

MIMO in network simulators: Design, implementation and evaluation of single-user MIMO in ns-3 5G-LENA

Biljana Bojović, Sandra Lagén

*Centre Tecnològic de Telecomunicacions de Catalunya (CTTC/CERCA), Avinguda Carl
Friedrich Gauss, 7, 08860 Castelldefels, Barcelona, Spain*

Abstract

MIMO technology has been studied in textbooks for several decades, and it has been adopted in 4G and 5G systems. Due to the recent evolution in 5G and beyond networks, designed to cover a wide range of use cases with every time more complex applications, it is essential to have network simulation tools (such as ns-3) to evaluate MIMO performance from the network perspective, before real implementation. Up to date, the well-known ns-3 simulator has been missing the inclusion of single-user MIMO (SU-MIMO) models for 5G. In this paper, we detail the implementation models and provide an exhaustive evaluation of SU-MIMO in the 5G-LENA module of ns-3. As per 3GPP 5G, we adopt a hybrid beamforming architecture and a closed-loop MIMO mechanism and follow all 3GPP specifications for MIMO implementation, including channel state information feedback with precoding matrix indicator and rank indicator reports, and codebook-based precoding following Precoding Type-I (used for SU-MIMO). The simulation models are released in open-source and currently support up to 32 antenna ports and 4 streams per user. The simulation results presented in this paper help in testing and verifying the simulated models, for different multi-antenna array and antenna ports configurations.

Keywords: ns-3, network simulator, 5G NR, 5G-LENA, full MIMO, spatial multiplexing, SU-MIMO, precoding Type-I

Email addresses: `biljana.bojovic@cttc.es` (Biljana Bojović), `sandra.lagen@cttc.es` (Sandra Lagén)

1. Introduction

The use of multiple antennas in communication systems (a.k.a., Multiple-Input Multiple-Output (MIMO) systems) has attracted a lot of attention in the past decades. MIMO permits increasing the data rate by sending multiple data streams simultaneously (known as spatial multiplexing) or increasing the robustness of the data transmission by sending replicated data (known as transmit diversity) or providing array gain into specific spatial areas by properly designing the antenna weights or beamforming vectors (known as beamforming) [1, 2]. Beamforming is particularly essential at high carrier frequencies within the millimeter wave (mmWave) region to combat the high pathloss propagation losses and blocking effects [3, 4, 5]. On the other hand, spatial multiplexing has the ability to approach the maximum channel capacity, and at the same time adapt the number of transmitted data streams (or layers) to the available propagation conditions. Due to the various trade-offs between beamforming and spatial multiplexing, in terms of performance and implementation complexity, hybrid precoding architectures arose as a very promising MIMO solution for 5G New Radio (NR) and future mobile network systems [6]. In hybrid precoding architectures, digital baseband processing (for spatial multiplexing plus digital beamforming, also known as precoding) and analog processing (for analog beamforming) are combined [7].

The 3rd Generation Partnership Project (3GPP) has already adopted the hybrid precoding architecture for MIMO in 5G NR, by introducing the concept of antenna ports, as shown in Fig. 1. This way, multiple data streams are mapped into multiple antenna ports by means of digital baseband processing (precoding), and then the multiple antenna ports are mapped to antenna elements by means of analog beamforming. For the antenna ports to antenna elements mapping, 3GPP supports either a sub-connected structure (each antenna port is connected to a unique subset of antenna elements) or a fully connected structure (each antenna port is connected to all antenna elements).

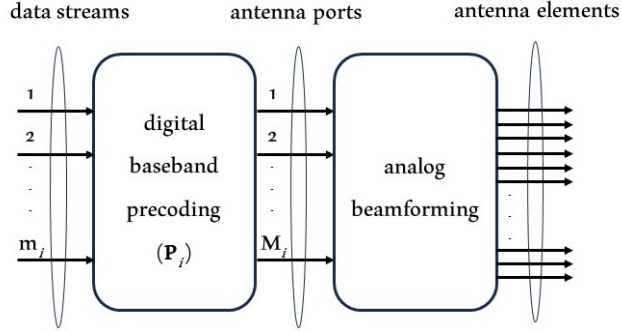


Figure 1: Hybrid precoding structure for MIMO systems in 3GPP 5G NR.

For the data streams to antenna ports mapping, 3GPP has adopted for 5G NR a closed-loop MIMO mechanism with codebook-based precoding, to reduce the feedback complexity and overhead. In closed-loop MIMO, the Channel State Information (CSI) is acquired at the base station (gNB) thanks to CSI feedback from the user equipment (UE), which includes the Precoding Matrix Indicator (PMI), the Rank Indicator (RI) (i.e., the number of streams), and the Channel Quality Indicator (CQI) [6]. Owing to the codebook-based precoding, the PMI selection (done at the UE side) results in an index of a set of predefined precoding matrices in a codebook. 5G NR supports two types of codebooks for MIMO: Type-I and Type-II [6]. Type-I codebook is mainly used for single-user MIMO (SU-MIMO) (which sends multiple streams towards the same user) and it is based on the same logic as LTE codebook. On the other hand, Type-II codebook is mostly for multi-user MIMO (MU-MIMO) (which sends multiple streams but each stream towards a different user) and it adopts a more general mathematical formula with many of parameters.

To support the evaluation of MIMO technology in 5G mobile networks, system-level simulators that include an end-to-end perspective of the network with a complete protocol stack but at the same time, accurate physical (PHY) layer modeling for MIMO are needed. Network simulators such as **ns-3** [8] offer a powerful tool to evaluate different “hot topic” 5G scenarios through complete network modeling and complex applications, such as augmented, virtual,

and mixed reality in 5G and beyond [9, 10]. At the same time, to allow the simulation of such demanding applications, it is necessary to enable realistic MIMO models in the simulation, including SU-MIMO and MU-MIMO. Accurate MIMO models are needed to be able to assess realistic XR application layer performance in 5G, which can be greatly affected by the performance of different MIMO implementations, i.e., PMI search/selection algorithms. In parallel, while on the road towards 6G, there is more and more interest in the research community in energy-efficient solutions for UE devices. Note that the computational complexity of a UE supporting MIMO directly translates into the UE's energy consumption. Because of all this, the academy and industry are each time more interested in having a high-fidelity PHY layer in network simulators such as **ns-3**.

The ns-3 **nr** module (a.k.a., 5G-LENA) for simulation of 5G NR in sub 6 and mmWave bands [11] supports two types of beamforming methods: ideal and realistic, for analog beamforming. Ideal beamforming methods determine the beamforming vectors based on either the assumption of the perfect knowledge of the channel (e.g., cell scan method), or the exact positions of the devices (e.g., direction-of-arrival method), and they do not consume any time/frequency overhead to design the beamforming vectors [12]. On the other hand, realistic beamforming methods, are expected to select the best beamforming based on some real measurements, e.g., estimate the channel based on Sounding Reference Signals (SRSs) [13]. To support spatial multiplexing and SU-MIMO, recently a Dual-Polarized MIMO (DP-MIMO) model was developed and implemented [14]. Such implementation is limited to a particular case of SU-MIMO, supporting only 2 streams per user with no possibility for extension. Indeed, it relies on some assumptions related to the mapping of the streams to ports that are limiting its application, i.e., each stream is mapped to the antenna elements with a specific polarization. This means that there is a hard limitation on the assignment of streams to a specific transmit port and the specific antenna elements of one polarization. In addition, it includes a not realistic inter-stream interference rejection model.

Apart from 5G-LENA, MIMO has been considered in other ns-3 modules. The ns-3 `mmwave` module for simulation of 5G NR in mmWave bands [12], supports beamforming and it has been recently extended to support MU-MIMO [15]. However, the `mmwave` module has been lacking up to now SU-MIMO, which is specially relevant for sub 6 GHz bands. In the area of IEEE, various MIMO models have been proposed [16, 17, 18]. In the ns-3 `lte` module (a.k.a., LENA) there is an abstraction model to simulate 2×2 SU-MIMO, which models up to two streams per user. However, such a model does not account for wireless propagation differences across different streams, assumes no correlation between antennas, and has a simplified inter-stream interference model, which are non-realistic assumptions for an accurate PHY modeling of SU-MIMO.

In this paper, we propose, implement, and evaluate a full SU-MIMO model for system-level simulation of 5G NR MIMO systems in `ns-3`, by extending the current framework with the support of MIMO spatial multiplexing to send multiple data streams towards the same user (SU-MIMO), thus enabling the hybrid beamforming feature of 3GPP. The code is publicly available and merged into the master branch of ns-3 5G-LENA module [19]. As the analog beamforming feature has been widely analyzed in ns-3 previously, in this paper we focus on the new MIMO digital baseband processing. We have implemented codebook Type-I precoding (for single panel case), assuming MMSE-IRC (Minimum Mean-Squared Error Interference Rejection Combining) receiver for inter-stream interference suppression, as adopted in 3GPP. As compared to the old DP-MIMO implementation [14], the new full MIMO implementation supports general SU-MIMO, through the inclusion of 3GPP-compliant PMI and RI reporting. Such general MIMO includes as a particular case the old DP-MIMO, but with a more generic and realistic model for DP-MIMO in which the stream/s/port/antennas mapping is not predefined but defined through digital precoding and in which a more accurate model for the inter-stream interference calculation is considered. The current model supports up to 32 antenna ports and rank 4. However, the full MIMO model can be easily extended to support more streams/ranks/ports.

The rest of the paper is organized as follows. Section 2 reviews MIMO background, with special emphasis on the digital baseband processing and the support for MIMO in 3GPP 5G NR. Section 3 details the implementation of the full MIMO system in ns-3 5G-LENA. Then, Section 4 presents the simulation results to validate the new MIMO model in ns-3. Section 5 discusses the MIMO model’s fidelity vs computational complexity in ns-3. Finally, Section 6 includes the conclusions and future work.

2. MIMO background

In real systems, devices capable of performing MIMO spatial multiplexing can use more than one stream to transmit, e.g., a gNB in the downlink can send multiple streams to those UEs that support MIMO spatial multiplexing and can decode multiple streams simultaneously. MIMO technology has been known in textbooks for several decades, and it is used in 4G-LTE and 5G-NR and has been in Wi-Fi products for more than 20 years. For optimal MIMO performance, the gNB should apply a precoding matrix (at the transmitter) that determines how the signal is aligned relative to the channel, and the UE (at the receiver) should apply a receive filter to suppress inter-stream interference and recover each stream. Usually, the precoding matrix that provides maximum Signal-to-Interference-plus-Noise Ratio (SINR) for the given channel matrix is selected [2] and the receive filter is usually designed to reduce the mean-square error between the transmitted and decoded data symbols, for which the MMSE-IRC receiver is usually adopted in 3GPP, as it provides a good balance between performance (mean-square error reduction) and implementation complexity (linear receiver).

2.1. MIMO digital baseband processing preliminaries

Consider a MIMO downlink system composed of K gNBs, each serving a single UE on a given frequency resource (resource block (RB) in 5G NR) and time resource (time slot in 5G NR). Assume the j -th gNB ($j = 1, \dots, K$) is equipped with M_j antenna ports, and the i -th user ($i = 1, \dots, K$) has N_i antenna ports.

