backhaul cell-free massive MIMO," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2019, pp. 1–7.

[19] M. Bashar et al., "Uplink spectral and energy efficiency of cell-free massive MIMO with optimal uniform quantization," *IEEE Trans. Commun.*, vol. 69, no. 1, pp. 223–245, Jan. 2021.

[20] M. Bashar, K. Cumanan, A. G. Burr, H. Q. Ngo, E. G. Larsson, and P. Xiao, "Energy efficiency of the cell-free massive MIMO uplink with optimal uniform quantization," *IEEE Trans. Green Commun. Netw.*, vol. 3, no. 4, pp. 971–987, Dec. 2019.

[21] H. Masoumi and M. J. Emadi, "Performance analysis of cell-free massive MIMO system with limited fronthaul capacity and hardware impairments," *IEEE Trans. Wireless Commun.*, vol. 19, no. 2, pp. 1038–1053, Feb. 2020.

Authorized licensed use limited to: KU Leuven Libraries. Downloaded on January 29,2025 at 13:57:53 UTC from IEEE Xplore. Restrictions apply.

[22] D. Maryopi, M. Bashar, and A. Burr, "On the uplink throughput of zero forcing in cell-free massive MIMO with coarse quantization," *IEEE Trans. Veh. Technol.*, vol. 68, no. 7, pp. 7220–7224, Jul. 2019.
[23] R. Van Rompaey and M. Moonen, "Scalable and distributed MMSE algorithms for uplink receive combining in cell-free massive MIMO systems," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2021, pp. 4445–4449.
[24] E. Björnson and L. Sanguinetti, "Scalable cell-free massive MIMO systems," *IEEE Trans. Commun.*, vol. 68, no. 7, pp. 4247–4261, Jul. 2020.
[25] J. R. Sanchez, F. Rusek, M. Sarajlic, O. Edfors, and L. Liu, "Fully decentralized massive MIMO detection based on recursive methods," in *Proc. IEEE Int. Workshop Signal Process. Syst.*, Oct. 2018, pp. 53–58.
[26] J. R. Sánchez, F. Rusek, O. Edfors, M. Sarajlic, and L. Liu, "Decentralized massive MIMO processing exploring daisy-chain architecture and recursive algorithms," *IEEE Trans. Signal Process.*, vol. 68, pp. 687–700, 2020.
[27] E. Björnson, J. Hoydis, and L. Sanguinetti, "Massive MIMO networks: Spectral, energy, and hardware efficiency," *Found. Trends Signal Process.*, vol. 11, nos. 3–4, pp. 154–655, 2017.
[28] T. L. Marzetta, E. G. Larsson, H. Yang, and H. Q. Ngo, *Fundamentals of Massive MIMO*. Cambridge, U.K.: Cambridge Univ. Press, 2016.
[29] J. R. Sánchez, "Systems with massive number of antennas: Distributed approaches," Ph.D. thesis, Dept. Elect. Inf. Technol., Lund Univ., Lund, Sweden, 2022.
[30] T. Report. (2019). *Arm Neoverse E1 Core, Technical Reference Manual*. [Online]. Available: https://developer.arm.com/documentation/101560/latest
[31] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. Hoboken, NJ, USA: Wiley, 2012.
[32] A. Zahedi, J. Østergaard, S. H. Jensen, P. Naylor, and S. Bech, "Distributed remote vector Gaussian source coding with covariance distortion constraints," in *Proc. IEEE Int. Symp. Inf. Theory*, Jun. 2014, pp. 586–590.
[33] A. El Gamal and Y.-H. Kim, *Network Information Theory*. Cambridge, U.K.: Cambridge Univ. Press, 2011.
[34] J. Kang, O. Simeone, J. Kang, and S. S. Shitz, "Joint signal and channel state information compression for the backhaul of uplink network MIMO systems," *IEEE Trans. Wireless Commun.*, vol. 13, no. 3, pp. 1555–1567, Mar. 2014.
[35] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*. Cambridge, U.K.: Cambridge Univ. Press, 2005.
[36] D. A. Patterson and J. L. Hennessy, *Computer Organization and Design: The Hardware/Software Interface*, 5th ed. San Francisco, CA, USA: Morgan Kaufmann, 2013.
[37] *Further Advancements for E-UTRA Physical Layer Aspects (Re-Lease 9)*, Standard 3GPP TS 36.814, 3GPP, Mar. 2017.
[38] *NR; User Equipment (UE) Radio Transmission and Reception; Part 1: Range 1 Standalone*, Standard 3GPP TS 38.101-1, Mar. 2023.
[39] C. Pozrikidis, *An Introduction to Grids, Graphs, and Networks*. London, U.K.: Oxford Univ. Press, 2014.
[40] K. B. Petersen and M. S. Pedersen, *The Matrix Cookbook*. Denmark: Technical Univ. of Denmark, 2012.
[41] D. Bertsekas, *Convex Analysis and Optimization*. Belmont, MA, USA: Athena Scientific, 2003.



