# Taming the latency in multi-user VR 360°: A QoE-aware deep learning-aided multicast framework

Cristina Perfecto, *Member, IEEE*, Mohammed S. Elbamby, *Member, IEEE*,
Javier Del Ser, *Senior Member, IEEE*, Mehdi Bennis, *Senior Member, IEEE*

*Abstract*—Immersive virtual reality (VR) applications require ultra-high data rate and low-latency for smooth operation. Hence in this paper, aiming to improve VR experience in multi-user VR wireless video streaming, a deep-learning aided scheme for maximizing the quality of the delivered video chunks with low-latency is proposed. Therein the correlations in the predicted field of view (FoV) and locations of viewers watching 360° HD VR videos are capitalized on to realize a proactive FoV-centric millimeter wave (mmWave) physical-layer multicast transmission. The problem is cast as a frame quality maximization problem subject to tight latency constraints and network stability. The problem is then decoupled into an HD frame request admission and scheduling subproblems and a matching theory game is formulated to solve the scheduling subproblem by associating requests from clusters of users to mmWave small cell base stations (SBSs) for their unicast/multicast transmission. Furthermore, for realistic modeling and simulation purposes, a real VR head-tracking dataset and a deep recurrent neural network (DRNN) based on gated recurrent units (GRUs) are leveraged. Extensive simulation results show how the content-reuse for clusters of users with highly overlapping FoVs brought in by multicasting reduces the VR frame delay in 12%. This reduction is further boosted by proactiveness that cuts by half the average delays of both reactive unicast and multicast baselines while preserving HD delivery rates above 98%. Finally, enforcing tight latency bounds shortens the delay-tail as evinced by 13% lower delays in the 99th percentile.

*Index Terms*—Mobile virtual reality (VR) streaming, 5G, multicasting, millimeter wave (mmWave), Lyapunov optimization, deep recurrent neural network (DRNN), hierarchical clustering, resource allocation.

## I. Introduction

**V**IRTUAL reality (VR) is expected to revolutionize how humans interact and perceive media by inducing artificial sensory stimulation to the brain and immersing them into an alternative world [1]. Yet, given the well established fact that the quality of the visual feedback highly impacts the sense of presence [2], for true engagement to succeed high definition (HD) content needs to be consistently streamed while the end-to-end latency or motion-to-photon (MTP) delay is kept below 15-20 milliseconds. Otherwise VR sickness or cybersickness –a phenomenon similar to motion sickness due to the exposure to low quality or delayed VR content– might ruin the experience. For this reason, high-end VR manufacturers have been long compelled to using wired connections between the head mounted displays (HMDs) and VR servers with high processing and storage capabilities. However, this constraint physically limits the movement of VR users and hence, degrades the quality-of-experience (QoE), thereby calling for further development of mobile/wireless solutions that are able to provide both convenience and a high-quality VR. Moreover, as access to social VR experiences surges, driven by location-based VR and 360° formats [3], the gap between the available bandwidth and the prohibitively high demands by mobile immersive VR is likely to prevail. Clearly, disruptive content provisioning paradigms are needed to unleash the plethora of new business opportunities for leisure/entertainment industry that mobile interconnected VR will bring. In this context, mobile VR spearheads the newly coined highly reliable low latency broadband (HRLLBB) use cases sitting across enhanced mobile broadband (eMBB) and ultra reliable low latency communication (URLLC) service categories in Fifth Generation (5G) networks [4]. The distinctive feature of HRLLBB, if compared to URLLC [5], is the need to reliably provide massive data delivery to multiple users with low-latency.

Focusing on omnidirectional 360° or spherical video format [6] and reducing its bandwidth consumption, video coding solutions that adapt the streaming to users' attention by tracking their visual region of interest are abundant in the literature. Their common goal is to stream in HD[1] only users' *viewports*, i.e., the portion of sphere in a user's field of view (FoV) while wearing an HMD and, optionally, in lower quality the rest. To do so, foveated, tile-based, and projection/viewport-based FoV-streaming are the most commonly adopted approaches, all requiring real-time tracking of either users' eye-gaze or head angles i.e., the 3 degrees-of-freedom (3DoF) *pose* expressed by yaw, pitch and roll angles[2] as represented in Fig. 1. Foveated solutions as per [7], [8] are conditioned to availability of advance eye-gaze tracking mechanisms in the HMDs and real-time frame rendering in the servers, whereas

[1]In the context of 360°, 4K is widely viewed as the minimum resolution in current HMDs, and ideally 8K or higher is desired.

[2]Pitch, yaw and roll head orientation angles represent rotations over the x, y and z-axis, respectively.
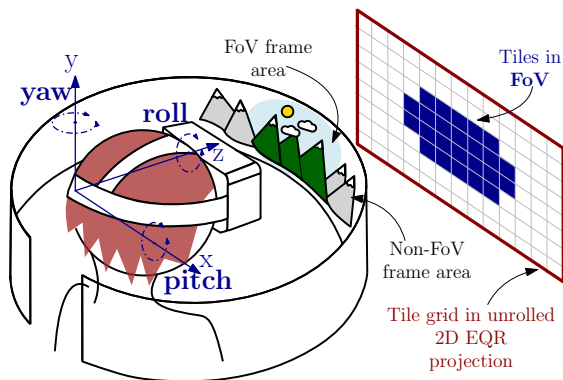
Figure 1. Tiled-FoV mapping of a user's 3DoF pose in the equirectangular (EQR) projection of a 360° video frame.
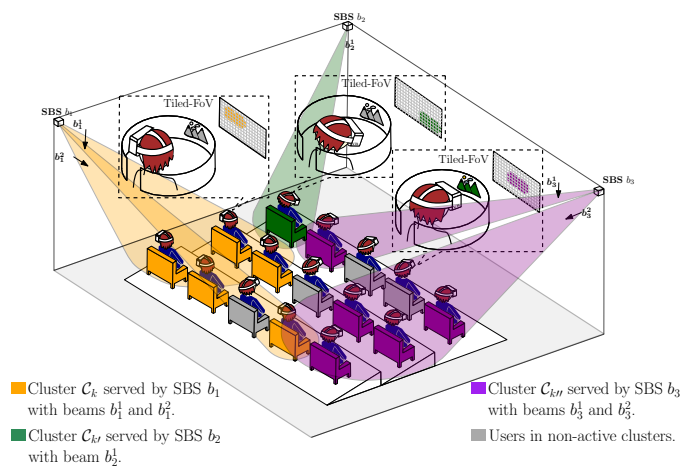


Figure 2. Schematic representation of tiled-FoV mmWave multicast scheduled transmission to VR users with overlapping FoVs. Users belonging to a given cluster are served by a single mmWave SBS through different non-overlapping beams of variable beamwidth.

for tile and viewport-based solutions, regular head-tracking is enough. The main drawback of projection-streaming [9] lies in its large server storage needs, given that for each frame multiple viewpoints are kept. Lastly, in tile-based streaming approaches as per [10]–[13], the video frame is divided in a grid of regular tiles, each encoded in both HD and at a lower resolution. Then, only the tiles within a user's FoV region are streamed in HD.

Real-time tile-based FoV streaming to a network of VR users involves a number of time-consuming steps: edge controllers/servers need to first acquire pose data, process it to compute the set of tiles within the FoV, and lastly schedule their transmission. Then, on-HMD processing will be performed to compose –stich–and display the corresponding portion of the video frame. The end-to-end delay of the process is non-negligible. Thus, as the number of users in the network increases, operating this cycle within the MTP delay budget for each frame for every user becomes challenging; even more so if the server/edge controllers and users are not wired. The latter realization calls for unconventional solutions at different network levels. Among them, the use of proactive transmission of FoV tiles –contingent on the availability of prediction results on the FoV for the upcoming frames– can significantly improve the efficiency of content delivery even with non-perfect prediction accuracy, especially in fluctuating channel environments, as illustrated by the following numerical example: Let the tile size be 1 Mb and the FoV of the user in each frame have a total of 32 tiles. This means that without proactivity, a rate budget of 32 Mb on each frame transmission interval is needed to fully deliver a user's FoV tiles. Assume that a user's rate budget in the transmission period of two subsequent frames $f_1$ and $f_2$ be 70 Mb and 10 Mb, respectively. This will lead to an outage in the second frame. Assume that during the transmission period of frame $f_1$, the small cell base station (SBS) predicts the user's future FoV of frame $f_2$ with an accuracy of 75%, i.e., 24 tiles out of the 32 are correctly predicted. Then, using proactive transmission, the SBS can deliver during the first transmission period the 32 tiles of frame $f_1$ and the predicted 32 tiles of frame $f_2$. When the second transmission period starts and the user reports its actual FoV for frame $f_2$, the SBS will have to only transmit the remaining 25% of the FoV, i.e., 8 tiles, which the user's

rate budget allows handling without experiencing any outage.

### A. Related Work

The need to predict users' fixation opens the door for harnessing machine learning (ML) in wireless systems [14]. Moreover, public availability of extensive pose and saliency datasets for 360° video equips an invaluable input to optimize VR content streaming through supervised and unsupervised deep learning applied to FoV prediction. To that end, *content-based* [15] or *trajectory-based* [11], [16] strategies can be adopted: the former tries to mimic human capacity to detect the most salient regions of an upcoming video frame i.e., its point of interests (PoIs), and predominantly rely on convolutional neural networks (CNNs) to do so. The latter tracks head (or eye-gaze) movement of VR users through their HMDs and largely relies on the use of recurrent neural networks (RNNs). However, due to the distinctive nature of 360° videos, the use of content-based information alone might not be suitable since the PoIs in a video frame might fall out of a user's current FoV and not affect the viewing transition [17]. Similarly, both having several PoIs or none within the FoV might be problematic. Nevertheless, the availability of proper video saliency, could strongly improve the accuracy of the head movement prediction by combining saliency-based head movement prediction with head orientation history, as exemplified by [18]–[21]. Recently deep reinforcement learning (DRL) i.e., the hybridization of deep-learning with reinforcement learning (RL), have provided the required tractability to deal with highly-dimensional state and action space problems. Hence, its application to 360° video streaming has been facilitated. Good illustration of this trend are the works in [22] and [23], where a DRL-based head movement prediction and a general video-content independent tile rate-allocation framework are respectively presented. Despite the rich recent literature on FoV prediction, most of these and other works provide bandwidth-aware adaptive rate schemes with a clear focus on Internet VR video streaming. Hence, they do not address the particularities of the mobile/cellular environment.

Furthermore, HMDs have limited computing power and storage capacity, so the above predictions need to be offloaded to the network edge. In this sense, the emerging role of edge computing for wireless VR and its relation with communication has been largely covered in the recent literature [4], [24]–[28]. Therein, [4] outlined the challenges and the technology enablers to realize a reliable and low-latency immersive mobile VR experience. Whereas [24]–[28], explore different optimization objectives while investigating some fundamental trade-offs between edge computing, caching, and communication for specific VR scenarios. Yet, most of the few works considering VR multi-user scenarios focus either on direct [29] or device-to-device (D2D)-aided [30] content broadcasting, disregarding the potentials for bandwidth savings and caching of existing correlations between VR users or contents [31]. Also the few works that leverage content correlation through ML, such as [32], none capitalizes prediction related information to perform a proactive mmWave multicast transmission. Furthermore, to the best of the authors' knowledge, imposing high-reliability and low-latency constraints on such wireless VR service problem has not been studied so far.

Therefore, this manuscript proposes to incorporate ML and multicasting into the optimization problem of maximizing the streaming quality of FoV-based 360° videos with HRLLBB guarantees. The use of ML to predict users' FoV in advance and leverage inter-user correlations is pivotal to the system. Then, building upon the aforementioned correlations, multicast transmissions aimed for clusters of users with partially or fully overlapping FoVs will be proactively scheduled such that strict latency bounds are kept. Moreover, the adoption of millimeter wave (mmWave) frequency band communications –where at each time slot a given SBS will steer multiple spatially orthogonal beams towards a cluster of users– to transmit contents is key to benefiting from high transmission rates that contribute to a reduced on-the-air delay. For clarity, we summarize the main contributions of this paper as follows:

- To provide high capacity while ensuring bounded latencies in wireless 360° VR streaming, we propose a proactive physical-layer multicast transmission scheme that leverages future content and user location related correlations.
- We model the problem as a quality maximization problem by optimizing the users' HD frame request admission and scheduling to SBSs and, by borrowing tools from dynamic stochastic optimization, we recast the problem with traffic load stability and latency bound considerations. Subsequently, to solve it, a low complexity and efficient algorithm based on matching theory is proposed.
- To validate our proposed scheme, for FoV prediction we developed a deep recurrent neural network (DRNN) based on gated recurrent units (GRUs) that is trained with a dataset of real 360° VR poses. The predicted FoVs for a given time horizon are thus available to dynamically cluster users based on their FoV overlap and proximity within a VR theater, as well as to provide insights on how much the scheme is affected by the accuracy of the prediction and the clustering decisions.

- Extensive simulations are conducted to investigate the effects of the prediction horizon, of the VR frame size, the clustering, and of the network size, which conclude that the proposed approach outperforms considered reference baselines by delivering more HD quality frames, while ensuring tight transmission delay bounds.

The remaining of this manuscript is organized as follows: In Section II the system model and the underlying assumptions are described. The optimization problem of wireless VR video content delivery is formulated in Section III. Section IV presents our proposed matching theory algorithm to schedule wireless multicast/unicast transmission resources for VR video chunks under latency constraints. A detailed description of the FoV prediction and adopted user clustering schemes is provided in Section V. Simulation results and performance evaluation are presented in Section VI. Finally, Section VII concludes the paper.

*Notations*: Throughout the paper, lowercase letters, boldface lowercase letters, (boldface) uppercase letters and italic bold-face uppercase letters represent scalars, vectors, matrices, and sets, respectively. $\mathbb{E}[\cdot]$ denotes the expectation operator and $\Pr(\cdot)$ the probability operator. Function of $z$ and utility of $z$ are correspondingly expressed as $f(z)$ and by $U(z)$. $\mathbb{I}_{\{z\}}$ is the indicator function for logic $z$ such that $\mathbb{I}_{\{z\}} = 1$ when $z$ is true, and 0, otherwise. The cardinality of a set $\mathcal{S}$ is given by $S = |\mathcal{S}|$. Moreover, $[z]^+ \triangleq \max(z, 0)$ and $\overline{z}$ stands for the time average expectation of quantity $z$, given by $\overline{z} = \lim_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}[z(t)]$. Lastly, $\circledast$ represents the Hadamard product, $\tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$ is the hyperbolic tangent and $\sigma(z) = \frac{1}{1+e^{-z}}$ the sigmoid activation functions for the neural network.