Let $\mathbf{H}_{i,j} \in \mathbb{C}^{N_i \times M_j}$ denote the channel matrix between the transmit antenna ports at the j -th gNB and the receive antenna ports at the i -th user ¹. Let $\mathbf{P}_i \in \mathbb{C}^{M_i \times m_i}$ denote the transmit precoding matrix that the i -th gNB uses to transmit m_i (data) streams towards its intended i -th user. To achieve multiple data stream transmission, the following constraints must be satisfied: $m_i \leq M_i$ and $m_i \leq N_i$.

Assuming narrow-band transmissions for a given frequency resource, the equivalent baseband signal $\mathbf{y}_i \in \mathbb{C}^{N_i \times 1}$ observed at the i -th user is:

$$\mathbf{y}_i = \mathbf{H}_{i,i} \mathbf{P}_i \mathbf{b}_i + \mathbf{w}_i, \quad (1)$$

where $\mathbf{b}_i \in \mathbb{C}^{m_i \times 1}$ contains the symbols transmitted by the i -th gNB towards its serving (i -th) user, and \mathbf{w}_i is the noise-plus-interference received at the i -th user, i.e.,

$$\mathbf{w}_i = \sum_{j=1, j \neq i}^K \mathbf{H}_{i,j} \mathbf{P}_j \mathbf{b}_j + \mathbf{n}_i, \quad (2)$$

where \mathbf{n}_i refers to the additive noise vector with covariance matrix \mathbf{N}_i . Assuming that interference is treated as noise and that a linear receive filter is applied at the user, the symbols are estimated at the i -th user as:

$$\hat{\mathbf{b}}_i = \mathbf{R}_i^H \mathbf{y}_i, \quad (3)$$

where $\mathbf{R}_i \in \mathbb{C}^{N_i \times m_i}$ is the linear receive filter.

The mean square error (MSE) for the symbols transmitted towards the i -th user can be expressed through the so-called MSE-matrix $\mathbf{E}_i = \mathbb{E}\{(\hat{\mathbf{b}}_i - \mathbf{b}_i)(\hat{\mathbf{b}}_i - \mathbf{b}_i)^H\} \in \mathbb{C}^{m_i \times m_i}$, being $\mathbb{E}\{\cdot\}$ the expectation operator. The receive filter that minimizes the MSE, i.e., $\text{Tr}(\mathbf{E}_i)$, being $\text{Tr}(\cdot)$ the trace operator, is the MMSE-IRC receiver [2]. The MMSE-IRC receiver is given by: $\mathbf{R}_i^{\text{mmse}} = (\mathbf{H}_{i,i} \mathbf{P}_i \mathbf{P}_i^H \mathbf{H}_{i,i}^H + \mathbf{W}_i)^{-1} \mathbf{H}_{i,i} \mathbf{P}_i$, where \mathbf{W}_i is the covariance of the noise-plus-interference received

¹Note that $\mathbf{H}_{i,j}$ is the equivalent channel between transmit and receive antenna ports and so, with this formulation, it encapsulates the wireless propagation channel between all the transmit and receive antenna elements as well as the transmit and receive analog beamforming.

at the i -th user:

$$\mathbf{W}_i = \sum_{j=1, j \neq i}^K \mathbf{H}_{i,j} \mathbf{P}_j \mathbf{P}_j^H \mathbf{H}_{i,j}^H + \mathbf{N}_i. \quad (4)$$

Note that \mathbf{W}_i admits a Cholesky “LLT” decomposition as follows: $\mathbf{W}_i = \mathbf{L}_i \mathbf{L}_i^H$. This allows transforming the received signal in Eq. (1) through the following equivalent representation: $\mathbf{y}_i^{\text{eq}} = \mathbf{L}_i^{-1} \mathbf{y}_i = \mathbf{L}_i^{-1} \mathbf{H}_{i,i} \mathbf{P}_i \mathbf{b}_i + \mathbf{L}_i^{-1} \mathbf{w}_i$, for which the second term ($\mathbf{L}_i^{-1} \mathbf{w}_i$) has an identity covariance matrix (also known as interference whitening process) [2].

Using the MMSE-IRC receiver $\mathbf{R}_i^{\text{mmse}}$, the MSE-matrix results [2]:

$$\mathbf{E}_i = (\mathbf{I} + \mathbf{P}_i^H \mathbf{H}_{i,i}^H \mathbf{W}_i^{-1} \mathbf{H}_{i,i} \mathbf{P}_i)^{-1} = (\mathbf{I} + \mathbf{P}_i^H (\mathbf{H}_{i,i}^{\text{intfNorm}})^H \mathbf{H}_{i,i}^{\text{intfNorm}} \mathbf{P}_i)^{-1}, \quad (5)$$

where $\mathbf{H}_{i,i}^{\text{intfNorm}} = \mathbf{L}_i^{-1} \mathbf{H}_{i,i}$ represents the interference-normalized channel.

This way, the SINR of the l -th stream ($l = 1, \dots, m_i$) of the i -th user is expressed as [2]:

$$\text{SINR}_{i,l} = 1/\mathbf{E}_i(l, l) - 1, \quad (6)$$

where $\mathbf{E}_i(l, l)$ denotes the l -th diagonal element of the MSE matrix \mathbf{E}_i in Eq. (5)². Let us note that thanks to the introduction of $\mathbf{H}_{i,i}^{\text{intfNorm}}$, we only need $\mathbf{H}_{i,i}^{\text{intfNorm}}$ and \mathbf{P}_i to represent the signal for the i -th user and extract the SINRs for its streams.

2.2. MIMO in 3GPP 5G NR

In 3GPP, closed-loop MIMO is enabled thanks to the CSI feedback, which includes the PMI, the RI, and the CQI [6]. The selection of the precoding matrix (i.e., \mathbf{P}_i) that gives the best performance (i.e., maximum SINR) is done by the UE, and reported in the PMI through an index of a set of predefined precoding matrices from a codebook, as part of the CSI feedback message to the gNB. In addition, the UE also reports the RI (i.e., m_i) as part of the CSI feedback,

²The MSE computation Eq. (5) has a bias [20], however, often channel decoder requires an unbiased estimate as input and for unbiased estimates the SINR values per stream is 1 less than the that of the biased estimate [20].

Table 1: Supported antenna ports configurations in 3GPP

Number of ports	(N_h, N_v) configurations
2	(1,1)
4	(2,1)
8	(2,2), (4,1)
12	(3,2), (6,1)
16	(4,2), (8,1)
24	(4,3), (6,2), (12,1)
32	(4,4), (8,2), (16,1)

which indicates to the gNB how many streams to use for that UE (i.e., m_i). Even if RI=1, we can still use precoding to combine the signals to/from multiple antenna ports in an optimal way. Each antenna port is then indeed aligned to the channel through analog beamforming. 3GPP 5G NR, different from LTE, allows sending up to 4 streams in the same transport block (TB).

In 3GPP, the MIMO operations are enabled by the adoption of antenna arrays (usually modeled as dual-polarized linear antenna arrays [21]) and the introduction of antenna ports concept, in which, for an antenna array of multiple antenna elements, multiple antenna elements are combined into one antenna port for digital processing (precoding), while analog processing (analog beamforming) is applied for the antenna elements within one antenna port. 3GPP 5G NR supports up to 32 ports, limited by the CSI-RS (CSI reference signals) design, over which the channel measurements are carried out. Antenna arrays are defined by a number of horizontal antenna elements, a number of vertical antenna elements in the physical antenna array, and whether the elements are dual-polarized or not, for the single panel case. Similarly, the number of virtual antenna ports in the horizontal and vertical directions (N_h and N_v) are defined. In the case of cross-polarized antennas, the total number of ports for a device (gNB or UE) is given by: $2 \times N_h \times N_v$. Table 1 shows the supported configurations for the number of ports in horizontal and vertical dimensions of the logical antenna array (N_h, N_v) specified by 3GPP, for the case of a single panel.

5G NR supports two types of codebooks: Type-I (with single-panel and

multi-panel support) and Type-II. In the case of Type-I single-panel (which is the focus of this paper), the precoding matrix can be expressed as: $\mathbf{P}_i = \mathbf{P}_{i,1}\mathbf{P}_{i,2}$, where $\mathbf{P}_{i,1}$ includes wideband and long-term precoding and $\mathbf{P}_{i,2}$ includes the subband and frequency-dependent precoding. $\mathbf{P}_{i,1}$ is defined by a beam or group of beams pointing in various directions, while $\mathbf{P}_{i,2}$ chooses the group of vectors from $\mathbf{P}_{i,1}$ and applies phase shifts over the polarizations.

3. New full SU-MIMO model for ns-3 and 5G-LENA

The new full SU-MIMO model that we propose in this paper and that has been adopted recently in ns-3 and 5G-LENA can combine spatial multiplexing (with up to four streams per user, and 32 antenna ports) and beamforming (which applies for each of the streams). Multiple (up to four) streams are encoded in the same TB. PMI, RI, and CQI are implemented and included as part of the CSI feedback. It follows the 3GPP codebook-based Type-I model for precoding [6] and assumes an MMSE-IRC receiver. For precoding and rank (PMI and RI) selection, an exhaustive search is implemented. The number of streams is called the rank in the code, which affects the TB size and other performance characteristics. The inter-stream interference is correctly computed through matrix processing, and this is why the use of more than 2 streams requires the Eigen library to compute operations like matrix inverse, singular value decomposition (SVD), etc. As the SINR and interference computations are correctly modeled, following [2] and as described in Section 2, and multiple streams are fit in one TB, this allows using the SISO error model for MIMO error modeling, by vectorizing the 2D SINR (of dimensions number of RBs, rank) into 1D SINR (of dimensions number of RBs \times rank, 1).

In the following, we explain the design choices and implementation details to enable the new full SU-MIMO in ns-3 and 5G-LENA. These include: 1) the extensions to ns-3 core, spectrum, and antenna modules to support the design and implementation of MIMO and the efficient operations with matrices (i.e., `MatrixArray`) (described in Sections 3.1-3.2), and 2) the extensions performed

in 5G-LENA (detailed in Sections 3.3-3.9). Extensions in **ns-3** are two-fold; there were extensions to support the definition of multiple antenna ports and the spectrum channel matrix, and also the important set of changes was to improve the computational complexity of the operations that are needed on complex 3D data structures that represent channel matrices in time and frequency. The changes in the **nr** module include 1) the removal of “the OFDMA downlink trick”, 2) the MIMO interference and SINR calculations, as well as the interfaces to pass the results to other classes, 3) the computation of transport block error rate (TBLER) based on the MIMO SINR, 4) implementation of the 3GPP-compliant Type-I precoding codebooks, 5) the optimal precoding search algorithm, which the UE needs to send as a feedback to the gNB in the PMI, 6) adding the “rank” parameter to many interfaces throughout the code, and 7) the enabling of the new methods for configuration and enabling the MIMO (i.e., enabling the MIMO feedback). All these extensions to the 5G-LENA module have been included in the 5G-LENA mainline in its Release 3.0, except for the 32-port codebook, which is to be released shortly in the 5G-LENA 3.1 Release [19].