Vida Ranjbar (Graduate Student Member, IEEE) received the B.Sc. degree in electrical engineering from Babol University of Technology, Iran, in 2014, and the M.Sc. degree in electrical engineering from the University of Tehran, Iran, in 2017. She is currently pursuing the Ph.D. degree with the Department of Electrical Engineering (ESAT), KU Leuven, Belgium. Her research interests include signal processing, quantization, and optimization in massive MIMO systems.



Robbert Beerten (Graduate Student Member, IEEE) received the B.Sc. degree in engineering and the M.Sc. degree in electrical engineering from KU Leuven, Belgium, in 2019 and 2021, respectively. He is currently pursuing the Ph.D. degree with a focus on optimization and distributed signal processing for next-generation networks.



Marc Moonen (Fellow, IEEE) is currently a Full Professor with the Department of Electrical Engineering, KU Leuven, where he is heading a research team working in the area of numerical algorithms and signal processing for digital communications, wireless communications, DSL, and audio signal processing.

Dr. Moonen is a fellow of the European Association for Signal Processing (EURASIP) in 2018. He was a 1997 Laureate of Belgium Royal Academy of Science and received the KU Leuven Research Council Award in 1994, the Alcatel Bell Belgium Award (with Piet Vandaele) in 1997, the Alcatel Bell Belgium Award (with Raphael Cendrillon) in 2004, and the CELTIC Excellence Award for EU/CELTIC Project 'BANITS' (with project partners) in 2008. He was also a recipient of the Journal Best Paper Awards from IEEE TRANSACTIONS ON SIGNAL PROCESSING (with Geert Leus and Daniele Giacobello) and *Signal Processing* (Elsevier) (with Simon Doclo), and the Conference Best Paper Awards from IEEE ICASSP (with Ann Spriet), IEEE ICC (with R. Moraes, P. Tsiaflakis, and Jeroen Van Dyck), IEEE-IWAENC (with Ann Spriet and Simon Doclo), and IEEE-WIC Symposium (with Geert Van Meerbergen). He was the IEEE Signal Processing Society Vice President-Publications from 2021 to 2023, the Chairperson of the IEEE Benelux Signal Processing Chapter from 1998 to 2002, a member of the IEEE Signal Processing Society Awards Board from 2008 to 2010, a member of the IEEE Signal Processing Society Technical Committee on Signal Processing for Communications from 2005 to 2007 and from 2008 to 2010, and a member of the IEEE Signal Processing Society Fellow Selection Committee from 2014 to 2016. He was the President of the European Association for Signal Processing (EURASIP) from 2007 to 2008 and from 2011 to 2012, a member of the Board of Directors of EURASIP from 2000 to 2016, and the Chairperson of the EURASIP Fellow Selection Committee from 2019 to 2021. He was the Editor-in-Chief of the *EURASIP Journal on Applied Signal Processing* from 2003 to 2005, an Area Editor for Feature Articles in *IEEE Signal Processing Magazine* from 2012 to 2014, and has been a member of the editorial board of Signal Processing, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—II: EXPRESS BRIEFS, *IEEE Signal Processing Magazine*, *Integration, the VLSI Journal*, *EURASIP Journal on Wireless Communications and Networking*, and *EURASIP Journal on Advances in Signal Processing*.



Sofie Pollin (Senior Member, IEEE) is currently a Professor at KU Leuven, Belgium, with a focus on wireless communication systems. Before that, she worked at imec and the University of California at Berkeley, and she is currently a Principal Member of the technical staff at imec. Her research centers around wireless networks that require networks that are ever more dense, heterogeneous, battery-powered, and spectrum-constrained. Her research interests include cell-free networks, integrated communication and sensing, and non-terrestrial networks.

Authorized licensed use limited to: KU Leuven Libraries. Downloaded on January 29,2025 at 13:57:53 UTC from IEEE Xplore. Restrictions apply.