## II. SYSTEM MODEL

In this section we introduce the system model which encompasses the considered deployment scenario, as well as the adopted wireless channel and communication models. For ease of reading, a non-comprehensive list of the notations used throughout the rest of the manuscript is provided in Table I.

### A. Deployment Scenario

We consider a VR theater with seats arranged in $s_r$ rows and $s_c$ columns, and where a network of VR users $\mathcal{U}$, all wearing mmWave HMDs, are located. In this scenario, each user chooses to watch an HD 360° VR video $v \in \mathcal{V}$, with $\mathcal{V}$ denoting the set of available VR videos in the catalog. Due to their large size and to limited storage capacity in the HMDs, the HD frames of videos are cached in the edge network and are delivered to users through $B = |\mathcal{B}|$ SBSs distributed around the theater. To ensure timely and smooth streaming experience, a lower-quality standard definition (SD) version of the video frames are assumed to be pre-cached in the user's HMD to be streamed if the HD frames are not successfully delivered on time. The SBSs operate in the mmWave band and are endowed with multi-beam beamforming capabilities to boost physical layer multicast transmission [33] of shared video content to users grouped into clusters. This setting is graphically represented in Fig. 2.

Table I
SUMMARY OF MAIN NOTATIONS

| Symbol | Description | Symbol | Description |
|---|---|---|---|
| **Time and Indexing** | | **Channel Model** | |
| $t, T_t$ | Time index and slot duration | $h_{bu}$ | Channel gain for user $u$ from SBS $b$ |
| $t_a$ | Frame request arrival time | $P\ell_{bu}, S\ell_{bu}, B\ell_{bu}$ | Pathloss, shadowing and blockage from $b$ to user $u$ |
| $f$ | Video frame index | $P\ell_{\text{LOS}}, P\ell_{\text{NLOS}}$ | LOS and NLOS pathloss |
| $T_f$ | Time between frames | $\varsigma_{\text{LOS}}^{SF}, \varsigma_{\text{NLOS}}^{SF}$ | LOS and NLOS shadowing variance |
| $f_r$ | Real-time frame index | $f_c$ | Normalized central frequency |
| $f_P$ | Prediction frame index | $d_{bu}^{2D}, d_{bu}^{3D}$ | Azimuth and elevation plane distance |
| **Sets** | | $T_{\text{block}}$ | Time between blockage events |
| $\mathcal{U}, \mathcal{U}_{tr}$ | VR users (test set) and training users | **Communication Model** | |
| $\mathcal{B}$ | SBSs | $g_{bu}^{\text{Tx}}, g_{bu}^{\text{Rx}}$ | Transmit/receive antenna gains of SBS $b$ to user $u$ |
| $\mathcal{V}$ | Videos | $\varphi^{tx}, \varphi^{rx}$ | Transmit and receive beamwidths |
| $\mathcal{F}$ | Frames indexes in a video | $\vartheta_{bu}^{tx}, \vartheta_{bu}^{rx}$ | Transmit and receive beams angular deviation. |
| $C$ | VR clusters | $p_b, p_{b'}$ | Transmit powers of SBS $b$ and interfering SBSs $b'$ |
| $C_k^f$ | VR users in $k$-th cluster for video frame index $f$ | $BW_b$ | Bandwidth for SBS $b$ in mmWave band |
| $\mathcal{N}_u^f, \mathcal{N}_{C_k}^f$ | FoV tiles of user $u$ and cluster $C_k$ for frame index $f$ | $\text{SINR}_{bu}$ | SINR for user $u$ being served by SBS $b$ |
| $\widehat{\mathcal{N}}_u^f, \widehat{\mathcal{N}}_{C_k}^f$ | Predicted tiles in the FoV of user $u$ and of cluster $C_k$ at frame index $f$ | $\text{SINR}_{b, C_k^f}^{c_f}$ | Multicast SINR of chunk $c_f$ from SBS $b$ to cluster $C_k^f$ |
| **Problem Formulation** | | $I_u$ | Instantaneous interference for user $u$ |
| $r_u(t)$ | Total traffic admission for user $u$ | $N_0$ | Noise power spectral density |
| $L_{c_f}$ | Data size of chunk $c_f$ | **Lyapunov Optimization Framework** | |
| $a_{uf}, \eta_{buc_f}$ | Chunk admission and scheduling variables | $q_u$ | Traffic queue of user $u$ |
| $\tau_{uf}$ | Transmission delay of frame $f$ to user $u$ | $z_u, j_{uf}$ | Virtual queues of time averaged constraints |
| $\epsilon_d$ | Delay reliability metric | $\gamma_u$ | Auxiliary variables for the EOP |
| $\mu_{uc_f}$ | Rate of delivering chunk $c_f$ to user $u$ | $L(\cdot), \Delta L_t$ | Lyapunov and Lyapunov-drift functions |
| $d_{t2\text{MTP}}$ | Time left to schedule before MTP is exceeded | $V_\Delta$ | Lyapunov drift-plus-penalty trade-off variable |
| $\tau_{\text{MTP}}$ | Motion-to-photon latency | **FoV Prediction and DRNN** | |
| **Matching Theory** | | $M_\theta^{v, T_H}$ | Supervised learning model for video $v$ and $T_H$ |
| $\Upsilon, >_b, >_{C_k}$ | Matching function and preference relations | $M, \theta$ | Learning algorithm and parameter set |
| $U_{\mathcal{B}}^{b, C_k}, U_C^{C_k, b}$ | Clusters and SBSs utilities | $X_{tr}^v, Y_{tr}^{v, T_H}$ | Training dataset and binary-encoded matrix of target tiles |
| $\hat{U}_{\mathcal{B}}^{b, C_k}, \hat{U}_C^{C_k, b}$ | Modified utilities over the estimated parameters | $T_H, T_P$ | Prediction horizon and input sequence length |
| $\nu_1, \nu_2$ | Weight and sample number of moving average procedure | $p_{3u_{tr}}^f, p_{3u}^f$ | 3DoF pose vectors of a training user $u_{tr}$ and of a user $u$ |
| $\hat{I}_u, \tilde{I}_u^{\nu_2}$ | Estimated/moving-average interference for user $u$ | $r, \Gamma$ | Reset and update gates of the GRU |
| **User Clustering** | | $h_{f-1}, \tilde{h}_f, h_f$ | Previous, candidate and new hidden states in GRU |
| $\bar{d}_{u,u'}^{f_P}, d_{u,u'}^{f_P}$ | FOV and FOV+ user location based clustering distances | $\alpha, \beta_1, \beta_2$ | Learning rate and parameters for Adam algorithm |

In the network edge, without loss of generality, we assume that all videos in the catalog are encoded at the same frame rate $1/T_f$ –with $T_f$ the time between frames– and have the same length i.e., consist of a set of frames $\mathcal{F} = \{f\}_{f=1}^F \subset \mathbb{N}$. Moreover, the frames from the spherical videos are unwrapped into a 2D equirectangular (EQR) or lat-long projection with pixel dimensions $P_H \times P_V$ and divided into $\mathcal{N} = \{1, \dots, N\}$ partitions or *tiles* arranged in an $N_H \times N_V$ regular grid, so that $N = N_H \cdot N_V$ and each tile is sized $P_H/N_H \times P_V/N_V$ pixels. Therefore, when watching any given video $v \in \mathcal{V}$, the FoV of user $u \in \mathcal{U}$ during frame $f \in \mathcal{F}$ can be expressed as a tile subset $\mathcal{N}_u^f \subseteq \mathcal{N}$.

In a nutshell, two main indices are used in our manuscript: one refers to the decision slots indexed by $t = \{1, 2, \cdots\}$ and slot duration $T_t$ which is the prevailing time scale enforced for transmission/scheduling purposes, whereas the second index $f$ refers to the video frame index. Hence, the slotted time index $t$ and frame index $f$ satisfy $f \triangleq \lceil \frac{t*T_t}{T_f} \rceil$ so that $f = \{1, 2, \cdots\} \in \mathbb{N}$. With this division in mind, *chunk* hereafter denotes the part of the video that corresponds to one frame in time and one tile in EQR space.

### B. FoV and Spatial Inter-user Correlation

To leverage FoV and spatial correlations between users in the VR theater deployment from Section II-A , and as outlined in Fig. 3b , we assume that users report in the uplink (UL) their video index $v$ and their 6 degrees-of-freedom (6DoF) pose every $T_f$ ms. This 6DoF pose includes head orientation

angles and x, y and z-axis coordinates[3]. The information is then forwarded from the SBSs to the edge controller where FoV prediction, user-clustering and scheduling decisions take place.

New real-time/proactive chunk scheduling decisions will be taken every $T_t$ such that for all users chunks are scheduled for downlink (DL) transmission by the SBSs, and delivered before the frame deadline $d_f$ expires. In this regard, the chunks that correspond to real-time i.e., $f = f_r$, and to predicted future FoVs will be given by $\mathcal{N}_u^{f_r}$ and by $\{\widehat{\mathcal{N}}_u^f\}_{f=f_r+1}^{f_P}$, respectively.

To provide the estimated FoVs, let a supervised ML model $M_\theta^{v, T_H}$ be defined in the edge controller –with $M$ denoting the model's learning algorithm and $\theta$ its parameter set– associated to each video $v \in \mathcal{V}$ and time horizon[4] $T_H$ for which a model is constructed as

$$\widehat{y}_u^{f_P} \triangleq M_\theta^{v, T_H}(x_u^f). \tag{1}$$

Once the model's offline training, as detailed in Section V-C, has been completed and its parameters are known, given a sequence $x_u^f$ of $T_P$ length of past 3DoF poses[5] collected in $x_u^f$, the model will produce the vector of labels $\widehat{y}_u^{f_P} \triangleq \{\widehat{y}_{u,n}^{f_P}\}_{n=1}^N$ for frame index $f_P = f + T_H$ as per (1). The corresponding set

---

[3]We notice here that 360° videos are displayed from a centered perspective whereby users are only allowed to look around. Therefore, as users' physical location do not affect their view, for FoV prediction purposes only the 3DoF head orientation needs to be considered, even if the whole 6DoF pose is reported.

[4]Without loss of generality, that the time horizon for the prediction $T_H$ is measured as an integer multiple of the frames.

[5]This sequence consists of the user's last $T_P$ recorded head angles, as per $p_{3u}^f$ with video frame indexes $f \in \{f_r - T_P + 1, \dots, f_r\}$.

of tiles in the predicted FoV is a mapping such that $\{\widehat{\mathcal{N}}_u^{f_p} = \forall n \in [1, \ldots, N]: \widehat{y}_{u,n}^{f_p} = 1\}, \forall u \in \mathcal{U}$.

Subsequently, the predicted FoVs and reported poses will be fed into a user-clustering module whereby users watching the same VR video $v$ will be grouped together based on their FoV and spatial correlation. The inputs for the scheduler will therefore be: $\forall u \in \mathcal{U}$ the real-time FoV tile-sets $\mathcal{N}_u^{f_r}$ for the current index frame $f = f_r$ as well as the predicted $K$ user-clusters $\{C_k^{f_p}\}_{k=1}^K \mid \bigcup_{k=1}^K C_k^{f_p} = \mathcal{U}$ with their corresponding cluster-level predicted FoVs $\{\widehat{\mathcal{N}}_{C_k}^{f_p} = \bigcup_{u \in C_k^{f_p}} \widehat{\mathcal{N}}_u^{f_p}\}_{k=1}^K$.

Since spherical videos are not locally cached, a huge imbalance emerges in the amount of information being sent in the UL vs. DL. Therefore, for the purpose of this manuscript we will only focus on the effect of the DL HD 360° video transmission from edge SBSs to VR users, and assume that enough UL resources are available to users for timely pose update and channel state information (CSI) report.

Also, in the remaining of the manuscript and for a given time horizon $T_H$, following the UL report of users' pose, the availability of the real-time and predicted FoVs as well as of user-clustering partitioning results is assumed. The detailed description of the proposed FoV prediction and user-clustering schemes with their algorithmic implementation details to produce such inputs are provided in Section V. Next, the wireless channel model and the communication model between the SBSs and the VR users are presented.

### C. mmWave Channel and Communication Model

At mmWave frequencies, due to the quasi-optical nature of electromagnetic wave propagation, signals are highly directional. For that reason channels are composed of a single-path propagation component for the dominant path and a set of multi-path components. For tractability and without loss of generality, in this paper we will neglect the multi-path components and consider only the dominant path for the purpose of VR wireless streaming.

In this single-path, we adopt the 3GPP contributed channel model [34] which is valid for frequencies ranging from 0.5 to 100 GHz and bandwidths of up to 10% of the center frequency not exceeding 2 GHz. Among the different scenarios therein, typical indoor deployment cases including office environments and shopping malls, are showcased. Selecting the indoor open office scenario, with user devices and SBSs located at 1 m and 3 m height respectively, a distance dependent line-of-sight (LOS) probability is defined as

$$\Pr(\text{LOS}) = \begin{cases} 1, & d_{bu}^{2D} \leq 5 \text{ m}, \\ \exp\left(-\frac{d_{bu}^{2D}-5}{70.8}\right), & 5 \text{ m} < d_{bu}^{2D} \leq 49 \text{ m}, \\ 0.54\exp\left(-\frac{d_{bu}^{2D}-49}{211.7}\right), & 49 \text{ m} < d_{bu}^{2D}, \end{cases} \quad (2)$$

where $d_{bu}^{2D}$ stands for the distance in meters between the SBS and the user in the azimuth plane. Subsequently, results from $\Pr(\text{LOS})$ are exploited to calculate the large-scale fading effects in the channel. Specifically, pathloss $\ell$ is given (in dB) as follows,

$$\ell_{\text{LOS}} = 32.4 + 17.3 \cdot \log_{10} d_{bu}^{3D} + 20 \cdot \log_{10} f_c, \quad (3)$$

$$\ell_{\text{NLOS}}^{prev} = 38.3 \cdot \log_{10} d_{bu}^{3D} + 17.3 + 24.9 \cdot \log_{10} f_c, \quad (4)$$

$$\ell_{\text{NLOS}} = \max(\ell_{\text{LOS}}, \ell_{\text{NLOS}}^{prev}), \quad (5)$$

with $d_{bu}^{3D}$, $f_c$ in (3) and (4) representing the distance in meters between the SBS and the user in the elevation plane and the channel's central frequency normalized with 1 GHz, respectively. A log-normally distributed shadowing fading loss $S\ell$, with standard deviation for the LOS and non-line-of-sight (NLOS) cases of $\varsigma_{\text{LOS}}^{S\ell} = 3$ dB and $\varsigma_{\text{NLOS}}^{S\ell} = 8.03$ dB respectively, supplements the large-scale fading calculations.