3.1. Introducing *MatrixArray* into *ns-3*

The 3GPP channel matrix in **ns-3** implements a 3D structure whose dimensions depend on the number of the transmit antenna, the receive antenna, and the number of clusters. Such a structure models the channel in a time domain, i.e., there is a channel matrix per cluster. To extend the 3GPP channel model to support full MIMO, we need to consider the frequency domain channel matrix, having three dimensions: the number of receive ports, the number of transmit ports, and the number of RBs, i.e., as mentioned in Section 2.1, $\mathbf{H}_{i,j} \in \mathbb{C}^{N_i \times M_j}$, i.e., there is a $\mathbf{H}_{i,j}$ per RB. Hence, when moving to MIMO design there is a shift of the heavy computations from time to frequency domain. While in the previous **ns-3** 3GPP channel model implementation the third dimension is relatively small, e.g., the number of clusters is between 10 and 20, the third dimension of the frequency domain channel matrix can be much larger, e.g. 275 RBs.

During the development of MIMO models in ns-3, we found out that this third dimension can affect significantly the computational performance when using the 3D data structures that were available in ns-3 before our contribution, i.e., to represent such 3D structures ns-3 and its spectrum module used a C++ `std` vector of a vector of a vector. Pagan et al. in [22] has improved the efficiency of the 3GPP channel model in ns-3 by allowing the usage of Eigen [23] to perform computations on 3D structures. Thanks to that upgrade when Eigen is available the 3GPP channel is represented as `std::vector` of Eigen matrices. This improved the performance of the 3GPP channel model significantly. However, we found out that this improvement was not enough for MIMO computations. We observed that using such vector of matrices (regardless they are C++ or Eigen) becomes inefficient when the third dimension increases which is the case with the frequency domain matrices needed for MIMO models. For this reason, we have investigated and proposed to use a new data type in ns-3 that can provide more efficient storage, access, and linear algebra operations for 3D structures, i.e., arrays of matrices, which are typically used either in the current SISO or future MIMO models. In this respect, we proposed a new class called `ValArray` that leverages `std::valarray` to represent 3D structures. We extended this interface, by an additional class called `MatrixArray`, which treats this 3D array, as an array of matrices, on which linear algebra operations can be performed page by page (matrix by matrix). We created tests for `ValArray` and `MatrixArray` to test all the implemented functionalities. Apart from proposing a more efficient representation of 3D structures for SISO and MIMO, `MatrixArray` has improved significantly the interfaces that are using Eigen because a single class is a unique wrapper for both implementations, the one using Eigen, and one that is not. This is a much better approach than having two independent definitions of the 3D structures, as it was implemented in ns-3 before our changes. The proposed `MatrixArray` approach has passed the ns-3 review process and it was included in the official ns-3 since ns-3.38 Release [24].

We performed profiling to demonstrate that the proposed approach is notably more efficient than other approaches when the third dimension increases.

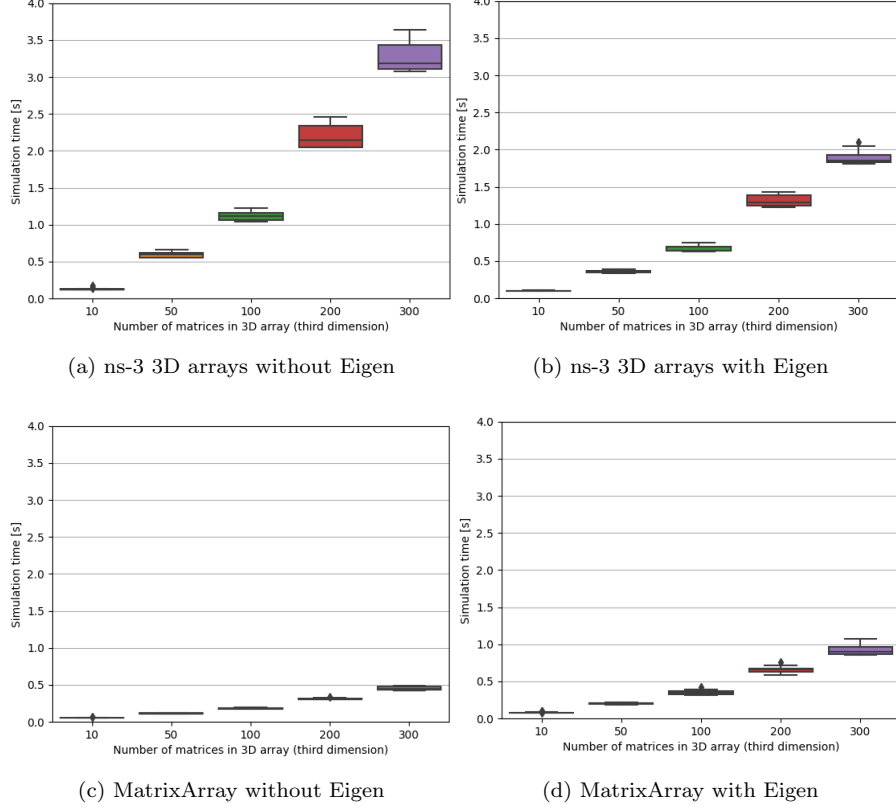


Figure 2: Comparison of the computational complexity of the ns-3 3D array data structure based on the `std::vector` of matrices and the proposed `MatrixArray`-based approach

To perform profiling we created an example that performs $1e5$ multiplications of two 3D arrays and assigns the value to the third 3D array. We test different values for the third dimension (10, 50, 100, 200, 300). Matrices in the 3D array are of 2x2 dimensions. Profiling is carried out by using the optimized mode. In Figure 2 are shown the benchmark results. We observe that depending on the setup the proposed `MatrixArray`-based approach is performing equally well or better. The major improvement comes from the usage of underlying `std::valarray` to represent the 3D structure instead of the usage of `std::vector` of matrices. The usage of C++ `std::valarray` instead of the C++ `std::vector` of the vector of vectors allows for lower overheads when accessing the values, i.e., single direct

access versus the access operation composed of 3 redirections. Also, the usage of a single `std::valarray` guarantees that all the data belonging to the same matrix is in the contiguous memory allocations, which significantly reduces the overheads of loading and caching, i.e., the efficiency of each load of the fragment of memory increases concerning the C++ std vector-based 3D structures, where we load parts of memory that are not necessary for the current matrix operation. Notice that `MatrixArray`-based approach when using Eigen leverages the `Eigen::Map` function which uses directly the piece of memory where the specific matrix is placed, which allows us to use the unique `MatrixArray` interface for both cases, with and without Eigen. We do, however, notice that there is an overhead of using Eigen when the matrices are of small dimensions, such as 2x2, as in this profiling example.

3.2. Extending ns-3 antenna and spectrum modules to support MIMO

In the ns-3 `antenna` module is added support for dual-polarized antenna arrays and for multiple ports in the antennas, which are a prerequisite to support a full MIMO model implementation. `UniformPlannarArray` class that models the uniform planner arrays is extended to support the concept of antenna ports following the sub-array partition model for TXRU virtualization, as described in Section 5.2.2 of 3GPP TR 36.897 [25]. In case the antenna array is dual polarized, the total number of antenna elements is doubled and the two polarizations are overlapped in space.

On the other hand, the 3GPP channel model in the ns-3 `spectrum` module is extended to support multiple ports, dual-polarized antenna elements, and the generation of the frequency domain spectrum channel matrix. In MIMO systems multiple transmit and receive antenna ports can exist, hence the power spectral density (PSD) is multidimensional. It was not anymore enough to represent a signal as before with a single PSD, instead, the frequency domain channel matrix is introduced to represent the channel. The dimensions of such a frequency domain channel matrix are the number of receive antenna ports, the number of transmit antenna ports, and the number of subbands in frequency (or RBs). Ad-

ditionally, the 3D precoding matrix is introduced to be able to correctly perform the MIMO calculations of the received signal and interference. The dimensions of the precoding matrix are the number of transmit antenna ports, the number of transmit streams, and the number of subbands (or RBs). To achieve this, `CalcRxPowerSpectralDensity` is not returning a single `Ptr<SpectrumValue>` as before but is returning `Ptr<SpectrumSignalParameters>` which is extended to include the frequency domain spectrum channel matrix and the precoding matrix. Finally, ns-3 `ThreeGppChannelModel` is extended to support the calculation of the MIMO channel matrix by taking into account multiple ports and/or dual-polarization. When multiple ports are being configured the subarray partition model is adopted for TXRU virtualization [25], and so equal beam weights are used for all the ports. Support of the full-connection model for TXRU virtualization would need extensions. To preserve backward compatibility, SISO 3GPP channel calculations are still performed as before these changes for MIMO. To achieve this, i.e., to compute a single received PSD, the frequency domain 3D spectrum channel matrix is used. In the case of multiple ports at the transmitter, the received PSD is calculated by summing per each RB the real parts of the diagonal elements of the $(\mathbf{H}_{i,i}\mathbf{P}_i)^H * (\mathbf{H}_{i,i}\mathbf{P}_i)$, where $\mathbf{H}_{i,i}$ is the frequency domain spectrum channel matrix and \mathbf{P}_i is the precoding matrix (see Section 2.1). To test these new MIMO extensions in the antenna and spectrum module, we developed two new additional tests:

- `ThreeGppCalcLongTermMultiPortTest` which tests that the channel matrices are correctly generated when multiple transmit and receive antenna ports are used, and
- `ThreeGppMimoPolarizationTest` which tests that the channel matrices are correctly generated when dual-polarized antennas are being used.

These extensions to support full MIMO have been included in the ns-3 mainline in the ns-3.41 Release [24].

3.3. Removal of the “OFDMA downlink trick”

The module was initially designed with TDMA as the primary scheduling mechanism for downlink transmissions from a single gNB (base station) to multiple UEs (user equipment, e.g., a mobile phone) within a single time slot. However, when OFDMA was introduced, the model employed a technique called “OFDMA downlink trick” to combine multiple transmissions into a single one. This approach posed challenges for a seamless MIMO design in the nr module. Consequently, eliminating the “OFDMA downlink trick” became the main prerequisite for a clean MIMO design and implementation.

In TDMA, the 5G-LENA model is straightforward. Packets destined for a specific UE are combined into a single packet burst, creating an independent representation for each signal to model the transmission. This maintains a one-to-one relationship between the data packets sent to each UE (packet burst), the downlink allocation specified in Downlink Control Information (DCI), which contains metadata about the transmission, and the physical layer signal transmitted from the antenna. This relationship is ensured by considering the starting symbol (symStart) as an additional parameter when mapping packets to packet bursts.

However, in OFDMA, the “OFDMA downlink trick” disrupts this one-to-one relationship. Now, there is only one packet burst containing all packets for all UEs, as symStart is the same for all UEs. Because of this, only the downlink allocation with metadata for the first transmission (e.g., DCI for UE 1) is used, with additional logic implemented to skip the other DCIs. Also, the physical layer signal is created as a combination of all allocations. OFDMA downlink trick poses challenges for MIMO design, as gNB may transmit towards various UEs using different ranks. In such cases, the OFDMA downlink trick requires merging the representations of the different MIMO signals, which is difficult to implement and error-prone. For this reason, the OFDMA downlink trick technique is removed. To implement this removal, RNTI is introduced as an additional parameter in the packet burst mapping. This modification establishes a one-to-one correspondence between the downlink allocation or DCI towards

each UE and the respective packet burst.

We removed the logic that was skipping all the downlink allocation transmissions other than the first, i.e., after the removal of the trick, all downlink locations are transmitted in independent signal representations. To avoid incorrect modeling of analog beamforming, we added logic to ensure that the analog beamforming vector is changed only once per transmission event. We removed the combining of all allocations into a single physical signal. We kept parts of the existing combining algorithm to ensure the correct scaling of the transmit PSD.

Notice that the removal of the “OFDMA downlink trick” is a step forward towards the implementation of the full OFDMA in 5G-LENA without having the current limitation of scheduling at the same time only the UEs belonging to the same beam. The current implementation schedules UEs with the same analog beam, which applies to all ports (and so all streams) of each UE, but different digital beams can be used through digital processing and the precoding matrix \mathbf{P}_i , which is designed per UE and can vary across RBs. Since now, after the removal of this trick, the signal to each UE is independently transmitted, for each of them, and gNB can use a different digital beam on each RB. For this full OFDMA to work correctly, the OFDMA schedulers in 5G-LENA should be updated to not limit the scheduling at the same time of the UE’s belonging to the same analog beam.

3.4. *The MIMO interference and SINR calculations*

The `nr` model for the interference calculation is extended to support the calculation of the MIMO interference and MIMO SINR calculations. The main class for the calculation of the interference in the NR module is `NrInterference` class. This class is extended with new functions for the computation of the interference-and-noise covariance matrix and SINR. These functions are `CalcOutOfCellInterfCov`, `CalcCurrInterfCov`, `AddInterference`, and `ComputeSinr`. `CalcOutOfCellInterfCov` computes the interference signals from all out-of-cell interferers. `CalcCurrInterfCov` prepares `NrInterference` class for MU-MIMO by supporting the calculation of

the interference signals by considering the interferers from the same cell. For example, in the uplink MU-MIMO, UEs from the same cell could act as interferers. **AddInterference** adds the covariance of the signal to an existing covariance matrix. Finally, **ComputeSinr** computes the SINR as follows:

1. the interference-plus-noise covariance matrix of the received signal is computed (i.e., \mathbf{W}_i in Eq. (4)),
2. the received signal is transformed into an equivalent representation where the interference-plus-noise covariance is an identity matrix (a.k.a., interference whitening transformation, see Section 2.1) and $\mathbf{H}_{i,i}^{\text{intfNorm}} = \mathbf{L}_i^{-1} \mathbf{H}_{i,i}$ is computed,
3. a dummy precoding matrix \mathbf{P}_i is created when none exists, and
4. the SINR of each stream based on the MSE matrix is computed as detailed in Eq. (5), by using $\mathbf{H}_{i,i}^{\text{intfNorm}}$ and \mathbf{P}_i .

To support all these MIMO operations, it was not enough to use a single dimensional **SpectrumValue** type that has been traditionally used in **NrInterference** for SISO. To support efficient storage and computations of MIMO operations new classes were defined, such as **NrCovMat**, **NrIntfNormChanMat** and **NrSinrMatrix**. **NrCovMat** stores the interference-plus-noise covariance matrices of a MIMO signal, with one matrix page for each frequency bin. This class also provides some functions for efficient computations on covariance matrices. Its function **CalcIntfNormChannel** performs interference whitening (see Section 2.1). **NrIntfNormChanMat** stores the interference-whitened channel matrix, the channel matrix after normalizing/whitening the interference, i.e., $\mathbf{H}_{i,i}^{\text{intfNorm}} = \mathbf{L}_i^{-1} \mathbf{H}_{i,i}$. Its function **ComputeSinrForPrecoding** computes the SINR based on MSE, see Eq. (6). Finally, **NrSinrMatrix** stores the MIMO SINR matrix whose dimensions are the rank and the number of RBs.

The MIMO interference and SINR calculations require Eigen3 [23], a C++ template library for linear algebra: matrices, vectors, numerical solvers, and related algorithms. However, the Eigen library is not always available. To allow the compilation even when Eigen is not available a CMake switch is added:

a) when Eigen is enabled, the file `nr-mimo-matrices-eigen.cc` is compiled
 b) when Eigen is disabled, the file `nr-mimo-matrices-no-eigen.cc` is compiled
 (the implementations just contain a single `NS_FATAL_ERROR`). In this case, users can still compile but can only use SISO. The functions used from the Eigen library could be in the future implemented in ns-3 to reduce the dependency of the nr module on the Eigen library. Then `nr-mimo-matrices-no-eigen.cc` could be implemented to call these ns-3 alternatives of Eigen functions.

To support the multi-dimensional MIMO signals a new interference chunk processor called `NrMimoChunkProcessor` is introduced. This class mirrors the original `LteChunkProcessor` that is originally used in `NrInterference` for SISO. `LteChunkProcessor` is not sufficient for MIMO because it can only store a frequency-domain vector of SINR values whereas MIMO requires a 2D matrix with the dimensions: number of RBs and number of MIMO streams. `LteChunkProcessor` stores the sum of the different signals' power spectral density values and performs the averaging once the function `End` is called. Such SINR averaging in the time domain limits the fidelity. In general, each received signal may have a different number of MIMO layers, hence combining the SINR of different signals is not trivial. To avoid all this, `NrMimoChunkProcessor` keeps a list with full information of all different signals, and no averaging is performed at this stage. The averaging now must be implemented later in the error model which opens the door also for different possible implementations, e.g., the error model may apply exponential effective SINR both over time and frequency. Hence `NrMimoChunkProcessor` only looks like `LteChunkProcessor` but is mainly used as a storage to pass the information to other entities that can perform different computations by exploiting the full information of all different signals. `NrMimoChunkProcessor` provides two kinds of callbacks:

- Providing MIMO SINR: one 2D matrix for each different time-domain chunk, which is used by the error model to compute TBLE, and
- Providing the interference covariance matrices for each different time-domain chunk are passed to CQI feedback function and are used together

with the channel matrix to compute the precoding matrix PMI feedback.

3.5. Computation of TBLE based on the MIMO SINR

A new function called `GetTbDecodificationStatsMimo` is added to `NrErrorModel` to determine if a transport block was received successfully. `GetTbDecodificationStatsMimo` performs a simple weighted average over potentially multiple different signal values received over time to get a single SINR matrix. The SINR matrix is then linearized to a vector and passed to the existing error model for SISO by calling a function `GetTbDecodificationStats`. Effectively, `GetTbDecodificationStatsMimo` is a translation layer between the new MIMO code and the existing SISO error model.

When using MIMO one should configure the `AmcModel` as `ErrorModel`. The `ShannonModel` is not yet supported with MIMO, some additions are needed to `NrAmc` to allow its usage.

3.6. Implementation of the 3GPP-compliant Type-I precoding codebooks

The main base class for the implementations of Type-I codebooks is called `NrCbTypeOne`. This is a wrapper class for the implementations of Type-I precoding matrices in 3GPP TS 38.214. The main functions that `NrCbTypeOne` offers are: `Init`, `GetNumI1`, `GetNumI2` and `GetBasePrecMat(i1,i2)`. `Init` is used to initialize the codebook parameters based on the configuration through the attribute values. `GetI1` returns the number of wideband precoding indices `i1`. Similarly, `GetI2` returns the number of subband precoding indices `i2`. Finally, `GetBasePrecMat(i1,i2)` returns the 2D precoding matrix of size `gNB` ports x rank. While `NrCbTypeOne` represents only the interface, we provide two 3GPP-compliant specializations of this class: `NrCbTwoPort` and `NrCbTypeOneSp`.

`NrCbTwoPort` provides the implementation of the two-port codebook as per 3GPP TS 38.214. This class implements a codebook for a `gNB` with at most 2 antenna ports. For a single port, it returns a single-element matrix with a value of 1.0. For two ports, it implements Table 5.2.2.2.1-1 [6]: Codebooks for 1-layer and 2-layer CSI reporting using antenna ports 3000 to 3001.

NrCbTypeOneSp provides the implementation of Type-I Single-Panel Codebook 3GPP TS 38.214 (Rel. 15, Sec. 5.2.2.2.1 [6]) that supports up to 32 antenna ports. It supports codebook mode 1 only and the rank is limited to 4. Codebook mode 1 means the per-subband i2 beam index is used only for the phase shift of the second polarization, while codebook mode 2 would use i2 also for beam refinement. **NrCbTypeOneSp** extends the **NrCbTypeOne** interface significantly by adding many new functions for the initialization of different parameters, such as i11, i12, i13, k1-k2 for different ranks, and the initialization of the full precoding matrix \mathbf{P}_i . **CreateUniqueBfvs** is implemented to create a list of different (digital) beamforming vectors used for the first polarization. When the number of ports is 2, this function returns only a single element of value 1.0. Otherwise, it returns:

- for rank 1, the vector $v_{l,m}$ (TS 38.214 Table 5.2.2.2.1-5 [6]),
- for rank 2, the two vectors $v_{l,m}$ and $v_{l',m'}$ (TS 38.214 Table 5.2.2.2.1-6 [6]),
- for rank 3 or 4 with less than 16 antenna ports, the two vectors $v_{l,m}$ and $v_{l',m'}$ (these are the unique vectors in the upper parts of TS 38.214 Tables 5.2.2.2.1-7 and 5.2.2.2.1-8 [6]), and finally,
- for rank 3 or 4 with at least 16 antenna ports, the two concatenated vectors $[\tilde{v}_{l,m}; \theta_p \times \tilde{v}_{l,m}]$ and $[\tilde{v}_{l,m}; -\theta_p \times \tilde{v}_{l,m}]$.

Additional functions are added to support the codebook implementation, such as, among others, **Kroneckerproduct** to create a Kronecker product of two vectors, e.g., used when creating the vector $v_{l,m}$ and $\tilde{v}_{l,m}$. Also, **GetNumI11**, **GetNumI12**, and **GetNumI13** are added to provide the number of i11, i12, and i13 indices which represent, respectively, the horizontal and the vertical beam directions, and co-phasing shifts for a secondary beam. Notice that $i1$ is defined as a vector in [6]. To reduce the number of loops and parameters, this vector is mapped to a unique integer. The mapping that we propose is as follows: $i1$ is created from i11, i12, i13 as $i1 = i11 + numI11 * (i12 + numI12 * i13)$, where $numI11$ is the number of i11 values (horizontal beam indices) and $numI12$ is

the number of `i12` values (vertical beam indices). This mapping is an arbitrary choice, and the ordering of this mapping does not impact a full search over all indices. The functions, `MapToI11(i1)`, `MapToI12(i1)`, and `MapToI13(i1)` are created to map the composite `i1` index to an `i11` (horizontal beam direction), `i12` (vertical beam direction) and `i13` (co-phasing of a secondary beam) indexes, respectively. When the function `GetBasePrecMat(i1,i2)` is called, then the indices `i11`, `i12` and `i13` are obtained by using these mapping functions and `GetBasePrecMatFromIndex(i11,i12,i13,i2)` function is called that returns the precoding matrix for such indices. Additional Type-I codebooks, e.g., codebook mode 2, could be implemented by defining a new specialization of `NrCbTypeOne` codebook.