In addition to the pathloss and shadowing fading, the channel intermittency due to sporadic human blockage is also accounted for. This blockage $B\ell(t_B)$ could hinder the communication by bringing 20-30 dB penalty upon the channel gain. To that end, based on the spatial location of users within the VR theater, the prospective human blockers that might obstruct the direct ray in the azimuth plane between a given user and each of the available SBSs are counted. Thereupon, the count-weighted probabilistic arrival of blockage-events is evaluated every $T_{\text{block}}$, with the value of $T_{\text{block}}$ chosen such that $T_{\text{block}}/T_t \gg 1$ and the index $t_B$ satisfy $t_B \triangleq \lceil \frac{t*T_t}{T_{\text{block}}} \rceil$. The reason for operating on larger time-scale lies on correlation between blockage events along several transmission intervals due to the relative slowness of human head and body limb movement with respect to other channel fading effects. Indeed, human blockage durations of few hundreds of ms or more are reported in the literature [35]. By combining the channel as per [34] with the statistic blockage model, both dynamic channel fluctuations and the arrival of sporadic and longer-lasting human blockage events are suitably captured. Accordingly, the channel gain $h_{bu}(t)$ in dB from SBS $b \in \mathcal{B}$ to user $u \in \mathcal{U}$ is given as

$$h_{bu}(t) = \ell_{bu}(t) + S\ell_{bu}(t) + B\ell_{bu}(t_B). \quad (6)$$

To benefit from multi-beam transmission, we assume that SBSs are equipped with a limited number of radio frequency (RF) chains, whereas users' HMDs will have a single RF chain, limiting their beamforming and combining capability. These assumptions are grounded on current high costs and power consumption of analog-to-digital converters for mmWave frequencies. For tractability, the actual antenna radiation pattern is approximated by a 2D sectored antenna model [36] in the SBSs and in the HMDs. In this model antenna gains are considered constant within the mainlobe, and equal to a smaller constant in the sidelobes. Let $g_{bu}^{\text{Tx}}(\varphi^{\text{Tx}}, \vartheta_{bu}^{\text{Tx}}(t))$ and $g_{bu}^{\text{Rx}}(\varphi^{\text{Rx}}, \vartheta_{bu}^{\text{Rx}}(t))$ denote the transmission and reception antenna gains from SBS $b$ to the HMD of VR user $u$ while using beams of beamwidth $\varphi$, given by

$$g_{bu}^{\measuredangle}(\varphi^{\measuredangle}, \vartheta_{bu}^{\measuredangle}(t)) = \begin{cases} \frac{2\pi - (2\pi - \varphi^{\measuredangle})g_{sl}}{\varphi^{\measuredangle}}, & |\vartheta_{bu}^{\measuredangle}(t)| \leq \frac{\varphi^{\measuredangle}}{2}, \\ g_{sl}, & \text{otherwise}, \end{cases} \quad (7)$$

with $\measuredangle \in \{\text{Tx}, \text{Rx}\}$, where $\vartheta_{bu}^{\measuredangle}(t)$ stands for the angular deviation from the boresight directions of SBS $b$ and of VR user $u$, and $g_{sl}$ is the constant sidelobe gain with $g_{sl} \in [0, 1)$. High directionality of mmWave communication often implies a search process to find the best steering directions. In our
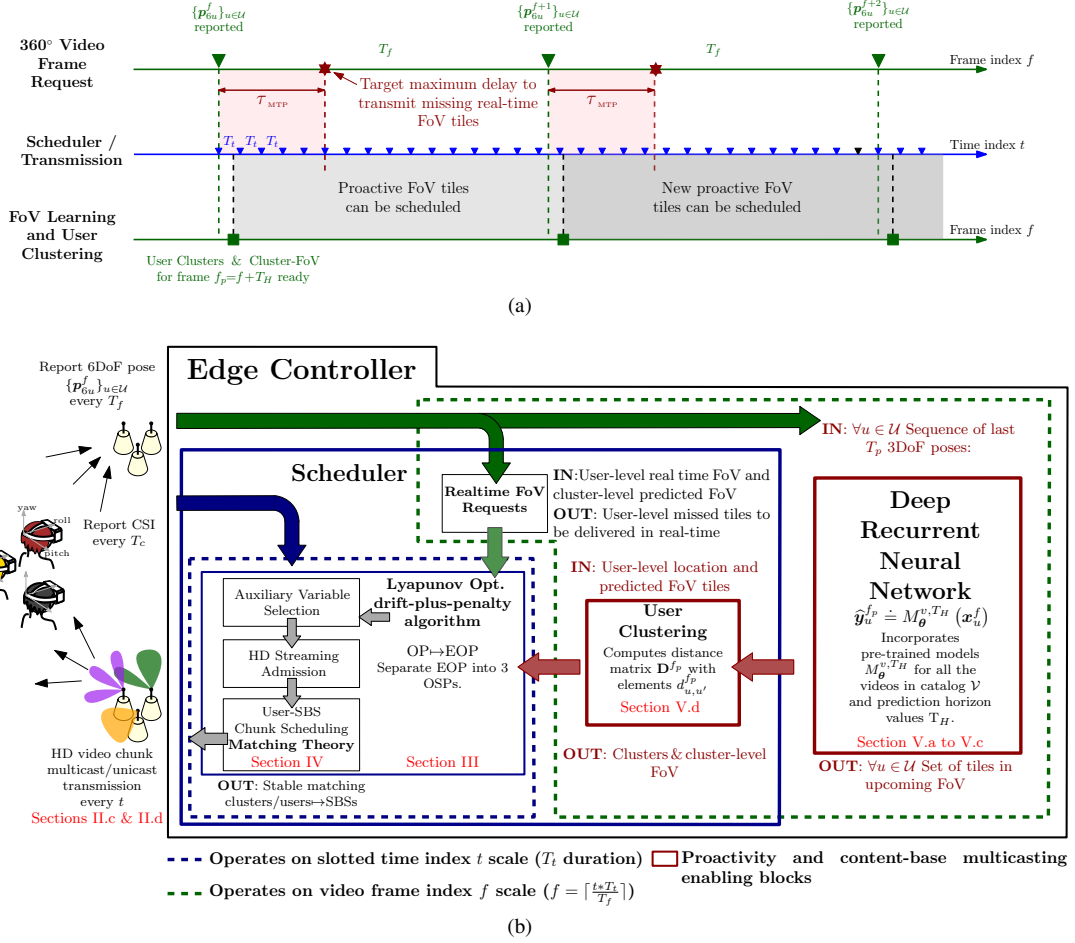
Figure 3. (a) Timing sequence to showcase operation at transmission/scheduling level and at videoframe level. (b) Flowchart of the proposed DRNN aided wireless scheduling of HD 360° video FoV chunks. Once user pose has been reported in the UL, FoV prediction and user clustering follow. Subsequently, a scheduler balances HD chunk admission vs. queue stability subject to latency constraints. Finally a matching theory algorithm associates scheduled chunks to SBS-user clusters pairs leveraging multi-beam mmWave multicast transmission in the DL.

system model, full knowledge of the seating area layout and of the fixed locations of the $B$ SBSs is assumed. Moreover, as stated before, users in $\mathcal{U}$ will report their 6DoF pose information in the UL with $T_f$ periodicity. With these assumptions, even if both the SBSs and users are aware of each other's location and know a priori what their respective optimal beams' boresight directions are, the above antenna model effectively captures subtle misalignment errors arriving from the limited availability of unique beam patterns common in codebook-based beam alignment approaches under analog beamforming. In other words, the angular deviation $\vartheta_{bu}^{\angle}(t)$ in (7) reflects the inability of the analog beamformer to direct the mainbeam at any arbitrary location. The signal-to-interference-plus-noise ratio (SINR) for user $u$ served by SBS $b$ is thus given by

$$\text{SINR}_{bu}(t) = \frac{p_b h_{bu}(t) g_{bu}^{\text{Rx}}(t) g_{bu}^{\text{Tx}}(t)}{I_u(t) + BW_b N_0}, \qquad (8)$$

where the numerator represents the power of the received signal at user $u$ from SBS $b$ under transmit power $p_b$, and the denominator is the sum of the interference power and Gaussian noise power. In our system $I_u(t) = \sum_{b' \in \mathcal{B} \setminus \{b\}} p_{b'} h_{b'u}(t) g_{b'u}^{\text{Rx}}(t) g_{b'u}^{\text{Tx}}(t)$ is the interference that ar-

rives from the transmission of other SBSs reaching user $u$ through channel, transmit and receive antenna gains, and power level $h_{b'u}(t)$, $g_{b'u}^{\text{Rx}}(t)$, $g_{b'u}^{\text{Tx}}(t)$ and $p_{b'}$, respectively. The noise power is given by the noise power spectral density $N_0$ in watts per hertz multiplied by the system bandwidth $BW_b$.

Note that since multicast transmission is considered, the achievable rate of user $u \in C_k^f$ depends on the composition of $C_k$ for each frame index $f$, with the assumptions behind this composition being the FoV and spatial correlation as detailed in Section V-D.

## III. PROBLEM FORMULATION

In this section, building upon the multi-user VR scenario described in Section II, we formulate the network-wide optimization problem of scheduling the FoV contents of an HD 360° video frame by a deadline $d_f$ such that VR sickness can be avoided. The problem formulation explicitly incorporates the proactive/real-time nature of content requests, as well as multicast/unicast transmission capabilities in the SBSs. We pose the problem as a frame quality maximization, while the latency constraints and transmission queues stability are maintained.

For admission purposes, in our scenario a user $u$ associated during video frame index $f = f_p$ to a cluster $C_k^{f_p} \subset C$ is allowed to request chunks that belong either to the cluster-level predicted FoV $\widehat{\mathcal{N}}_{C_k}^{f_p}$ or to its real-time FoV $\mathcal{N}_u^{f_r}$. In this sense, the admission of the predicted FoV proactive requests allows to leverage cluster-level multicast transmissions of shared FoV chunks, i.e. content reuse. On their behalf, real-time FoV chunk requests are the result of missed tiles, i.e. $\mathcal{N}_u^{f_r} \setminus \widehat{\mathcal{N}}_{C_k}^{f_r} \neq \emptyset$, due to imperfect prediction accuracy in the DRNN module. In the latter case, these requests need to be expedited to meet MTP latency related constraints and provide a smooth VR experience.

Let $r_u(t)$ be the total traffic admission for user $u \in C_k^f$ with $\bigcup_{k=1}^K C_k^f = \mathcal{U}, \forall f \in \mathcal{F}$

$$r_u(t) = \sum_{f \in \mathcal{F}} \left( \mathbb{I}_{\{f=f_r\}} a_{uf}(t) \sum_{c_f \in \mathcal{N}_u^f} L_{c_f} + \mathbb{I}_{\{f=f_p\}} a_{uf}(t) \sum_{c_f \in \widehat{\mathcal{N}}_{C_k}^f} L_{c_f} \right), \quad (9)$$

where $L_{c_f}$ is the data size of chunk $c_f$, and $a_{uf}(t)$ is a binary variable that indicates if the video frame $f$ is admitted for offloading to user $u$. The aforementioned two-fold nature of the chunk admission requests is evinced through the two separate terms in (9). Moreover, we notice here that the value of $r_u(t)$ in (9) is upper bounded by the maximum value $r_{max}$ such that $r_u(t) = r_{max} \Rightarrow a_{uf} = 1, \ \forall f \in \mathcal{F}$ that, in practice, represents a situation where the system's traffic load is so low that all frames can be admitted in HD without risking queue stability.

Extending the notation of the admission to consider unicast and multicast transmission of chunks in real-time and proactive respectively, the rate of delivering chunk $c_f$ to user $u$ is

$$\mu_{uc_f}(t) = \begin{cases} \sum_{b \in \mathcal{B}} \eta_{buc_f}(t) \mu_{bu}(t), & f = f_r, \\ \sum_{b \in \mathcal{B}} \eta_{buc_f}(t) \min_{\forall u' \in C_k^f | c_f \in \widehat{\mathcal{N}}_{u'}^f} \mu_{bu'}(t), & \text{otherwise}, \end{cases} \quad (10)$$

with $\mu_{bu}(t) = BW_b \log_2(1 + \text{SINR}_{bu}(t))$ the unicast real-time rate and $\eta_{buc_f}(t)$ the binary scheduling variable for chunk $c_f$ to user $u$ from base station $b$. For the proactive multicast case with $f_r < f \leq f_p$, the SBS will adapt its rate to match that of the worst user in the cluster $C_k^f$ to whom user $u$ has been assigned for the frame index at hand[6]. This way it guarantees that the chunk will be correctly decoded by all the interested cluster-users. Moreover, we remark here that the value of $\mu_{uc_f}(t)$ in (10) is bounded above by a maximum achievable service rate $\mu_{max}$ that in practice is determined by the highest available modulation and coding scheme (MCS) index.