3.7. Search for the optimal precoding matrix

`NrPmSearchFull` class is implemented to find the optimal precoding matrix, rank indicator, and corresponding CQI, and creates a CQI/PMI/RI feedback message. `NrPmSearchFull` uses exhaustive search for 3GPP Type-I codebooks. Optimal rank is considered as the rank that maximizes the achievable TB size when using the optimal PMI. To determine the rank indicator the algorithm loops through all ranks (the number of MIMO streams), and for each rank it computes PMI, and it computes the maximum supported MCS, and associated TB size, and finally, it selects the rank and PMI that result in the highest TB size. The optimal wideband (WB) and subband (SB) PMI values are periodically updated based on the configured update intervals, which can be configured using two attributes in the `NrUePhy` class: `WbPmiUpdateInterval` and `SbPmiUpdateInterval`.

When a PMI update is requested, the optimal precoding matrices are updated using an exhaustive search over all possible precoding matrices specified in a codebook compatible with 3GPP TS 38.214 Type-I. The exhaustive search loops over all possible subband precoding matrices computes the SINR that would be achieved by each precoding matrix and selects the precoder resulting in the highest average SINR. The full search algorithm is illustrated in Algo-

Algorithm 1 Exhaustive PM search

input: Interference-normalized channel matrix

```
for < Each rank  $\in R$  > do
  for < Each  $i1 \in I1$  > do
    for < Each  $sb \in S$  > do
      for < Each  $i2 \in I2$  > do
         $C_{sb}(sb, i1, i2, rank) \leftarrow \text{ComputeCapacityForPrecoders}(H(sb) \times$ 
 $P(i1, i2, rank))$ 
        if  $C(sb, i1, i2, rank) > C_{sb}(sb, i1, \text{bestI2}(sb), rank)$  then
           $\text{bestI2}(sb) \leftarrow i2$ 
        end if
      end for
       $C_{wb}(i1, rank) = C_{wb} + C_{sb}(sb, i1, \text{bestI2}(sb), rank)$ 
    end for
    if  $C_{wb}(i1, rank) > C_{wb}(\text{bestI1}, rank)$  then
       $\text{bestI1}(rank) \leftarrow i1$ 
    end if
  end for
  if  $C_{wb}(\text{bestI1}(rank), rank) > C_{wb}(\text{bestI1}(rank), \text{bestRank})$  then
     $\text{bestRank} \leftarrow rank$ 
  end if
end for
output: bestRank, bestI1, bestI2 (per each subband)
```

rithm 1. Finally, the feedback message is created that includes the optimal rank, the corresponding optimal PMI, and CQI.

`NrPmSearch` is the base class and `NrPmSearchFull` is one possible specialization that finds PMI, RI, and CQI by using the described exhaustive search. One could create another specialization of `NrPmSearch` that would implement a different algorithm to find PMI and RI values.

3.8. Rank parameter

Many interfaces are extended in 5G-LENA to allow passing the rank number (the number of MIMO streams). In this way, for example, already existing functions for the calculation of the transport block size for SISO could be easily updated to be used for both, SISO and MIMO (in this section we refer to SISO as the single stream transmission, although multiple antennas are supported, and MIMO for the multiple-stream transmission). Also, packet traces are extended to include the rank number.

3.9. MIMO activation

`NrHelper` class is responsible for setting the `NrPmSearch` algorithm to `NrUePhy` instance, and the configuration of the corresponding parameters, such as the type of the search algorithm, the type of the codebook, and the rank limit. `NrHelper` also creates `NrMimoChunkProcessor` and adds the necessary callbacks. These callbacks are:

- `NrSpectrumPhy::UpdateMimoSinrPerceived` which is called to provide MIMO SINR feedback
- `NrUePhy::GenerateDlCqiReportMimo` which is called to provide MIMO signal to functions that perform PMI search and create CQI/PMI/RI feedback

To enable MIMO in the simulation, the `EnableMimoFeedback` attribute of the `NrHelper` should be set to true. To configure PMI search parameters (such as rank limit, PMI search method, and the codebook) `NrHelper` provides a function `SetupMimoPmi`. The `EnableMimoFeedback` enables MIMO feedback including PMI/RI/CQI, while `RankLimit` limits the possible RI value (e.g., to 1 stream). So, even if RI is limited to 1, the usage of MIMO feedback can provide benefits because of the PMI feedback.

4. Evaluation and analysis

In this section, we evaluate and confirm the correct behavior of the newly developed full SU-MIMO model, through multiple simulation campaigns over various antenna array configurations.

4.1. Evaluation scenario

To evaluate the MIMO implementation we created a simple topology consisting of one gNB and one UE ³. The UE receives the downlink constant bit rate (CBR) traffic from the remote host. The default scenario parameters, such as the central frequency, bandwidth, gNB transmit power and down-tilt angle, and gNB and UE antenna heights, are set up according to a typical 3GPP urban macro dense scenario, such as given in Table 10 in [27]. In Table 2, we list all the simulation parameters, except for the specific number of gNB and UE antenna array elements, antenna ports, and polarizations that depend on the specific MIMO configuration being evaluated, which we provide in Table 3 and will specify in the following sub-sections which specific configuration is used in each simulation campaign. The antenna port configurations used in simulation campaigns are according to those supported by 3GPP, as shown in Table 1. We define 5 different simulation campaigns to evaluate different aspects of MIMO implementation. These simulation campaigns are created to analyze:

1. MIMO implementation using a single-panel 2-port codebook, in which we analyze the impact of using the precoding matrix versus a base-line SISO antenna and 2-stream MIMO in Section 4.2,
2. MIMO implementation using a single-panel 32-port codebook, in which we analyze how different gNB antenna configurations result in different rank selections and achieved performance (Section 4.3),
3. the impact of gNB antenna ports configuration when using single-panel 32-port codebook (Section 4.4),

³An additional pair of gNB and UE can be enabled to simulate the impact on the interference. Such scenario configuration will be used in Section 4.6.

Table 2: Scenario configuration parameters

Parameter name	Parameter Value
ns-3 simulator release	Release 41
5G-LENA NR module	Release v3.0 [26]
Independent random simulation runs	100 repetitions per setup (ns-3 RNG values: 1-100)
Simulation time	1 seconds
CBR traffic inter-packet interval time	$40\mu s$
CBR traffic packet size	1000 bytes
Channel model	3GPP SCM TR 38.901 UMa
Channel update period	100 ms
LoS/NLoS condition model	3GPP TR 38.901 UMa
LoS/NLoS update period	100 ms
Central frequency	4 GHz
Bandwidth	10 MHz
gNB transmit power	41 dBm
gNB and UE antennas	UPA 3GPP [21]
MIMO codebook	Two-port and 32-port codebook [6]
WB and SB update intervals	10 ms and 2ms
gNB and UE number of panels	Single-Panel
gNB and UE antenna spacing	$d_h = 0.5\lambda$, $d_v = 0.5\lambda$
gNB and UE bearing angles	0 and 180 degrees
gNB and UE polarization slant angles	0 and 90 degrees
gNB and UE antenna height	25 and 1.5 meters
gNB and UE noise figure	5 dB and 7 dB
gNB and UE antenna gain	8 dBi (max) and 0 dBi
Numerology	0 (15 kHz subcarrier spacing)
Duplexing mode	TDD
Multiple access scheme	TDMA
TDD subframe pattern	Flexible (F F F F F F F F)
TDD slot format	19 (DL F F F F F F F F F UL)
MCS Table	Table 2, TS 38.214 Table 5.1.3.1-2
HARQ combining method	Incremental redundancy
Max no. of HARQ processes	20
Max no. of HARQ retx	3
UE processing delay (K1)	2 slots
RLC mode	Unacknowledged mode

4. the impact of SB and WB PMI interval update on the achieved throughput and the simulation execution time when using a single-panel 32-port codebook (Section 4.5), and

Table 3: Antenna configuration parameters for different simulation campaigns

Antenna configuration	Device	N_h	N_v	Rows	Cols	X-polarized
Conf 1	gNB	2	1	2	4	No
Conf 1	UE	2	1	2	2	No
Conf 2a	gNB	2	2	4	2	Yes
Conf 2b	gNB	4	2	8	4	Yes
Conf 2	UE	2	1	1	2	Yes
Conf 3 with 8 ports	gNB	2	2	8	4	Yes
Conf 3 with 16 ports	gNB	4	2	8	4	Yes
Conf 3 with 32 ports	gNB	4	4	8	4	Yes

5. MIMO implementation when an interfering gNB and UE are present in the scenario (Section 4.6).

In each simulation campaign, we vary the distance between gNB and UE, i.e., by moving the UE away from gNB in the x-axis direction. We collect the measurements of the end-to-end application layer throughput, delay, jitter, selected MCS, rank, and execution time. We are interested not only to understand the performance improvements but also to understand the impact of the MIMO implementation on the computational complexity of the simulations. Finally, for each specific scenario configuration (e.g., antenna configuration, distance between gNB and UE, etc.), we run 100 independent random repetitions to produce statistically confident results.

4.2. Single-panel two-port codebook

The single-panel two-port codebook simulation campaign covers three different cases: 1) when MIMO feedback is not enabled (“No MIMO FB” in figures), 2) when MIMO feedback is enabled, but the maximum rank is limited to 1 (“MIMO FB - maxRi=1” in figures), and 3) when MIMO feedback is enabled, and the maximum rank is not limited, i.e., when the maximum rank is set to 2 for two-port antenna, it is not limiting the rank because the maximum rank in such case is 2 (“MIMO FB - maxRi=2” in figures). In this simulation campaign, the gNB and UE antennas are configured as Conf 1 in Table 3. In Figure 3, we

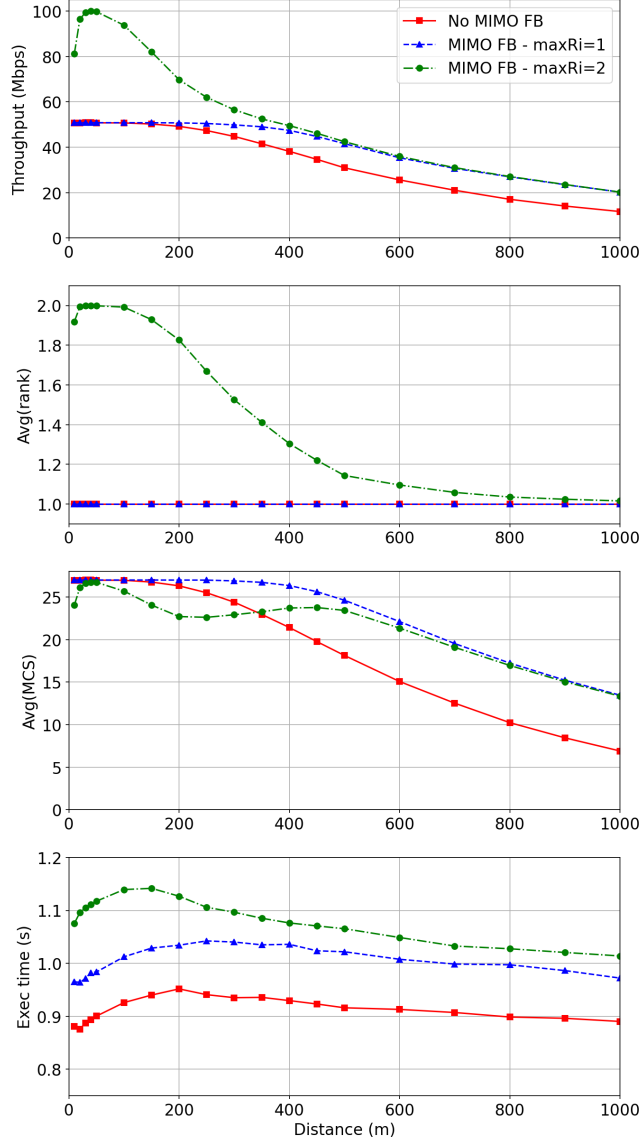


Figure 3: Two-ports codebook (Table 5.2.2.2.1-1)

show the results of the simulation campaign when using a two-port codebook as defined in TS 38.214 in Table 5.2.2.2.1-1 [6].

We observe that the MIMO feedback helps achieve better throughput compared to the case when it is disabled. This is thanks to the digital precoding

that results in a higher SINR and hence more often is selected higher MCS, than in the case without MIMO feedback. This results in an increased throughput for distances larger than 200 meters. As expected, MIMO with up to 2 streams results in the highest performance, but only for distances up to around 400 meters, where rank 2 is often selected. It can be seen that this happens until around 400 meters, and while the average RI is around 1.3. After, 400 meters it is less and less likely that the rank selected will be 2, but the MCS slightly starts to increase as the distance increases as more often a single stream is selected. We also see a soft drop in the selection of the MCS when the average rank drops below 1.8, which happens because the MCS for two ranks starts to drop, but since still is sometimes being selected rank 1, the MCS for rank 1 is much higher, which results in average MCS that seems quite stagnant from 200 to 400 meters. To summarize, this is because while the rank is dropping at the same time is being more and more selected rank 1 which on average has a higher MCS than the case when rank 2 is selected.

Finally, Figure 3, also shows the execution time, and as expected, when MIMO feedback is enabled we have a more computationally complex simulation because of the exhaustive PM search explained in Section 3.7. Also, when the rank is limited to 1 since the PM search space reduces, the execution time is lower than in the case of $\text{maxRi} = 2$.

4.3. Single-panel 32-port codebook

The single panel 32-port codebook simulation campaign tests the correct behavior of the 32-port codebook implemented according to 3GPP TS 38.214 [6]. We configure two different simulation campaigns for two different configurations of antenna elements and ports at gNB:

- 16 AEs and 8 ports at gNB: the antenna elements at gNB and UE are configured by using the configuration Conf 2a and Conf 2, respectively, from Table 3.
- 64 AEs and 16 ports at gNB: the antenna elements at gNB and UE are

configured by using the configuration Conf 2b and Conf 2, respectively, from Table 3.

In this simulation campaign, we have 5 different cases: MIMO feedback is not enabled (“No MIMO FB” in figures), and the 4 cases when MIMO feedback is enabled and the rank limit is set to 1, 2, 3, and 4 (i.e., “MIMO FB - maxRi=1,2,3,4” in figures). When the rank limit is set to 4, the rank is not limited, because the maximum rank per UE is 4, since as per 3GPP, a single transport block supports the maximum of 4 streams.

In Figures 4 and 5 we illustrate the performance for these two configurations of the gNB antenna, Conf 2a and Conf 2b, respectively. We observe that in both configurations of the gNB antenna, the SINR is sufficiently large to support rank 2 for almost all simulated distances. The reason for this is also that in this scenario there is no interference, so with large antenna arrays and multiple ports, the signal quality improves significantly. We observe that for lower distances when the rank limit is set to 4, the UE achieves a throughput of around 190 Mbps. The maximum throughput that can be achieved is 200 Mbps, which is 4 times of SISO case which supports at most 50 Mbps, as MCS Table 2 from TS 38.214 [6] is configured. We also observe when increasing the distance how different curves are joining. First maxRi=4 join maxRi=3, at around 200 meters in Conf 2a and around 300 meters in Conf 2b. Then they both join the curve for maxRi=2 at around 300 meters in Conf 2a and around 500 meters in Conf 2b. Finally, these 3 eventually join the maxRi=1 curve at some larger distance than 800 meters. We see a similar but reverse effect when looking at the average MCS, i.e., the maxRi=4 has the lowest average MCS since it has to split its power into 4 streams, hence its SINR is lower, then this curve joins with the maxRi=3, then later with maxRi=2 and finally with maxRi=1. We observe also a stagnant trend of the average MCS when maxRi is 2, 3, and 4. This is because while the selected rank drops with distance, the average MCS per rank being used increases, which on average gives a similar value for the average MCS for a large range of distances, e.g., for Conf 2a, even if the average

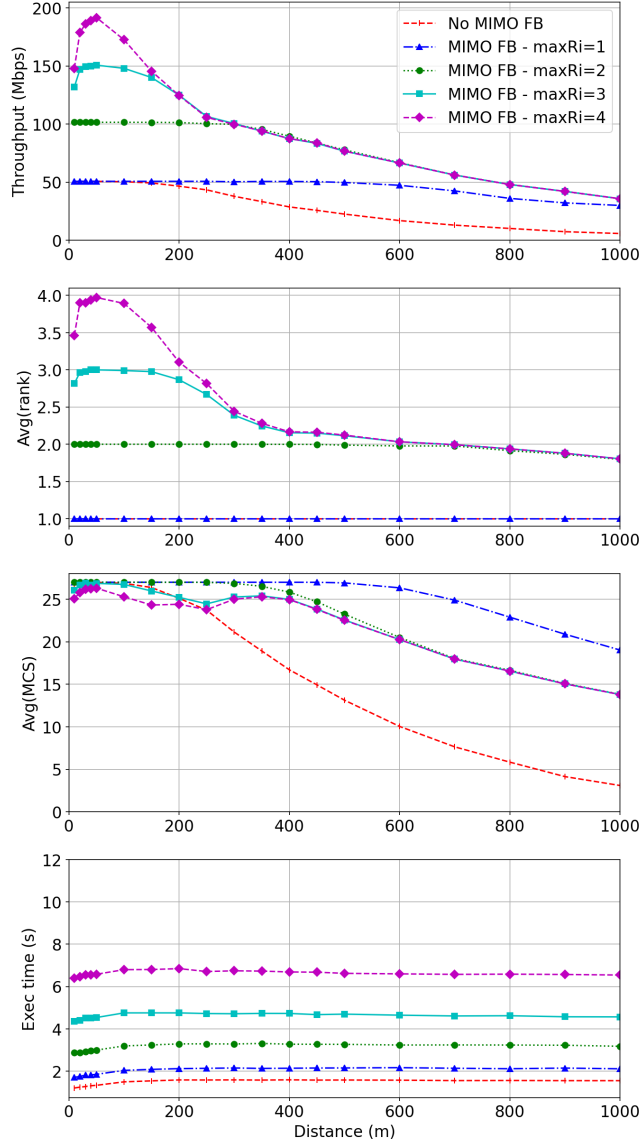


Figure 4: Single-Panel Codebook 3GPP TS 38.214, 16 AEs, 8 ports

rank constantly drops until around 400 meters, we can observe that the MCS is almost constant until the same distance.

Regarding the computational complexity, we can observe that the complexity increases with the number of antenna elements and the number of ports.

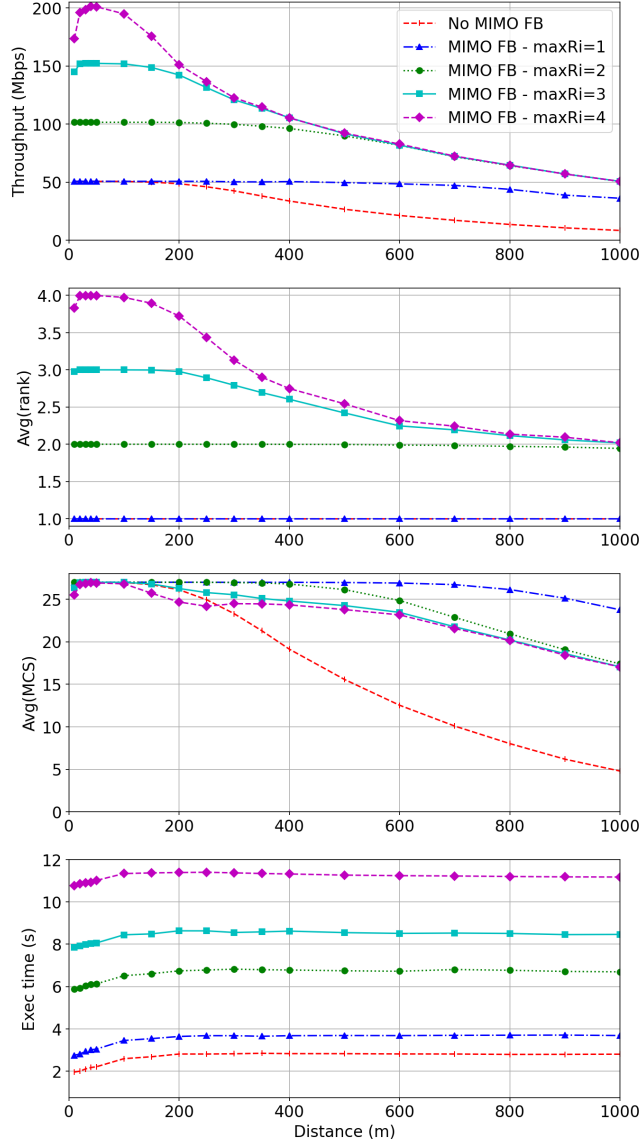


Figure 5: Single-Panel Codebook 3GPP TS 38.214, 64 AEs, 16 ports

The large number of antenna elements increases the complexity of the operations with channel matrices. On the other hand, the increased number of ports increases the complexity of the calculation of the spectrum channel matrices, MIMO interference and the PM search. While in Conf 1 the execution time was

around 1 second with $\text{maxRi}=2$, we observe for Conf 2a and Conf 2b that the execution time for $\text{maxRi}=2$ has increased 3 and 6 times, respectively. Then further time increases in Conf 2a and 2b increase linearly with maxRi .

4.4. Impact of antenna ports configuration

In this section, we analyze the impact of the different port configurations, mainly on the throughput and execution performance, when using an antenna array of a fixed number of dual-polarized antenna elements. Here, it is important to remember that the UE moves from gNB in the x-axis direction and that the bearing angles as shown in Table 2 are configured so that gNB and UE antenna arrays are pointing one to another in the xz plane. We configure 3 different cases of gNB antenna ports:

1. 8 ports, $N_h = 2, N_v = 2$
2. 16 ports, $N_h = 4, N_v = 2$, and
3. 32 ports, $N_h = 4, N_v = 4$.