For notational compactness, to express that a requested chunk $c_f$ corresponds either to the user's real-time FoV or to the user's cluster-level predicted FoV, we will hereafter denote the targeted FoV chunk set as $\widetilde{\mathcal{N}}_u^f = \mathbb{I}_{\{f=f_r\}} \mathcal{N}_u^f + (1 - \mathbb{I}_{\{f=f_r\}}) \widehat{\mathcal{N}}_{C_k}^f$. Then $q_u(t)$ –namely, the traffic queue of a user $u$, $\forall u \in C_k^f$ evolves as

$$q_u(t+1) = \left[ q_u(t) - \sum_{f=[f_r,f_p]} \sum_{c_f \in \widetilde{\mathcal{N}}_u^f} \mu_{uc_f}(t) \right]^+ + r_u(t). \quad (11)$$

We remark here that although only chunks for frame indexes $f = \{f_r, f_p\}$ are admitted, the range of frame indexes corresponding to chunks co-existing in a user's queue at a given time may span to values $f = [f_r, f_r + 1, \cdots, f_p]$. A scheduling policy, that is aware of the VR specific latency-reliability constraints, will timely determine which chunks need be expedited from these priority-based queues. Without loss of generality, we assume that the video streaming and FoV chunk scheduling start simultaneously, and denote the transmission delay of the video frame $f = f_r$ to user $u$ as $\tau_{uf}(t)$. Let $d_{\text{t2MTP}}(t)$ represent the available time to schedule the frame before the considered MTP delay deadline is exceeded and given by

$$d_{\text{t2MTP}}(t) = [t_a + \tau_{\text{MTP}} - t]^+, \quad (12)$$

where $t_a$ corresponds to the timestamp when the chunk was requested, and $\tau_{\text{MTP}}$ is the constant MTP latency-aware[7] maximum transmission latency. In this regard, the following HRLLBB constraint is imposed to ensure that the transmission delay of the current playing frame does not exceed the MTP delay with high probability:

$$\lim_{T \to \infty} \frac{1}{T} \sum_{t=1}^T \Pr(\tau_{uf_r}(t) \geq d_{\text{t2MTP}}(t)) \leq \epsilon_d, \quad (13)$$

where $\epsilon_d \ll 1$ is a predefined delay reliability metric. We then recast the probability in (13) as the expectation over an indicator function, i.e., the constraint is rewritten as:

$$\lim_{T \to \infty} \frac{1}{T} \sum_{t=1}^T \mathbb{E}[\mathbb{I}_{\{\tau_{uf_r}(t) \geq d_{\text{t2MTP}}(t)\}}] \leq \epsilon_d. \quad (14)$$

Collecting the HD frame admission and the binary scheduling variables as $A(t) = \{a_{uf} : \forall u \in \mathcal{U}, \forall f \in \{f_p, f_r\}\}$ and $\eta(t) = \{\eta_{buc_f}(t) : \forall b \in \mathcal{B}, \forall u \in \mathcal{U}, \forall c_f \in \widetilde{\mathcal{N}}_u^f\}$, our optimization problem is to maximize the user's quality by optimizing the HD frame admission and scheduling policies subject to the latency constraints:

**OP:** $\max_{\eta(t), A(t)} U(\{\overline{r_u}\}) = \sum_{u \in \mathcal{U}} (f(\overline{r_u}))$      (15a)

s.t.    $\overline{q_u} \leq \infty, \ \forall u \in \mathcal{U},$      (15b)

$a_{uf}(t) \in \{0, 1\}, \ \forall u \in \mathcal{U}, \forall f \in \{f_p, f_r\},$      (15c)

$r_u(t) \leq r_{max}, \ \forall u \in \mathcal{U},$      (15d)

$\mu_{uc_f}(t) \leq \mu_{max}, \ \forall u \in \mathcal{U}, \forall c_f \in \widetilde{\mathcal{N}}_u^f, \forall f \in \mathcal{F},$      (15e)

$\eta_{buc_f}(t) \in \{0, 1\}, \ \forall b \in \mathcal{B}, \forall u \in \mathcal{U}, \forall c_f \in \widetilde{\mathcal{N}}_u^f, \forall f \in \mathcal{F},$      (15f)

$\lim_{T \to \infty} \frac{1}{T} \sum_{t=1}^T \mathbb{E}[\mathbb{I}_{\{\tau_{uf_r}(t) \geq d_{\text{t2MTP}}(t)\}}] \leq \epsilon_d, \forall u \in \mathcal{U}.$    (15g)

---

[7]To realize a mobile VR operating within MTP bounds, a holistic approach that considers transmission, computing, and control latency performance should be considered. Yet, 360° videos require substantially higher data rates to provide acceptable quality to users and pose a greater challenge on the content delivery process. Hence, in this work we argue that the question of how this high data rate transmission can be realized with such latency constraint is on itself important to study and focus on the transmission delay saving a $d_f - \tau_{\text{MTP}}$ budget for other edge-server computing or on-HMD processing delays.

[6]In practice, this is accomplished by adapting the MCS that reflects users' perceived channel quality.

where $f(\cdot)$ is a non-decreasing and concave function that can be adjusted to reflect different optimization objectives.

To find a tractable solution for the above stochastic optimization problem, we first define a set of auxiliary variables $\{\gamma_u(t)\}, \forall u \in \mathcal{U}$. Accordingly, the stochastic optimization problem in (15) can be transformed from a utility function of time-averaged variables into an equivalent optimization problem of time-averaged utility function of instantaneous variables:

$$\textbf{EOP:} \quad \max_{A(t), \eta(t), \{\gamma_u(t)\}} \overline{U(\{\gamma_u(t)\})} = \overline{\sum_{u \in \mathcal{U}} \left(f(\gamma_u)\right)}$$

$$\text{s.t.} \quad \overline{\gamma_u} \le \overline{r_u}, \ \forall u \in \mathcal{U}, \tag{16a}$$

$$\gamma_u(t) \le r_{\max}, \ \forall u \in \mathcal{U}, \tag{16b}$$

$$(15b) - (15g) \tag{16c}$$

Next, by invoking the framework of Lyapunov optimization [37], *virtual queues* are constructed to help satisfy the time-average inequality constraints. By ensuring that these queues are stable, the time average constraints, namely (15g) and (16a), are guaranteed to be met. Therefore, we define $z_u(t)$ and $j_{uf}(t)$ virtual queues that correspond to the constraints over the auxiliary variables and over the transmission delay, respectively. Accordingly, the virtual queues are updated as follows:

$$z_u(t+1) = \left[z_u(t) - r_u(t) + \gamma_u(t)\right]^+, \tag{17}$$

$$j_{uf}(t+1) = \left[j_{uf}(t) + (\mathbb{I}_{\{\tau_{ufr}(t) \ge d_{\text{t2MTP}}(t)\}} - \epsilon_d) q_u(t+1)\right]^+. \tag{18}$$

Notice that the virtual queue in (18) is built after having scaled-up the constraint in (14) by multiplying both sides of it with the actual queue size. Hereinafter, for readability reasons $\mathbb{I}_{\{\tau_{uc_f}(t) \ge d_{\text{t2MTP}}(t)\}}$ will be shortened to $\mathbb{I}_{\{d_{\text{t2MTP}}(t)\}}$ to denote $\tau_{uc_f}$ exceeding the MTP delay.

Let $\chi(t) = \{q_u(t), z_u(t), j_{uf}(t) : u \in \mathcal{U}, f \in \mathcal{F}\}$ be the vector of combined traffic and virtual queues with $\chi(t) = [\chi_u(t)]_{u \in \mathcal{U}}$. Then, to represent a scalar metric of the congestion, let the quadratic Lyapunov function be given by

$$L(\chi(t)) \triangleq \frac{1}{2} \sum_{u \in \mathcal{U}} q_u(t)^2 + \frac{1}{2} \sum_{u \in \mathcal{U}} z_u(t)^2 + \frac{1}{2} \sum_{u \in \mathcal{U}} \sum_{f \in \mathcal{F}} j_{uf}(t)^2, \tag{19}$$

and the one-timeslot Lyapunov drift function be $\Delta L_t = L(\chi(t+1)) - L(\chi(t))$. Hence, we leverage the drift-plus-penalty algorithm to find the control actions that greedily minimize a bound of the drift function minus a scaled utility function, i.e., $\Delta L_t - V_\Delta \mathbb{E}\{U(\{\gamma_u(t)\})\}$, where $V_\Delta$ is the parameter that controls the trade-off between minimizing the queue backlog and approaching the optimal solution.

**Lemma 1.** *At each time instant t, the following bound satisfies the drift-plus-penalty function* $\Delta L_t - V_\Delta \mathbb{E}\{U(\{\gamma_u(t)\})\}$ *under any queue state and control strategy:*

$$\Delta L_t - V_\Delta \mathbb{E}\{U(\{\gamma_u(t)\})\} \le \Delta_0(t)$$

$$- \sum_{u \in \mathcal{U}} \left[z_u(t)\gamma_u(t) - V_\Delta U(\{\gamma_u(t)\})\right]_{\#1}$$

$$- \sum_{u \in \mathcal{U}} \left[\left(\alpha_u(t) - z_u(t)\right)r_u(t)\right]_{\#2}$$

$$- \sum_{u \in \mathcal{U}} \left[\alpha_u(t) \sum_{f \in \mathcal{F}} \sum_{c_f \in \widetilde{\mathcal{N}}_u^f} \mu_{uc_f}\right]_{\#3}, \tag{20}$$

*where $\Delta_0(t)$ is an upperbounded constant parameter at each time slot t in (20), and $\alpha_u(t)$ collects the terms related to the traffic queue and to the transmission delay virtual queue as*

$$\alpha_u(t) = \alpha_q(t) + \mathbb{I}_{\{d_{\text{t2MTP}}(t)\}}\alpha_j(t), \tag{21}$$

*which are given as $\alpha_q(t) = q_u(t)(1 + \epsilon_d^2) - \epsilon_d \sum_{f \in \mathcal{F}} j_{uf}(t)$ and $\alpha_j(t) = \sum_{f \in \mathcal{F}} j_{uf}(t) + (1 - 2\epsilon_d)q_u(t)$.*

*Proof:* See Appendix A ∎

The solution to (16) can be found by greedily minimizing the right-hand side of (20) for each time slot. Instead, since the optimization variables are decoupled in (20), we split the optimization problem into three disjoint subproblems that are solved concurrently based on the observation of the traffic and the virtual queues.

It can be also seen from (20) that the optimization problem at a given time instant $t$ is only function of the current states of the traffic and the virtual queues. This means that a solution to the problem does not require the knowledge of the traffic statistics or queue evolution probabilities. The equation also shows that the *separability criteria* [38] is met i.e., that the admission and scheduling are not dependent on the variables of each other.

### A. Auxiliary Variable Selection

The first subproblem is the minimization of the term #1 in (20) i.e., the selection of the auxiliary variables. The problem can be decoupled on a per user basis as follows:

$$\textbf{OSP1:} \quad \max_{\{\gamma_u\}} V_\Delta U(\gamma_u(t)) - z_u(t)\gamma_u(t) \tag{22a}$$

$$\text{s.t.} \quad \gamma_u(t) \le r_{\max}, \ \forall u \in \mathcal{U}. \tag{22b}$$

By selecting a linear utility function, i.e., $U(\gamma_u(t)) = \gamma_u(t)$, the optimal value of the auxiliary variable is found to be:

$$\gamma_u(t) = \begin{cases} r_{\max}, & z_u(t) \le V_\Delta, \\ 0, & \text{otherwise.} \end{cases} \tag{23}$$

It is worth noting here that different utility functions can be selected depending on the network optimization objective. For example, a logarithmic function of the admission rates could be selected to provide proportional fairness between users.

### B. HD Streaming Admission

Next, the HD chunk admission problem is optimized by solving the subproblem given by the term #2 of (20). The optimization subproblem is formulated as:

$$\textbf{OSP2:} \quad \max_{A(t)} \sum_{u \in \mathcal{U}} \left(z_u(t) - \alpha_u(t)\right)r_u(t) \tag{24a}$$

$$\text{s.t.} \quad a_{uf}(t) \in \{0, 1\}, \ \forall u \in \mathcal{U}, \forall f \in \{f_p, f_r\}, \tag{24b}$$

The above admission rate maximization problem is convex and its optimal solution is:

$$a_{uf}(t) = \begin{cases} 1 & z_u(t) \ge \alpha_u(t), \\ 0 & \text{otherwise.} \end{cases} \tag{25}$$

In other words, the optimal HD chunk admission control is to either admit or discard the whole frame, depending on the physical and virtual queue state.

### C. User-SBS Chunk Scheduling

The third subproblem aims at scheduling user requests of HD video chunks to base stations. The optimization subproblem is formulated by maximizing the term #3 in (20) as follows:

$$\textbf{OSP3:} \max_{\boldsymbol{\eta}(t)} \sum_{u \in \mathcal{U}} \alpha_u(t) \sum_{f=\{f_r,f_p\}} \sum_{c_f \in \widetilde{\mathcal{N}}_u^f} \mu_{uc_f}(t) \tag{26a}$$

$$\text{s.t. } \mu_{uc_f}(t) \leq \mu_{\max}, \forall c_f \in \widetilde{\mathcal{N}}_u^f, \forall f \in \mathcal{F}, \forall u \in \mathcal{U}, \tag{26b}$$

$$\eta_{buc_f}(t) \in \{0,1\}, \forall b \in \mathcal{B}, \forall u \in \mathcal{U}, \forall c_f \in \widetilde{\mathcal{N}}_u^f, \forall f \in \mathcal{F}. \tag{26c}$$

We emphasize that OSP3 is a combinatorial problem where video chunks for users need to be scheduled by SBSs using a mmWave multicast transmission. Subsequently, a matching algorithm is designed to associate chunk scheduling requests arising either from clusters of users or from individual users to the set of SBSs operating in mmWave band in the theater.

## IV. A MATCHING THEORY APPROACH TO HD CHUNK SCHEDULING

The use of Matching Theory [39] –a mathematical framework from labor economics that attempts to describe the formation of mutually beneficial relationships over time–, has recently garnered a considerable interest in the context of resource allocation for wireless networks [40]. However, for the sake of completeness we will first provide several definitions to properly address the fundamentals of this framework adapted to the problem at hand. Then, we will formulate the utility functions that lie at its core for both sets of agents.