As shown in Table 3, in each configuration the gNB antenna array has the same number of antenna elements, which is 64, i.e., 32 in each polarization. In Figure 6, we show the performance for different gNB port configurations.

The throughput curves almost overlap, which is the direct result of the overlap of the average rank and the average MCS curves. This confirms that the MIMO implementation provides a correct balance between the analog and the digital beamforming (i.e., when using more transmit antenna ports, we have more gain from the digital beamforming but less from the analog beamforming). It is interesting to see that both analog and digital provide similar gains, and this is why the curves are so similar, because of the near-perfect alignment of gNB and UE antennas. Also, as expected the increase in the vertical ports does not help, instead, the impact of increasing the horizontal ports is slightly noticeable in the average rank and the average MCS curve, which makes sense in mainly 2D deployments. The results of this campaign validate the MIMO implementation. As expected, the execution time increases with the increase in

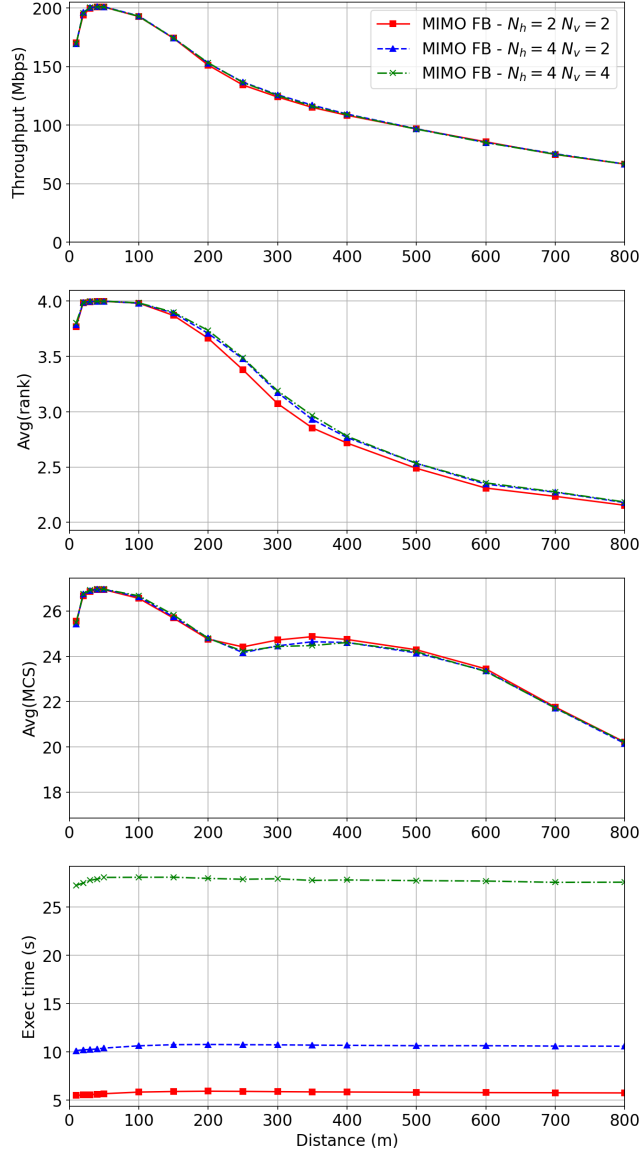


Figure 6: Different gNB ports configurations 8, 16, and 32

the number of ports. Therefore, in situations with good alignment of antenna arrays, it is better to use analog beamforming and a lower number of antenna ports, to have less complex gNB/UE device operations.

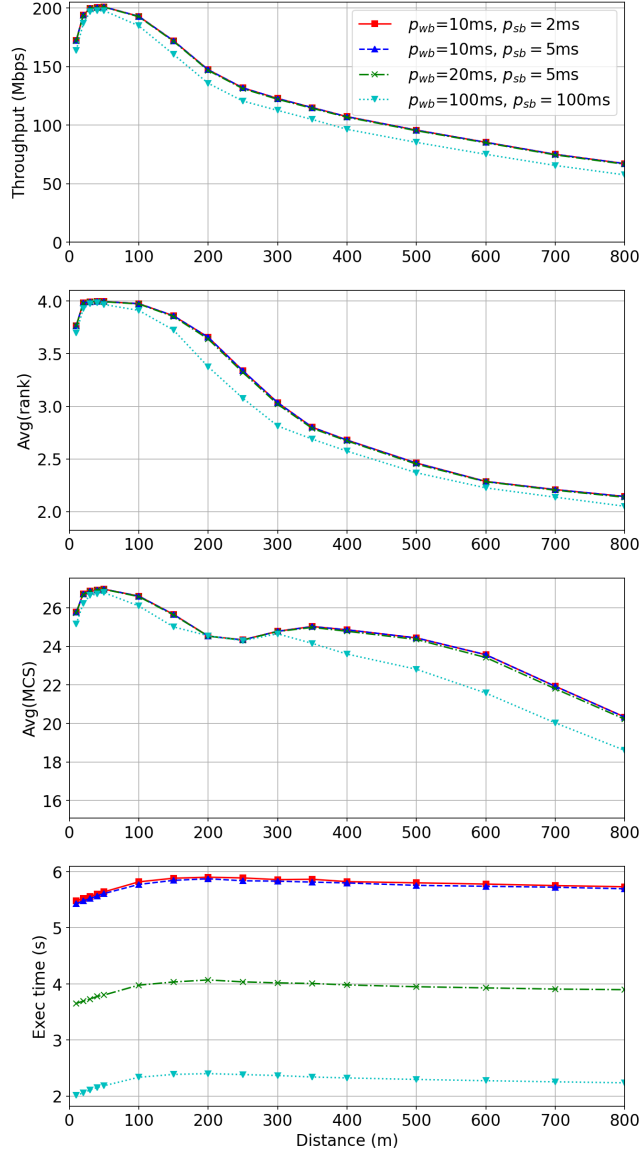


Figure 7: Impact of different SB and WB PMI periodicity

4.5. Impact of different SB and WB PMI update intervals

Even if in the ideal case the SB and WB precoding matrix should be updated on each received signal from the gNB, to reduce the computational complexity, we have created two parameters that allow us to configure how often the WB and

SB precoding matrices will be updated. This follows the 3GPP standard in the sense that CSI reporting is over a certain period. In this simulation campaign, we vary some typical values that we use in the simulation campaigns, such as 10ms and 2ms for WB PMI and SB PMI updates, respectively. However, we also set a much higher value to analyze the impact that the update frequency has on the final achieved throughput and the computational complexity.

In Figure 7 we illustrate the impact of different SB and WB PMI periodicities. We observe that there is almost no difference in achieved throughput for the three configurations with the frequencies $p_{wb} = 10\text{ms}$, $p_{wb} = 20\text{ms}$, but we can notice an important difference in the simulation execution time between the $p_{wb} = 10\text{ms}$ and $p_{wb} = 20\text{ms}$, i.e., the simulation time increases around 1.5 times. In the fourth case in which we set $p_{wb} = 100\text{ms}$, $p_{sb} = 100\text{ms}$, we observe a drop in the throughput performance of about less than 7.5%, but we achieve the speedup in the simulation execution of almost 3 times.

This simulation campaign shows that in the real system if the maximum throughput is not needed, the energy could be saved by reducing the frequency of the WB and SB PMI updates. On the other hand, in the simulator, we could significantly reduce the execution time, even without impacting the performance, e.g. reducing the WB PMI update frequency from 10ms to 20ms ⁴.

4.6. Scenario with the interference

The original topology composed of a single gNB and UE pair is extended to support the possibility of enabling interfering gNB and UE pair. The interfering gNB is placed at a fixed distance of 1000 meters from the original gNB and it is placed on the x-axis its position is $(x,y,z) = (1000,0,25)$. The UE of the interfering gNB and UE pair is placed at the fixed position with coordinates $(x,y,z)=(500, 0, 1.5)$. The bearing angles of the interfering gNB and UE are 180 and 0, respectively. The antennas of the original and interfering gNB are

⁴These values are not cross-checked against the 3GPP standard and common cellular configuration parameters, but just simulated to see the impact of various periodicities.

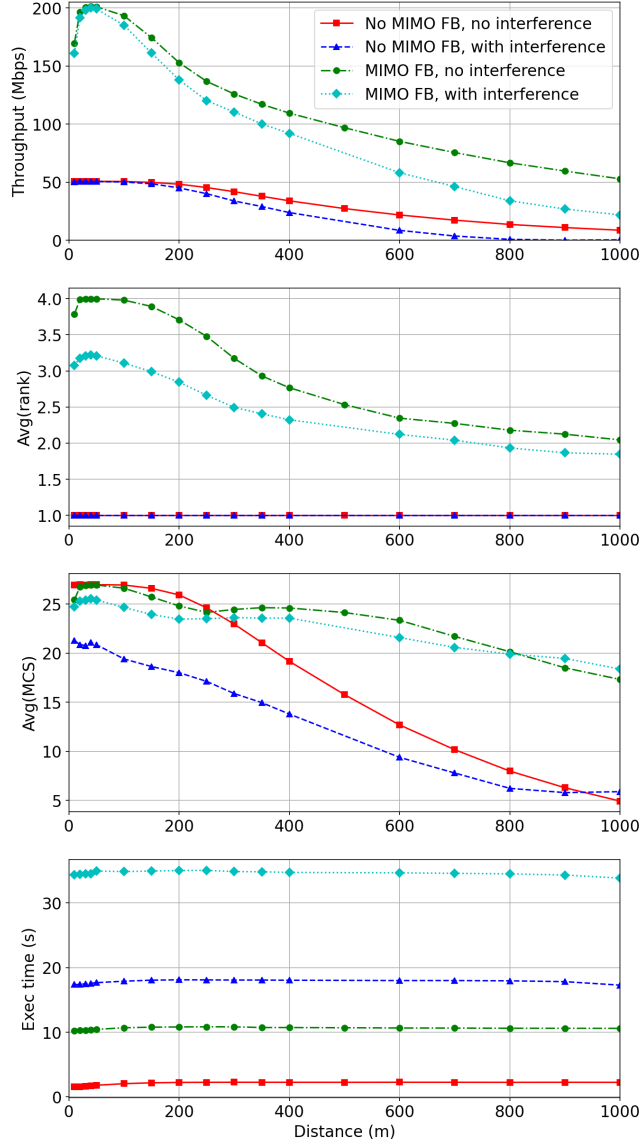


Figure 8: Impact of the interference

configured according to Conf 3 with 16 ports, while UEs' antennas are configured as in Conf 2. In Figure 8 we illustrate the performance of the four cases: when the MIMO feedback is disabled and enabled and when the interfering node is not present and when it is. As expected, we observe better throughput

performance when the interfering node is not enabled regardless of whether the MIMO feedback is enabled. When MIMO feedback and the interfering node are enabled, we observe that on average a lower rank is selected, leading to very similar average values of MCS, i.e., when the interfering node pair is present the lower rank is on average selected, which allows using on average a higher MCS. In this simulation campaign, we can observe a huge increase in the execution time when the interfering node pair is enabled. Also, the simulation time increases by more than 15x when having just an additional node pair and the MIMO FB is enabled.