### A. Matching Theory Preliminaries

**Definition 1.** *A matching game is defined by two sets of players* $(C, \mathcal{B})$ *and two preference profiles denoted by* $\succ_{\mathcal{B}}$ *and* $\succ_C$, *allowing each player* $b \in \mathcal{B}$, $C_k \in C$ *to accordingly rank the players in the opposite set.*

**Definition 2.** *The output of a matching game is a matching function* $\boldsymbol{\Upsilon}(t) = \{\Upsilon_{b,C_k}(t)\}$ *that bilaterally assigns players* $\Upsilon_b(t) \triangleq \{b \in \mathcal{B} : \Upsilon_{b,C_k}(t) = 1\}$ *and* $\Upsilon_{C_k}(t) \triangleq \{C_k \in C : \Upsilon_{b,C_k}(t) = 1\}$ *such that* $|\Upsilon_{C_k}(t)| \leq q_C$ *and* $|\Upsilon_b(t)| \leq q_{\mathcal{B}}$ *are fulfilled, with* $q_{\mathcal{B}}$, $q_C$ *the quota of the players which, for a one-to-one matching game satisfy* $q_{\mathcal{B}} = q_C = 1$.

**Definition 3.** *A preference* $\succ$ *is a complete, reflexive and transitive binary relation between the players in* $\mathcal{B}$ *and* $C$. *Therefore, for any SBS* $b \in \mathcal{B}$ *a preference relation* $\succ_b$ *is defined over the set of clusters* $C$ *such that for any two clusters* $(C_k, C_k') \in C \times C$ *with* $C_k \neq C_k'$, *and two matchings* $\boldsymbol{\Upsilon}(t)$, $\boldsymbol{\Upsilon}'(t)$ *so that* $\Upsilon_b(t) = C_k$ *and* $\Upsilon_b'(t) = C_k'$:

$$(C_k, \boldsymbol{\Upsilon}(t)) \succ_b (C_k', \boldsymbol{\Upsilon}'(t)) \Leftrightarrow U_{\mathcal{B}}^{b,C_k}(t) > U_{\mathcal{B}}^{b,C_k'}(t). \tag{27}$$

*Similarly, for any cluster of users* $C_k \in C$ *a preference relation* $\succ_{C_k}$ *is defined over the set of SBS* $\mathcal{B}$ *such that for any two*

SBSs $(b, b') \in \mathcal{B} \times \mathcal{B}$ *with* $b \neq b'$, *and two matchings* $\boldsymbol{\Upsilon}(t)$, $\boldsymbol{\Upsilon}'(t)$ *we have that* $\Upsilon_{C_k}(t) = b$ *and* $\Upsilon'_{C_k}(t) = b'$:

$$(b, \boldsymbol{\Upsilon}(t)) \succ_{C_k} (b', \boldsymbol{\Upsilon}'(t)) \Leftrightarrow U_C^{C_k,b}(t) > U_C^{C_k,b'}(t), \tag{28}$$

*where* $U_{\mathcal{B}}^{b,C_k}(t)$ *and* $U_C^{C_k,b}(t)$ *denote the utility of cluster* $C_k$ *for SBS* $b$ *and the utility of SBS* $b$ *for cluster* $C_k$, *correspondingly.*

### B. Matching Utility Formulation

The HD chunk scheduling subproblem in (26) is formulated as a matching game between the SBSs and the clusters of users. As such, both sides seek to greedily maximize the overall VR experience by efficiently allocating the mmWave transmission resources while VR QoE related constraints are met. Hence, each timeslot with updated information on channel and queue state, new scheduling requests for video chunk transmission will be prioritized in each cluster and in each SBS, and new sets of matching pairs will be found using the proposed approach. With the above principles in mind, we formulate the utilities for both sets.

The utility of serving a given cluster of users with at least one pending chunk request from the SBSs point of view will essentially reflect two aspects: the *priority* and *relevance* of the chunk $c_f$ at hand. The priority of the whole frame to which the requested chunk belongs to is controlled by the dynamics of $q_u(t)$ and $j_{uf}(t)$ as per (11) and (18) through $d_{\text{t2MTP}}(t)$ as given by (12). The relevance of the chunk within the cluster FoV is related to its popularity i.e., how many of the cluster members have requested this chunk. Intuitively, the cluster-level multicast approach decreases the wireless network load by transmitting each chunk falling into the cluster-level FoV only once. Moreover, transmitting first the most relevant chunks also contributes to increasing the overall system welfare. Therefore, SBSs will build their preference profile using the following utility function:

$$U_{\mathcal{B}}^{b,C_k}(t) = \sum_{u \in C_k} \mathbb{I}_{\{c_f \in \widetilde{\mathcal{N}}_u^f\}} \alpha_u(t) \tag{29}$$

$$= \sum_{u \in C_k} \mathbb{I}_{\{c_f \in \widetilde{\mathcal{N}}_u^f\}} \{\alpha_q(t) + \mathbb{I}_{\{d_{\text{t2MTP}}(t)\}} \alpha_j(t)\}.$$

Notice that in (29) by definition, $\mathbb{I}_{\{d_{\text{t2MTP}}(t)\}}$ can only be non-zero for the currently playing frame index. Similarly, the utility of a SBS from the clusters' perspective will depend on the goodness of the transmission opportunity through the offered rate in (10). In other words, we define the utility as

$$U_C^{C_k,b}(t) = \mathbb{I}_{\{f=f_r\}} \min_{\forall u \in C_k^f | c_f \in \mathcal{N}_u^f} \mu_{bu}(t)$$

$$+ (1 - \mathbb{I}_{\{f=f_r\}}) \min_{\forall u \in C_k^f | c_f \in \widetilde{\mathcal{N}}_u^f} \mu_{bu}(t), \tag{30}$$

### C. Stability of the Matching

Next, the notion of stability is introduced and an interference estimation method is proposed to guarantee that the HD chunk scheduling game converges to a stable matching.

**Definition 4.** *Given a matching* $\boldsymbol{\Upsilon}$ *with* $\Upsilon_b = C_k$ *and* $\Upsilon_{C_k} = b$, *and a pair* $(b', C_k')$ *with* $\Upsilon_b(t) \neq k'$ *and* $\Upsilon_{C_k} \neq b'$, $(b', k')$ *is*

*said to be blocking the matching $\Upsilon$ and form a blocking pair if: 1) $b' \succ_k b$, 2) $k' \succ_b k$. A matching $\Upsilon*$ is stable if there is no blocking pair.*

Gale-Shapley's deferred acceptance (DA) algorithm [41] provides a polynomial time solution guaranteed to be two-sided stable for one-to-one canonical matchings i.e., those matching games where the preference profiles of the players are not affected by any other player's decisions. The influence of a given player's matching over another's is referred to as *externality*. As the game evolves, the existence of externalities triggers dynamic updates in the values of the perceived utilities and, consequently, ensuring the stability of the matching is challenging.

The above matching game cannot be directly solved using DA; the utilities in (29)-(30) are function of the instantaneous service rate which, in turn depends on the interference –a well-known source of externalities– through the SINR. Moreover, in the context of directional communications, the arrival direction of the interference caused by other SBSs[8] greatly impacts the service rate. Hence, evaluating the instantaneous interference and casting preferences accordingly implies full knowledge of the system-wide current matching state by all the players, which is impractical in terms of signaling overhead. Different approaches are used in the literature to handle the matching externalities either through finding preference profiles that do not induce externality [42], or through centralized approval methods that evaluate how each matching change affects other players before allowing it [43]. Alternatively, we consider a distributed and computationally efficient algorithm that suits the low latency nature of the considered application by replacing the instantaneous values of the service rate in the utilities with estimated ones.

Let the measured inter-SBS interference at user $u$ in the previous time instant $t-1$ be denoted as $I_u(t-1)$, and $\hat{I}_u(t)$ the estimated inter-SBS interference at time instant $t$. Adopting an interference estimation procedure with learning parameter $\nu_1$ and moving average inference $\tilde{I}_u^{\gamma_2}(t-1)$ with a window of $\nu_2$ samples, the estimated interference is given by

$$\hat{I}_u(t) = \nu_1 I_u(t-1) + (1-\nu_1)\tilde{I}_u^{\gamma_2}(t-1). \tag{31}$$

Let $\hat{U}_{\mathcal{B}}^{b,C_k}(t)$, $\hat{U}_C^{C_k,b}(t)$ be the new expressions for the utilities which exploit the estimated service rate $\hat{\mu}_{bu}(t) = BW_b \log_2\left(1 + \frac{p_b h_{bu}(t) g_{bu}^{\mathrm{Rx}}(t) g_{bu}^{\mathrm{Tx}}(t)}{\hat{I}_u(t) + BW_b N_0}(t)\right)$ through $\hat{\mu}_{uc_f}(t)$, such that

$$\hat{U}_C^{C_k,b}(t) = \mathbb{I}_{\{f=f_r\}} \min_{\forall u \in C_k^f \mid c_f \in \mathcal{N}_u^f} \hat{\mu}_{bu}(t)$$
$$+ (1 - \mathbb{I}_{\{f=f_r\}}) \min_{\forall u \in C_k^f \mid c_f \in \widehat{\mathcal{N}}_u^f} \hat{\mu}_{bu}(t), \tag{32}$$

$$\hat{U}_{\mathcal{B}}^{b,C_k}(t) = \sum_{u \in C_k} \mathbb{I}_{\{c_f \in \widetilde{\mathcal{N}}_u^f\}} \hat{\alpha}_u(t)(t)$$
$$= \sum_{u \in C_k} \mathbb{I}_{\{c_f \in \widetilde{\mathcal{N}}_u^f\}} \left\{ \hat{\alpha}_Q(t) + \mathbb{I}_{\{d_{\mathrm{t2MTP}}(t)\}} \hat{\alpha}_F(t) \right\}. \tag{33}$$

[8]We remark here that by matching each SBS to a single cluster with orthogonal non-overlapping beams for the multicast transmission, only the impairment due to inter-SBS interference needs to be considered.

---

**Algorithm 1:** HD chunk scheduling between SBSs and User-clusters

**Phase I - Interference learning and candidate chunk selection**
- Each $u \in \mathcal{U}$, updates $\hat{I}_u(t)$ as per (31) and reports channel in the UL.
- In the edge controller, queues in $\{\chi(t)\}_{u \in \mathcal{U}}$ are updated by solving (22), (24).
- For each $C_k \in C$ a cluster-level chunk request pool is created and each request therein is assigned an urgency tag $\alpha_{C_k}^{c_f} = \sum_{u \in C_k \mid c_f \in \mathcal{N}_u^f} \alpha_u(t)$ with $\alpha_u(t)$ as per (21).Then, the request pool is sorted in descending order of $\alpha_{C_k}^{c_f}$.

**Phase II - Matching game construction**
- Each cluster $C_k \in C$, updates $\hat{U}_C^{C_k,b}$ over the SBSs in $\mathcal{B}$ as per (32).
- Each SBS $b \in \mathcal{B}$, updates $\hat{U}_{\mathcal{B}}^{b,C_k}$ over $\{C_k\}_{k=1}^K$ as per (33) evaluating the cluster utility by its most urgent chunk-request, i.e. by $\max\{\alpha_{C_k}^{c_f}\}$.

**Phase III - Deferred Acceptance for SBS-Cluster allocation**
- For each SBS $b$, initialize the subset of eligible clusters, $\mathcal{E}_C^b \subseteq C$ so that initially $|\mathcal{E}_C^b| = |C|$.
- For each SBS $b$, each cluster $C_k$, initialize the subset of unmatched clusters $\mathcal{S}_C \subseteq C$ and SBS $\mathcal{S}_{\mathcal{B}} \subseteq \mathcal{B}$, so that initially $|\mathcal{S}_C| = |C|$, $|\mathcal{S}_{\mathcal{B}}| = |\mathcal{B}|$.

**while** $|\mathcal{S}_{\mathcal{B}}| \neq \emptyset$ and $\sum_{b \in \mathcal{B}} |\mathcal{E}_C^b| \neq \emptyset$ **do**

  Pick a random SBS $b \in \mathcal{B}$;
  **if** $|\mathcal{E}_C^b| \neq \emptyset$ **then**
    SBS $b$ sends a chunk scheduling proposal to its best ranked *eligible* cluster $C_k$, $C_k \in \mathcal{E}_C^b$;
    **if** $C_k \in \mathcal{S}_C$ **then**
      Match $b$ and $C_k$ setting $\Upsilon_b(t) = C_k$ and $\Upsilon_{C_k}(t) = b$;
      Remove $b$ and $C_k$ from $\mathcal{S}_{\mathcal{B}}$ and $\mathcal{S}_C$ respectively;
    **else**
      **if** $\hat{U}_C^{C_k,b}(t) > \hat{U}_C^{C_k,\Upsilon_{C_k}(t)}(t)$ **then**
        Reject proposal from $\Upsilon_{C_k}(t)$; add back $\Upsilon_{C_k}(t)$ to $\mathcal{S}_{\mathcal{B}}$ and remove $C_k$ from $\mathcal{E}_C^{\Upsilon_{C_k}(t)}$;
        Match $b$ and $C_k$ setting $\Upsilon_b(t) = C_k$ and $\Upsilon_{C_k}(t) = b$;
        Remove $b$ from $\mathcal{S}_{\mathcal{B}}$;
      **else**
        Refuse proposal from $b$;
        Remove $C_k$ from $\mathcal{E}_C^b$;
      **end**
    **end**
  **end**
**end**

**Phase IV - Stable matching**

---

Under this new utility formulation there are no longer externalities in the system. Therefore, the HD chunk scheduling matching game can be solved using DA, which is guaranteed to converge to a stable matching $\Upsilon(t)$ once the inter-cell interference learning process converges. The process described above as well as the details of the matching rounds are described in Algorithm 1.

## V. DRNN FoV Prediction and FoV+Location Aware User Clustering

In this section, the DRNN that predicts VR users' FoVs for upcoming video frames and the clustering scheme that leverages FoV and spatial inter-user correlations are described. We first motivate the selection of the adopted sequential learning model and briefly summarize its operation dynamics. Following that, the DRNN architecture implementation details are provided and the training process is explained. Finally, the distance metric driving the user-clustering partitioning and its algorithmic implementation are specified.