5. Considerations on MIMO models fidelity vs computational complexity in ns-3 network simulator

ns-3 and 5G-LENA simulators are network simulators, as such their main objective is to allow full-stack end-to-end and large-scale network simulations. In addition, NGMN and 3GPP extended reality and cloud gaming traffic models have been recently included in the 5G-LENA simulator, to allow the simulation of different “hot topic” scenarios related to augmented, virtual, and mixed reality in 5G and beyond [9]. As mentioned earlier, the academy and industry are each time more and more interested in having a high-fidelity PHY layer in network simulators such as ns-3 and 5G-LENA. This poses many challenges related to the dramatically increased computational complexity of such simulations. As it could be observed through multiple simulation campaigns presented in previous sections, the MIMO model adds a level of computational complexity to the already computationally expensive ns-3 3GPP spatial channel model. One of the main bottlenecks is the exhaustive PM search method, which is executed frequently when MIMO is enabled. For this reason, we started recently to work on alternative solutions for PM search that would cut down the simulation execution time. We consider this as a very interesting research topic, for two reasons: 1) the ns-3 and 5G-LENA simulators need much lighter PHY models to allow its main promise which is to perform large-scale end-to-end

simulations, and 2) improvements in the PM search algorithm are of crucial interest for the real UE devices, e.g., mobile phones, where the computational complexity translates directly into energy consumption. Hence, to allow the use of MIMO in real devices without draining a UE device’s battery, it is necessary to come up with computationally efficient PM search algorithms, which at the same time provide good performance in selecting the rank and PMI.

Yet another performance degradation that was introduced with the MIMO model is due to the removal of the “OFDMA downlink trick”. For example, in the scenario in which we have a single gNB and 10 UEs, in the case of OFDMA transmissions, if gNB was transmitting at the same time to 10 UEs, we would have only a single transmit signal entering into the spectrum channel. Since 10 UEs are listening, that means that we had 10 times called functions `DoCalcRxPowerSpectralDensity`. However, now instead of 1 signal, we have an independent signal per each UE, hence, 10 transmit signals will enter into the spectrum channel, and each of them will be received by 10 UEs, resulting in 100 calls of `DoCalcRxPowerSpectralDensity`. This causes the degradation of the 5G-LENA simulator performance, even when MIMO is disabled. We plan to solve this soon, as soon as CSI-RS is implemented and hence each UE in the OFDMA system will have a reference signal based on which it will be able to estimate the channel in all the RBs. Meanwhile, there is no CSI-RS implementation, the 5G-LENA model needs each UE to receive all these signals that are not intended for that device, to be able to estimate the channel over all the RBs and perform the PM search in the case of MIMO. Once CSI-RS is implemented, we can filter out undesired signals (i.e., discarding by RNTI identification or based on the non-overlapping RBs).

6. Conclusions and Future work

In this work, we proposed, implemented, and evaluated a full SU-MIMO model for system-level simulation of 5G NR MIMO systems in ns-3, by extending the current framework with the support of MIMO spatial multiplexing to

send multiple data streams towards the same user, thus enabling the hybrid beamforming feature of 3GPP.

The code is publicly available and merged in the master branch of ns-3 5G-LENA module [26]. In this paper, we focused mainly on the most novel aspects of this MIMO implementation in the ns-3 simulator and that is the MIMO digital baseband processing. We have implemented the codebook Type-I precoding (for single panel case), assuming MMSE-IRC receiver for inter-stream interference suppression, as adopted in 3GPP. Compared to the old DP-MIMO implementation, the new model provides a more generic and realistic model for DP-MIMO in which the streams, ports, and antenna mapping are not predefined, and instead are defined through digital precoding. Thanks to this feature a more accurate model for the inter-stream interference calculation is considered. The current model supports up to 32 antenna ports at gNB and UE and 4 ranks per user.

Through multiple simulation campaigns, we tested different aspects of the newly implemented full SU-MIMO model, and the results showed that the model performs correctly. New MIMO interference computations and PM search algorithms increase significantly the computational complexity of the 5G-LENA simulator. In our future work, we plan to implement less computationally expensive PM search algorithms and also to provide multiple optimizations that would reduce the computational complexity of the MIMO interference calculations. Such improvements are essential for the use of the implemented MIMO models in large-scale network simulation scenarios. We also plan to extend the current MIMO model to support uplink MIMO and MU-MIMO through the support of Precoding Type-II and new MU-MIMO scheduling mechanisms.

Acknowledgment

The work has received funding from Grant PID2021-126431OB-I00 funded by MCIN/AEI/10.13039/501100011033 and “ERDF A way of making Europe”, TSI-063000-2021-56/57 6G-BLUR project by the Spanish Government, and

Generalitat de Catalunya grant 2021 SGR 00770.

References

- [1] J. Bach Andersen, Antenna arrays in mobile communications: Gain, diversity, and channel capacity, *IEEE Antennas and Propagation Magazine* 42 (2000) 12–16.
- [2] D. P. Palomar, A Unified Framework for Communications through MIMO Channels, Ph.D. thesis, Universitat Politècnica de Catalunya, 2003. PhD thesis.
- [3] Z. Pi, F. Khan, An introduction to millimeter-wave mobile broadband systems, *IEEE Communications Magazine* 49 (2011) 101–107.
- [4] J. G. Andrews, T. Bai, M. N. Kulkarni, A. Alkhateeb, A. K. Gupta, R. W. Heath, Modeling and analyzing millimeter wave cellular systems, *IEEE Transactions on Communications* 65 (2017) 403–430.
- [5] S. Lagén, L. Giupponi, S. Goyal, N. Patriciello, B. Bojović, A. Demir, M. Beluri, New radio beam-based access to unlicensed spectrum: Design challenges and solutions, *IEEE Communications Surveys Tutorials* 22 (2020) 8–37.
- [6] TSG RAN; NR; Physical layer procedures for data, 3GPP TS 38.214, Release 15, v18.1.0, Jan 2024.
- [7] I. Ahmed, H. Khammari, A. Shahid, A. Musa, K. S. Kim, E. De Poorter, I. Moerman, A survey on hybrid beamforming techniques in 5g: Architecture and system model perspectives, *IEEE Communications Surveys & Tutorials* 20 (2018) 3060–3097.
- [8] T. R. Henderson, M. Lacage, G. F. Riley, C. Dowell, J. Kopena, Network simulations with the ns-3 simulator, *SIGCOMM demonstration* 14 (2008).

- [9] B. Bojović, S. Lagén, K. Koutlia, X. Zhang, P. Wang, L. Yu, Enhancing 5g qos management for xr traffic through xr loopback mechanism, *IEEE Journal on Selected Areas in Communications* 41 (2023) 1772–1786.
- [10] Revised SI on XR Evaluations for XR, Qualcomm, RP-201145, 3GPP TSG-RAN 88e Meeting, 2020.
- [11] N. Patriciello, S. Lagen, B. Bojovic, L. Giupponi, An E2E Simulator for 5G NR Networks, *Simulation Modelling Practice and Theory* 96 (2019) 101933.
- [12] M. Mezzavilla, M. Zhang, M. Polese, R. Ford, S. Dutta, S. Rangan, M. Zorzi, End-to-End Simulation of 5G mmWave Networks, *IEEE Communications Surveys & Tutorials* 20 (2018) 2237–2263.
- [13] B. Bojovic, S. Lagen, L. Giupponi, Realistic Beamforming Design using SRS-based Channel Estimate for ns-3 5G-LENA Module, in: *Proceedings of the Workshop on ns-3, WNS3 2021, Association for Computing Machinery*, 2021, pp. 81–87.
- [14] B. Bojovic, Z. Ali, S. Lagen, ns-3 and 5G-LENA Extensions to Support Dual-Polarized MIMO, in: *Proceedings of the 2022 Workshop on Ns-3, WNS3 '22, Association for Computing Machinery, New York, NY, USA*, 2022, p. 1–9.
- [15] F. Gómez-Cuba, T. Zugno, J. Kim, M. Polese, S. Bahk, M. Zorzi, Hybrid beamforming in 5g mmwave networks: A full-stack perspective, *IEEE Transactions on Wireless Communications* 21 (2022) 1288–1303.
- [16] H. Assasa, N. Grosheva, T. Ropitault, S. Blandino, N. Golmie, J. Widmer, Implementation and evaluation of a wlan ieee 802.11ay model in network simulator ns-3, in: *Proceedings of the Workshop on ns-3, WNS3 2021, Association for Computing Machinery, New York, NY, USA*, 2021, p. 9–16.
- [17] H. Assasa, J. Widmer, J. Wang, T. Ropitault, N. Golmie, An implementation proposal for ieee 802.11ay su/mu-mimo communication in ns-3, in:

- Proceedings of the 2019 Workshop on Next-Generation Wireless with ns-3, WNGW 2019, Association for Computing Machinery, New York, NY, USA, 2019, p. 26–29.
- [18] S. Jin, S. Roy, W. Jiang, T. R. Henderson, Efficient abstractions for implementing tgn channel and ofdm-mimo links in ns-3, in: Proceedings of the 2020 Workshop on ns-3, WNS3 2020, Association for Computing Machinery, New York, NY, USA, 2020, p. 33–40.
 - [19] 5G-LENA, Available at <https://5g-lena.cttc.es/>, 2024.
 - [20] J. M. Cioffi, Appendix D - MMSE Estimation, 2002. <https://cioffi-group.stanford.edu/>.
 - [21] 3GPP, Study on Channel Model for Frequencies from 0.5 to 100 GHz, TR 38.901 (Rel. 15), v16.1.0, 2020.
 - [22] M. Pagin, S. Lagén, B. Bojovic, M. Polese, M. Zorzi, Improving the Efficiency of MIMO Simulations in ns-3, in: Proceedings of the 2023 Workshop on Ns-3, WNS3 '23, Association for Computing Machinery, New York, NY, USA, 2023, p. 1–9.
 - [23] Eigen library, Available at <https://eigen.tuxfamily.org/>, 2024.
 - [24] ns-3 Network Simulator, Available at <https://www.nsnam.org/>, 2024.
 - [25] 3GPP, TR 36.897, Study on elevation beamforming / Full-Dimension (FD) Multiple Input Multiple Output (MIMO) for LTE, V13.0.0, 2015.
 - [26] B. Bojovic, K. Koutlia, S. Lagen, N. Patriciello, Z. Ali, L. Giupponi, G. Ferreira, A. Ashtari, A. Larrañaga, 5G-LENA ns-3 nr module, Release 3.0, 5g-lena-v3.0.y, Zenodo DOI:10.5281/zenodo.10670856, 2024.
 - [27] On CSI feedback Type I, Intel Corporation, 3GPP R1-1707360, 3GPP TSG RAN WG1 1 89 Meeting, 2017.