### A. Sequential Deep Learning Model Operation

Predicting a VR user's tiled-FoV is an instance of movement prediction where an input sequence of a user's past and current pose vectors is mapped to a multi-label output. In the output,

each label represents one tile in the video frame and its value provides an estimate over the likelihood of the tile belonging to the user's future FoV.

To build our sequential learning model we adopt the GRU [44] architecture, a variant of RNNs that uses two simple gating mechanisms whereby long-term dependencies are effectively tackled and the memory/state from previous activations is preserved. Hence, compared to other models such as long short-term memory (LSTM) units [45], GRUs are faster to train and have proven to perform better for small datasets [46], the case considered in Section VI-A. Specifically, for every operation time step –which is measured in terms of video frames and therefore indexed with $f \in \mathcal{F}$– the GRU units update the value of their hidden state $h_f$ as a non-linear function of an input sequence $x_u^f$ and of the previous hidden state $h_{f-1}$. The non-linear function is parameterized by $\theta$ following a recurrence relation $h_f = f(h_{f-1}, x_u^f; \theta)$ that is visually sketched in Fig.4 and formally described by the following model equations:

$$\Gamma_f = \sigma(W_\Gamma x_u^f + Z_\Gamma h_{f-1} + b_\Gamma) \quad (34)$$

$$r_f = \sigma(W_r x_u^f + Z_r h_{f-1} + b_r) \quad (35)$$

$$h_f = (1 - \Gamma_f) \circledast h_{f-1}$$
$$+ \Gamma_f \circledast \tanh(W x_u^f + Z(r_f \circledast h_{f-1}) + b_h), \quad (36)$$

where weight matrices $W_\Gamma$, $Z_\Gamma$, $W_r$, $Z_r$, $W$, $Z$ and bias terms $b_\Gamma$, $b_r$, $b_h$ represent the model parameters comprised in $\theta$ that, with those of the fully connected neural layer in Fig. 5, are learned during the DRNN offline training process.

The value of the update gate vector $\Gamma_f$, as per (34), governs through the linear interpolation in (36) the amount of the previous hidden state $h_{f-1}$ and of the new hidden state candidate $\tilde{h}_f = \tanh(W x_u^f + Z(r_f \circledast h_{f-1}) + b_h)$ contributing to the next hidden state activation $h_f$. Likewise, the reset gate vector $r_f$, as per (35), controls the degree of the contribution of the previous hidden state $h_{f-1}$ preserved for the new hidden state candidate $\tilde{h}_f$. When the contribution from the previous state is deemed irrelevant, the next hidden state $\tilde{h}_f$ is reset and will depend only on the input sequence.

### B. DRNN architecture

The building blocks of our proposed deep recurrent learning model $M_\theta^{v,T_H}$ based on GRU layers and implemented using Keras [47], a high-level neural networks API running on top of a Tensorflow backend, are represented in Fig. 5.

**Input representation:** Every $T_f$, an input sequence of size $T_P$ corresponding to 3DoF pose vectors $x_u^f \triangleq \{p_{3u}^f\}_{f=f_r-T_P+1}^{f_r}$ is fed to the first GRU layer.

**Sequence processing:** The input is then processed following model equations (34)-(36) in Section V-A by a $T_P$ time-step GRU cell with a hidden state size equal to 512 examples. Following a rectified linear unit (ReLU) activation that performs a $[z]^+$ operation, the output of the first GRU layer goes through a second GRU layer with the same characteristics. The output state from this second layer $h_f^{(2)} \triangleq o_f^{(2)}$ is then fed to a serial to parallel (S/P) layer before going across a dense neural layer that connects with the $N$ output neurons.
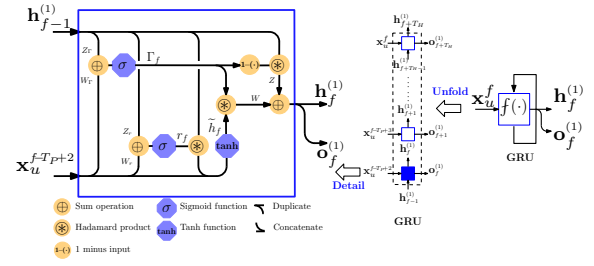


Figure 4. Detailed graphical representation of GRU unfolding and of the $h_f^{(1)}$ computation in the unfolded GRU cell. The notation $(\cdot)^{(1)}$ indicates that the GRU at hand belongs to the first layer, which is highlighted in blue in the DRNN architecture from Fig. 5.
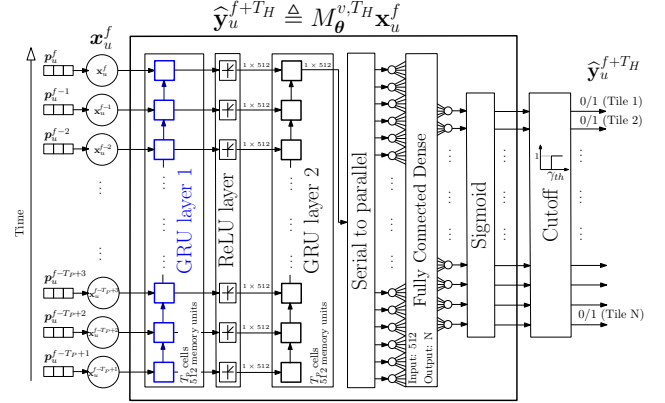


Figure 5. Block diagram of the deep learning model for the edge controller. The DRNN predicts the tiles in the FoV of user $u$ at frame index $f_P = f_r + T_H$, i.e. $T_H$ frames ahead.

**Output representation:** Given the multi-label nature of our learning model, a sigmoid activation layer is used to map the $N$ sized dense output into $N$ probability values or logits $\{\Pr(n) = \sigma(W_d h_f^{(2)} + b_d)_n\}_{n=1}^N$ that are Bernoulli distributed, i.e., the probability of each label is treated as independent from other labels' probabilities. The output of the sigmoid is then binarized with a cutoff layer such that

$$\widehat{y}_{u,n}^{f_P} = \begin{cases} 1, & \sigma(W_d h_f^{(2)} + b_d)_n \geq \gamma_{th}, \\ 0, & \text{otherwise,} \end{cases} \quad (37)$$

where $W_d$, $b_d$ are the weights and biases of the dense fully-connected layer and $\gamma_{th}$ is the threshold value for the cutoff layer, which is chosen to balance accuracy and recall. After the binarization, the predicted FoV for a user $u$ and frame index $f_P = f + T_H$ is retrieved as $\widehat{\mathcal{N}}_u^{f_P} = \{n \in [1, ..., N]: \widehat{y}_{u,n}^{f_P} = 1\}$.

### C. DRNN Training

The aim of the training in the supervised deep recurrent learning model $M_\theta^{v,T_H}$ is to iteratively find the $\theta$ parameters that minimize a binary cross-entropy loss function $\mathcal{L}(\theta)$ for all training instances. This loss function, for model parameters $\theta$, labels $y_{u_{tr,n}^v}^{f_P}$ and logits $\{\Pr(n)\}_{n=1}^N$ captured from the output of the sigmoid layer in Fig. 5, is expressed as

$$\mathcal{L}(\theta) = -\frac{1}{N} \sum_{n=1}^N \left[ y_{u_{tr,n}^v}^{f_P} \log(\Pr(n)) + (1 - y_{u_{tr,n}^v}^{f_P}) \log(1 - \Pr(n)) \right]. \quad (38)$$

During the model offline training, Backpropagation Through Time (BPTT) algorithm [48] and Adam algorithm [49] are used to optimize the gradients. Adam is set with learning rate $\alpha = 0.01$, parameters $\beta_1 = 0.9$, $\beta_2 = 0.999$ and no decay. The gradient backpropagation is performed over data batches of size 512 and during 20 training epochs.

Next, the information related to users' physical location and to their predicted FoVs is leveraged to develop a user clustering scheme.

### D. Proposed FoV and Location Aware User Clustering

Once the predictions of the entire set of users $\mathcal{U}$ for a frame index $f_P$ are ready, users viewing the same video $v$ are grouped into clusters based on their spatial and content correlations.

Mathematically, let $C_k^{f_P}$ denote the $k$-th cluster in which the set of users $\mathcal{U}$ is partitioned for frame index $f_P = f + T_H$ such that $\bigcup_{k=1}^{K} C_k^{f_P} = \mathcal{U}$. Here, the cluster partitioning can be obtained by i) computing the $|\mathcal{U}| \times |\mathcal{U}|$ distance matrix $\boldsymbol{D}^{f_P}$, whose $d_{u,u'}^{f_P} = \tilde{d}_{u,u'}^{f_P} \left( d_{uu'}^{2D} / d_{min}^{2D} \right)$ element results from quantifying the FoV-related distance or dis-similarity between any pair of users $\{u, u'\} \in \mathcal{U}$ which is given by $\tilde{d}_{u,u'}^{f_P} = 1 - \sum_{n=1}^{N} \mathbb{I}_{\{n \in \widehat{\mathcal{N}}_u^{f_P}\}} \mathbb{I}_{\{n \in \widehat{\mathcal{N}}_{u'}^{f_P}\}} / (N - \sum_{n=1}^{N} \mathbb{I}_{\{n \notin \widehat{\mathcal{N}}_u^{f_P}\}} \mathbb{I}_{\{n \notin \widehat{\mathcal{N}}_{u'}^{f_P}\}})$; and ii) scaling it by their relative physical distance $d_{uu'}^{2D}$ divided by $d_{min}^{2D}$, that denotes the minimum value for such relative distance as per the theater dimensions and seat arrangements.

To implement the clustering scheme that builds on the above distance metric, a hierarchical agglomerative clustering with average linkage has been considered. This clustering scheme allows operating over the constructed dendrogram to increase/decrease the number of resulting clusters and thereby investigating the trade-offs in terms of communication resource utilization versus achieved performance when many/few clusters, as per $K$, are used.

Once the clusters $\{C_k^{f_P}\}_{k=1}^{K}$ have been estimated using the specific clustering strategy, the cluster-level FoV is built and ready to be leveraged in the proposed multicast/unicast scheduling strategy as $\widehat{\mathcal{N}}_{C_k}^{f_P} = \bigcup_{u \in C_k^{f_P}} \widehat{\mathcal{N}}_u^{f_P}$.

## VI. SIMULATION AND PERFORMANCE EVALUATION

In this section, we numerically validate the effectiveness of the proposed solution. For that purpose, we start by describing the dataset with real head-tracking information for 360° videos and the DRNN FoV prediction accuracy results which will impact the performance evaluation of the mmWave multicast transmission. Following that, the deployment scenario and the considered baseline schemes are described. Finally, the performance evaluation of the proposed approach is evaluated and some insightful results are discussed[9].

### A. 360° Video Head-tracking Dataset and DRNN FoV Prediction Accuracy Results

To validate our proposed approach, the information fed into the DRNN for training and simulation corresponds to 3DoF

[9]For the interested reader, a demo showcasing the qualitative results achieved comparing our approach to the baselines described in Section VI-B is available at https://youtu.be/djt9efjCCEw.

#### Table II
#### FOV TEST ACCURACY: EFFECT OF PREDICTION HORIZON

| Video | Category[10] | Jaccard similarity index $J_v^{T_H}$ (mean ± std. dev.) | | | |
|---|---|---|---|---|---|
| | | $T_H = 5$ | $T_H = 10$ | $T_H = 20$ | $T_H = 30$ |
| SFRSport | NI, SP | 0.70±0.06 | 0.69±0.04 | 0.63±0.03 | 0.50±0.05 |
| MegaCoaster | NI, FP | 0.68±0.06 | 0.65±0.05 | 0.64±0.07 | 0.61±0.05 |
| RollerCoaster | NI, FP | 0.74±0.05 | 0.70±0.05 | 0.64±0.04 | 0.63±0.05 |
| SharkShipwreck | NI, SP | 0.53±0.03 | 0.48±0.03 | 0.44±0.03 | 0.36±0.03 |
| Driving | NI, FP | 0.76±0.04 | 0.71±0.04 | 0.63±0.03 | 0.58±0.02 |
| ChariotRace | CG, FP | 0.71±0.02 | 0.71±0.02 | 0.68±0.02 | 0.65±0.03 |
| KangarooIsl | NI, SP | 0.69±0.04 | 0.65±0.03 | 0.63±0.03 | 0.58±0.03 |
| Pac-man | CG, FP | 0.83±0.03 | 0.73±0.05 | 0.67±0.05 | 0.66±0.06 |
| PerilsPanel | NI, SP | 0.69±0.02 | 0.65±0.02 | 0.56±0.03 | 0.53±0.03 |
| HogRider | CG, FP | 0.68±0.04 | 0.66±0.04 | 0.65±0.04 | 0.57±0.05 |

#### Table III
#### FOV TEST ACCURACY: EFFECT OF THE NUMBER OF GRU LAYERS

| Video | Jaccard similarity index $J_v^{T_H}$ (mean ± std. dev.) | | | |
|---|---|---|---|---|
| | $T_H$ | Number of GRU layers | | |
| | | 1 | 2 | 3 |
| MegaCoaster | 5 | 0.52±0.08 | 0.68±0.06 | 0.68 ± 0.04 |
| | 10 | 0.50±0.07 | 0.65±0.05 | 0.65 ± 0.03 |
| | 20 | 0.46±0.07 | 0.64±0.07 | 0.63 ± 0.05 |
| | 30 | 0.32±0.04 | 0.61±0.05 | 0.61 ± 0.06 |
| Pac-man | 5 | 0.82±0.04 | 0.83±0.03 | 0.76 ± 0.04 |
| | 10 | 0.60±0.07 | 0.73±0.05 | 0.73 ± 0.04 |
| | 20 | 0.53±0.05 | 0.67±0.05 | 0.67 ± 0.05 |
| | 30 | 0.49±0.06 | 0.66±0.06 | 0.65 ± 0.05 |

traces from the dataset in [50] whereby the pose of 50 different users while watching a catalog of $V=10$ HD 360° videos from YouTube were tracked. The selected videos are 60 s long, have 4K resolution and are encoded at 30 fps. A $100° \times 100°$ FoV is considered and, to build the tiled-FoV, the EQR projection of each of the video frames has been divided into $N = 200$ square tiles of $192 \times 192$ pixels arranged in a regular grid of $N_V = 10$ and $N_H = 20$ tiles. The dataset provides the ground-truth labels after mapping the 3DoF poses to their corresponding tiled FoVs. In view of the size and characteristics of the dataset, the original 50 users have been split into disjoint $\mathcal{U}_{tr}$ and $\mathcal{U}$ sets for training and for test purposes with cardinalities $|\mathcal{U}_{tr}| = 35$ and $|\mathcal{U}| = 15$, respectively.

Results in Table II represent the accuracy of the prediction models for different values of $T_H$ in terms of the Jaccard similarity index, which is defined for each user $u$ viewing a frame $f$ of a video $v$ as the intersection over the union between the predicted and the actual FoV tile sets $J(\widehat{\mathcal{N}}_u^f, \mathcal{N}_u^f) = |\widehat{\mathcal{N}}_u^f \cap \mathcal{N}_u^f| / |\widehat{\mathcal{N}}_u^f \cup \mathcal{N}_u^f|$. In Table II, this index has been first averaged over the frames of the video at hand, and then over all the test users, i.e., $J_v^{T_H} = \frac{1}{|\mathcal{U}||\mathcal{F}|} \sum_{u \in \mathcal{U}} \sum_{f \in \mathcal{F}} \frac{|\widehat{\mathcal{N}}_u^f \cap \mathcal{N}_u^f|}{|\widehat{\mathcal{N}}_u^f \cup \mathcal{N}_u^f|}$. The results in the table confirm the anticipated decrease of the accuracy as the prediction horizon moves further away from the last reported pose. Similarly, results in Table III show that increasing the depth of the DRNN by adding more GRU layers is counter-productive; it overfits the training data and unnecessarily increases the complexity of the model.

### B. Deployment Details and Reference Baselines

We consider two VR theater settings, a small and a medium size capacity theaters with dimensions $\{s_r, s_c\} = \{5, 10\}$ and $\{s_r, s_c\} = \{10, 15\}$ seats, respectively. In both configurations

[10]With category codes: NI=Natural Image, CG=Computer Generated, SP=Slow-paced, FP=Fast-paced.

Table IV
MAIN SIMULATION PARAMETERS

| Parameter | Value |
|---|---|
| Simulation time | 60000 ms |
| Channel coherence time ($T_c$) | 1ms |
| Blockage re-evaluation time ($T_{block}$) | 100 ms |
| Transmission slot duration ($T_t$) | 0.25 ms |
| RF chains | 1 per HMD; 4 per SBS |
| Beam-level Rx beamwidth | 5° |
| Beam-level Tx beamwidths | [5°:5°:45°] |
| Carrier frequency ($f_c$) | 28 GHz |
| Bandwidth ($BW_b$) | 0.85 GHz |
| Noise spectral density ($N_0$) | -174 dBm/Hz |
| Noise figure | 9 dB |
| SBS transmit power ($p_i$) | 15 dBm |
| Motion-to-photon delay ($\tau_{MTP}$) | 10 ms |
| Delay reliability metric $\epsilon_d$ | 0.01 |
| Tiles per frame ($N$) | 200 |
| Video frame duration ($T_f$) | 33 ms (30 fps video) |
| Videos catalog size ($V$) | [1, 3, 5, 10] |
| Users per video | [10, 15] |
| Number of clusters ($K$) | [$2 \cdot V$, $3 \cdot V$, $4 \cdot V$] |
| Prediction horizon ($T_H$) | [5, 10, 20, 30] frames |
| DRNN input sequence ($T_P$) | 30 pose values |
| DRNN cutoff value ($\gamma_{th}$) | 0.5 |

the seats are separated from each other by 2 m, and there is a 4 m distance from the seat area to the walls of the enclosure. As detailed in Section II, SBSs are located at ceiling level in the upper 4 corners of the theater. A total of 7 different scenarios are studied for simulation: scenarios sT-$v correspond to the small theater with 10 users per video with $ = V = \{1, 3, 5\}$ videos being played; scenarios bT-$v correspond to the big theater with 15 users per video with $ = V = \{1, 3, 5, 10\}$ videos being played. The set of default parameter values for simulations is provided in Table IV. For benchmarking purposes, the following baseline and proposed schemes are considered:

- UREAC: Chunk requests are scheduled in real-time for mmWave unicast transmission.
- MREAC: Chunk requests are scheduled in real-time and multi-beam mmWave multicast transmission is used.
- MPROAC: Chunk requests are proactively scheduled and multi-beam mmWave multicast transmission is used.
- MPROAC+: Corresponds to the *proposed* approach which considers MPROAC and the HRLLBB constraint in the scheduler.

### C. Discussion

Next, the impact of the FoV prediction horizon, the requested video quality, the maximum number of clusters and, the network size are evaluated. To that end, the performance of each scheme is evaluated through its average and the 99th percentile (delay 99 pctl) transmission delay, calculated as the delay until the last tile in the FoV of the requesting user has been delivered. We observe here that focusing merely on the average system performance would fail to capture the real-time aspects of the problem. Consequently, to show that the adopted Lyapunov online control method is able to keep the latency bounded below the $\tau_{MTP}$ threshold with the desired probability $1 - \epsilon$, the 99th percentile delay is provided

too. Furthermore, alongside with the 99th percentile delay plots, the HD successful delivery rate metrics highlight the trade-off between the utility (maximizing the quality) and the probabilistic latency constraint.

Lastly, for each user $u$ and frame index $f$ the Jaccard similarity index between the successfully delivered chunks and the actual FoV is computed. The index is given by $J(\check{\mathcal{N}}_u^f, \mathcal{N}_u^f) = |\check{\mathcal{N}}_u^f \cap \mathcal{N}_u^f| / |\check{\mathcal{N}}_u^f \cup \mathcal{N}_u^f|$ with $\check{\mathcal{N}}_u^f \subseteq \bigcup \{\tilde{\mathcal{N}}_{C_k}^f, \mathcal{N}_u^f\}$ denoting the set of tiles correctly decoded at user $u \in C_k^f$. We notice here that $\check{\mathcal{N}}_u^f \setminus \mathcal{N}_u^f$ represents a measure of the multicasting overhead owing to operating with cluster-level predicted FoV chunk requests that, hence, will grow larger the less correlated the FoVs of the cluster-members are. Similarly, $\mathcal{N}_u^f \setminus \check{\mathcal{N}}_u^f \neq \emptyset$ hints missed tiles. Therefore the evolution of this index, averaged over all the users and frame indices, globally captures the trade-off related to FoV correlation among cluster members.

*1) Impact of the FoV Prediction Horizon:* We first consider the impact of the DRNN prediction horizon $T_H$ on the performance of the proposed approaches MPROAC+ and MPROAC, and compare it with the reactive baselines UREAC and MREAC, whose performance is not affected. Intuitively, in our scheme longer $T_H$ allow the scheduler to schedule future frames earlier, but increases the overall amount of data to be transmitted due to having lower prediction accuracy, as shown in Table II. In Fig. 6, it can be seen that the scheduler can maintain high HD quality streaming even with long prediction horizons. The frame delay is shown to first decrease, due to having more time to schedule chunks in advance, then increases again, due to having to schedule a higher number of chunks in real-time that were missed by the predictor. Transmitting more real time leads to lower utilization of the user's predicted FoV, which decreases the Jaccard index.

*2) Impact of the Requested Video Quality:* We move on now to investigate the impact of the requested HD video quality on the performance of the proposed scheme. To that end, looking into both the sT-3v and bT-3v scenarios, we evaluate the impact of the quality through the HD *chunk size* of the frame shown to the user. The performance metrics of each scheme are depicted in Fig. 7(a)-(d) and Fig. 7(e)-(h) for the small and big theater scenarios, respectively. The figures clearly show the trade-off between frame delay and HD streaming rate. As the chunk size increases, the average and 99th percentile delays increase for the different schemes. Moreover, comparing UREAC with the other schemes, it is shown that multicasting brings $40 - 50\%$ increase in the HD rate and $33 - 70\%$ latency reduction through the utilization of common FoVs of different users. At high chunk sizes, the higher network load clearly increases the service delay. By delivering the predicted frames in advance, both the MPROAC+ and MPROAC minimize the average delay without sacrificing the HD quality rate. The proposed MPROAC+ scheme is shown to also keep the worst delay values bounded due to imposing the latency constraint, as compared to the MPROAC. Further comparing UREAC to MREAC, it is shown that multicasting significantly reduces the delay due to the utilization of common FoVs of different users.
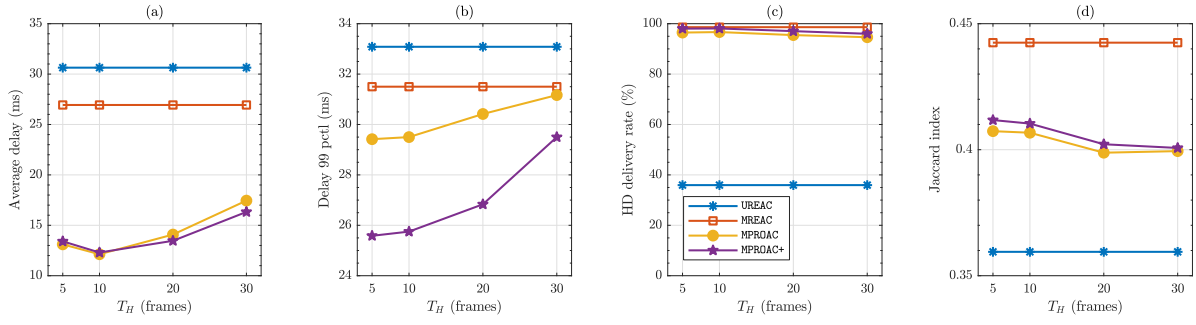
Figure 6. (a) Average delay, (b) 99th percentile delay, and (c) HD delivery rate and (d) Jaccard index performance versus the prediction horizon (in frames) in `bT-3v` with a load of 1 Gbps per user (0.972 Mb chunk size), and $V_\Delta = 1 \cdot 10^8$.
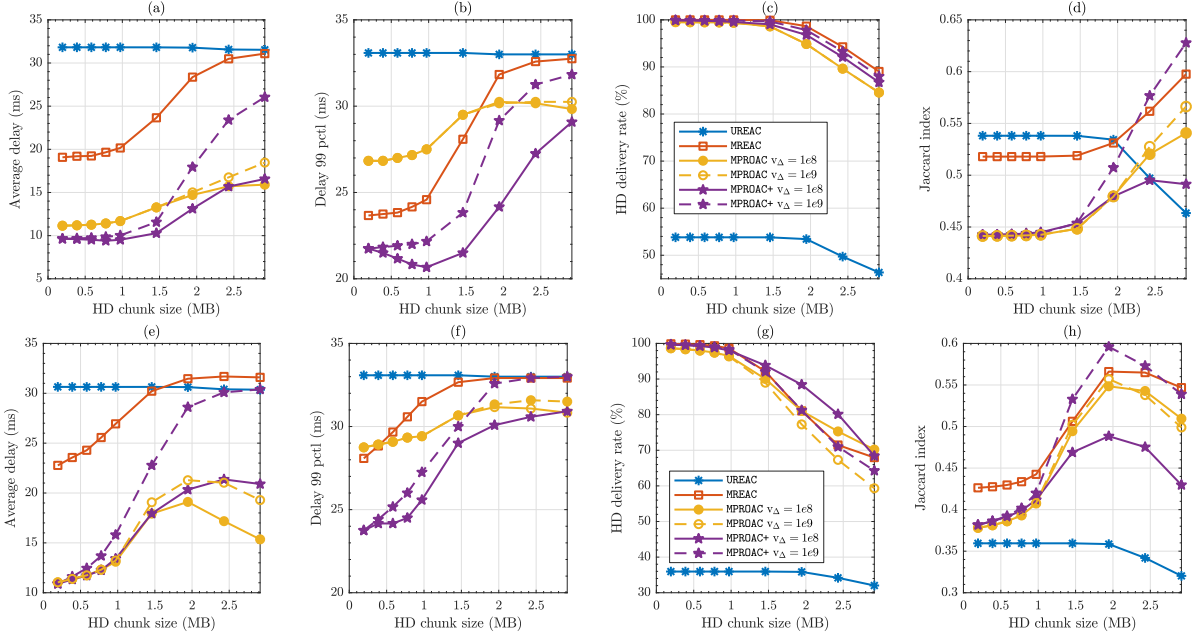


Figure 7. (a) and (e) Average delay, (b) and (f) 99th percentile delay, (c) and (g) HD delivery rate and (d) and (h) Jaccard index performance in `sT-3v` and `bT-3v`, respectively, as a function of the HD chunk size, for $V = 3$ videos, $K = 2 \times V$ clusters, $T_H = 5$ frames, and Lyapunov trade-off $V_\Delta = 1 \cdot 10^8$ and $V_\Delta = 1 \cdot 10^9$.
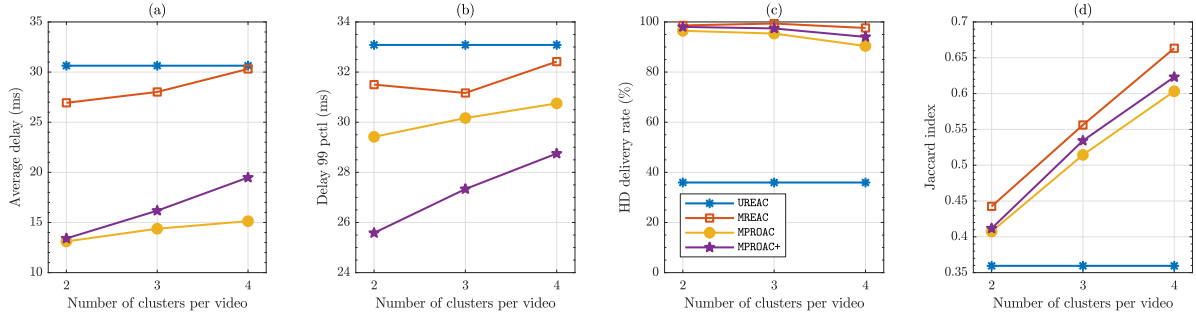


Figure 8. (a) Average delay, (b) 99th percentile delay, and (c) HD delivery rate and (d) Jaccard index performance versus cluster per video, in `bT-3v` with a load of 1 Gbps per user (0.972 Mb chunk size),, and $V_\Delta = 1 \cdot 10^8$.

As the performance of the `MPROAC+` and `MPROAC` schemes for different values of the Lyapunov parameter $V_\Delta$ are provided, the trade-off between the frame delay and the quality is further illustrated. Indeed, the results in Fig. 7 show that as the $V_\Delta$ increases, the scheduling algorithm prioritizes maximizing users' HD delivery rate, whereas at lower values of $V_\Delta$, the scheduler prioritizes stabilizing the traffic and virtual queues i.e., keeping the delay bounded with high probability. This comes at the expense of having lower HD delivery rate. The `MPROAC+` approach also achieves 17-37% reduction in the 99th percentile latency as compared to `MPROAC` and `MREAC` schemes, respectively.
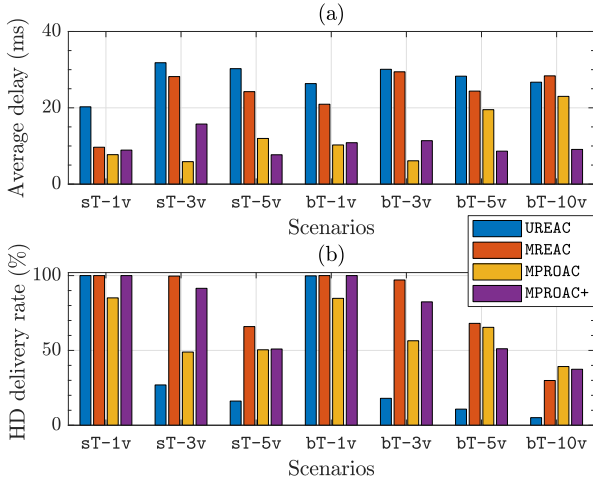
Figure 9. (a) Average delay and (b) HD delivery rate performance for different network scenarios with a load of 1 Gbps per user (0.972 Mb chunk size).

Furthermore, the Jaccard similarity in Fig. 7(d) and Fig. 7(h) illustrates the trade-offs of utility and latency versus transmission utilization. At low traffic loads, high quality rate and low latency result in lower Jaccard index, which is due to the large amount of extra data sent due to sending an estimated FoV. As the traffic load increases, the proactive schemes transmits more real-time frames, which increases the Jaccard index. The Jaccard index decreases again at higher traffic loads as the effect of missed frames goes up (the average delay approaches the deadline as can be seen in Fig. 7(a), Fig. 7(e))

*3) Impact of the Number of Clusters:* Subsequently, we examine how the maximum number of clusters affects the performance of the multicast schemes, as compared to the UREAC scheme, which operates in unicast. Fig. 8 shows that a lower number of clusters allows for higher content reuse of the overlapping areas in the FoVs of users, which results in lower delay and higher HD quality rate. By allowing a higher number of clusters per video, however, higher Jaccard similarity indices are scored, as less unnecessary chunks are sent to users, as shown in Fig. 8(d).

*4) Impact of the Network Size:* Finally, the effect of the network size is analyzed. To do so, both the small and the big theater configurations are considered under an increasing amount of users and videos. In Fig. 9, it is shown that the proposed scheme achieves close to 100% HD streaming rate in scenarios 1, 2, 4, and 5 while maintaining lower frame delay. Moreover, in the congested scenarios with high number of users and videos, i.e., scenarios 3, 6, and 7, the results show that multicasting provides substantial performance improvement through the gains of MREAC over UREAC. This demonstrates the capability of multicasting to minimize the latency of VR streaming to multi-user scenarios. Although the large amount of requested data in these congested scenarios limits the available resources to schedule the predicted frames in advance, the results in Fig. 9 show that the proposed scheme MPROAC+ can achieve higher HD delivery rate and lower delay compared to the baselines.

## VII. CONCLUSIONS

In this paper, we have formulated the problem of maximizing users' VR streaming QoE as a network-wide HD frame admission maximization problem subject to low latency constraints with very high reliability. We have proposed a Lyapunov-framework based approach which transforms the stochastic optimization problem into a series of successive instantaneous static optimization subproblems. Subsequently, for each time instant, a matching theory algorithm is applied to allocate SBS to user clusters and leverage a mmWave multicast transmission of the HD chunks. Using simulations, we have shown that the proposed DRNN can predict the VR users' future FoV with high accuracy. The predictions of this model are used to cluster users and proactively schedule schedule the multicast transmission of their future video chunks. Furthermore, simulations have provided evidence of considerable gains achieved by the proposed model when compared to both reactive baseline schemes, while notably outperforming the unicast transmission baseline.

## APPENDIX A
### PROOF OF LEMMA 1

By leveraging the inequality $(\max[x, o])^2 \le x^2$ for $x \ge 0$, after squaring the physical and virtual queues in (11), (17) and (18) the upper bounds for the above terms are derived as

$$q_u^2(t+1) - q_u^2(t) \le r_u^2(t) + \sum_{f=\{f_r, f_p\}} \sum_{c_f \in \widetilde{\mathcal{N}}_u^f} \left( \mu_{uc_f}^2(t) - 2r_u(t)\mu_{uc_f}(t) \right)$$
$$- 2q_u(t)\left(\mu_{uc_f}(t) - r_u(t)\right), \tag{39}$$

$$z_u^2(t+1) - z_u^2(t) \le r_u^2(t) + \gamma_u^2(t) - 2r_u(t)\gamma_u(t)$$
$$- 2z_u(t)\left(r_u(t) - \gamma_u(t)\right), \tag{40}$$

$$j_{uf}^2(t+1) - j_{uf}^2(t) \le q_u^2(t+1)\left(\mathbb{I}_{\{d_{t2MTP}\}} - \epsilon_d\right)^2$$
$$+ 2j_{uf}(t)\left(\mathbb{I}_{\{d_{t2MTP}\}} - \epsilon_d\right)q_u(t+1), \tag{41}$$

with the one time slot Lyapunov drift given by

$$\Delta L_t \triangleq L(\chi(t+1)) - L(\chi(t))$$
$$= \frac{1}{2} \sum_{u \in \mathcal{U}} \left\{ \left( q_u^2(t+1) - q_u^2(t) \right) + \left( z_u^2(t+1) - z_u^2(t) \right) \right.$$
$$\left. + \sum_{f \in \mathcal{F}} \left( j_{uf}^2(t+1) - j_{uf}^2(t) \right) \right\}. \tag{42}$$

Replacing the term $q_u(t+1)$ in (41) with $q_u(t+1) = q_u(t) + \sum_{f=\{f_r, f_p\}} \sum_{c_f \in \widetilde{\mathcal{N}}_u^f} r_u(t) - \mu_{uc_f}(t)$ due to the fact that having $\mathbb{I}_{\{d_{t2MTP}\}} = 1$ entails a non-empty queue guarantee, and combining (39)-(41), an upperbound on the drift function can be expressed as (43).

Note that the terms #b, #c, and #e in (43) are quadratic, therefore upper bounded to comply with the assumption of queue stability. Hence, let

$$\Delta_0(t) \ge \frac{1}{2} \sum_{u \in \mathcal{U}} \left\{ 2\left(\mathbb{I}_{\{d_{t2MTP}\}} - \epsilon_d\right)\left\{ q_u(t) \sum_{f=\{f_r, f_p\}} j_{uf}(t) \right\} \right. \tag{44}$$
$$\left. + \left(\mathbb{I}_{\{d_{t2MTP}\}} - \epsilon_d\right)^2 \left\{ q_u^2(t) + \left( \sum_{f=\{f_r, f_p\}} \sum_{c_f \in \widetilde{\mathcal{N}}_u^f} \mu_{uc_f}(t) - r_u(t) \right)^2 \right\} \right.$$

$$\Delta L_t \leq \frac{1}{2}\sum_{u\in\mathcal{U}}\left[-2\left(\mathbb{I}_{\{d_{t2\mathrm{MTP}}\}}-\epsilon_d\right)^2\left\{q_u(t)\left(\sum_{f=\{f_r,f_p\}}\sum_{c_f\in\widetilde{\mathcal{N}}_u^f}\mu_{uc_f}(t)-r_u(t)\right)\right\}_{\#a} + \left(\mathbb{I}_{\{d_{t2\mathrm{MTP}}\}}-\epsilon_d\right)^2\left\{q_u^2(t)+\left(\sum_{f=\{f_r,f_p\}}\sum_{c_f\in\widetilde{\mathcal{N}}_u^f}\mu_{uc_f}(t)-r_u(t)\right)^2\right\}_{\#b}\right.$$

$$+\left\{\left(\sum_{f=\{f_r,f_p\}}\sum_{c_f\in\widetilde{\mathcal{N}}_u^f}\mu_{uc_f}(t)-r_u(t)\right)^2+\left(r_u(t)-\gamma_u(t)\right)^2\right\}_{\#e} -2\left\{z_u(t)\left(r_u(t)-\gamma_u(t)\right)\right\}_{\#f} -2\left\{q_u(t)\left(\sum_{f=\{f_r,f_p\}}\sum_{c_f\in\widetilde{\mathcal{N}}_u^f}\mu_{uc_f}(t)-r_u(t)\right)\right\}_{\#g}$$

$$+2\left(\mathbb{I}_{\{d_{t2\mathrm{MTP}}\}}-\epsilon_d\right)\left\{q_u(t)\sum_{f=\{f_r,f_p\}}j_{uf}(t)\right\}_{\#c} -2\left(\mathbb{I}_{\{d_{t2\mathrm{MTP}}\}}-\epsilon_d\right)\left\{\sum_{f=\{f_r,f_p\}}\sum_{c_f\in\widetilde{\mathcal{N}}_u^f}j_{uf}(t)\left(\mu_{uc_f}(t)-r_u(t)\right)\right\}_{\#d}\right]. \qquad (43)$$

$$+\left\{\left(\sum_{f=\{f_r,f_p\}}\sum_{c_f\in\widetilde{\mathcal{N}}_u^f}\mu_{uc_f}(t)-r_u(t)\right)^2+\left(r_u(t)-\gamma_u(t)\right)^2\right\}\right\}$$

be the constant parameter at each time instant $t$ collecting the aforementioned terms from the drift above. After subtracting the penalty term $V_\Delta\mathbb{E}\left[U\left(\{\gamma_u(t)\}\right)\right]$ on both sides of (43), and further operating on #a, #d and #g to denote $\alpha_u$ the term

$$\alpha_u(t) = \left[\left(\mathbb{I}_{\{d_{t2\mathrm{MTP}}\}}-\epsilon_d\right)q_u(t)+\sum_{f\in\mathcal{F}}j_{uf}(t)\right]\left(\mathbb{I}_{\{d_{t2\mathrm{MTP}}\}}-\epsilon_d\right)+q_u(t),$$

we have that

$$\Delta L_t - V_\Delta\mathbb{E}\left[U\left(\{\gamma_u(t)\}\right)\right] \leq \Delta_0(t) - \sum_{u\in\mathcal{U}}V_\Delta\mathbb{E}\left[U\left(\{\gamma_u(t)\}\right)\right]$$
$$- \sum_{u\in\mathcal{U}}\alpha_u(t)\left(\sum_{f=\{f_r,f_p\}}\sum_{c_f\in\widetilde{\mathcal{N}}_u^f}\mu_{uc_f}(t)-r_u(t)\right)$$
$$- \sum_{u\in\mathcal{U}}z_u(t)\left(r_u(t)-\gamma_u(t)\right),$$

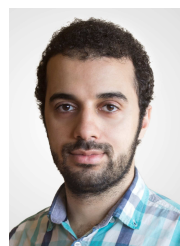which after some more rearrangements yields equation (20).

## REFERENCES

[1] E. Baştuğ, M. Bennis *et al.*, "Toward interconnected virtual reality: Opportunities, challenges, and enablers," *IEEE Commun. Mag.*, vol. 55, no. 6, pp. 110–117, June 2017.

[2] C. Hendrix and W. Barfield, "Presence within virtual environments as a function of visual display parameters," *Presence: Teleoperators and Virtual Environments*, vol. 5, no. 3, pp. 274–289, 1996.

[3] Cisco, "Cisco visual networking index: Global mobile data traffic forecast update, 2017â̆Ş2022," institution, Tech. Rep., 2 2017.

[4] M. S. Elbamby, C. Perfecto *et al.*, "Toward low-latency and ultra-reliable virtual reality," *IEEE Netw.*, vol. 32, no. 2, pp. 78–84, March 2018.

[5] M. Bennis, M. Debbah, and H. V. Poor, "Ultra-reliable and low-latency wireless communication: Tail, risk and scale," *Proc. IEEE*, vol. 106, no. 10, pp. 1834–1853, 10 2018.

[6] M. Zink, R. Sitaraman, and K. Nahrstedt, "Scalable 360° video stream delivery: Challenges, solutions, and opportunities," *Proceedings of the IEEE*, vol. 107, no. 4, pp. 639–650, April 2019.

[7] K. Doppler, E. Torkildson, and J. Bouwen, "On wireless networks for the era of mixed reality," in *Proc. Eur. Conf. on Netw. and Commun. (EuCNC)*, June 2017, pp. 1–6.

[8] P. Lungaro, R. Sjoberg *et al.*, "Gaze-aware streaming solutions for the next generation of mobile vr experiences," *IEEE Trans. Vis. Comput. Graphics*, vol. 24, no. 4, pp. 1535–1544, 2018.

[9] X. Corbillon, G. Simon *et al.*, "Viewport-adaptive navigable 360-degree video delivery," in *Proc. of IEEE Int. Conf. on Commun. (ICC)*, 2017, pp. 1–7.

[10] M. Hosseini and V. Swaminathan, "Adaptive 360 VR video streaming: Divide and conquer," in *IEEE Int. Symp. on Multimedia (ISM)*, Dec 2016, pp. 107–110.

[11] F. Qian, L. Ji *et al.*, "Optimizing 360° video delivery over cellular networks," in *Proc. Int. Conf. Mobile Comp. and Netw. (MOBICOM)*, New York, NY, USA, 2016, pp. 1–6.

[12] M. Xiao, C. Zhou *et al.*, "Optile: Toward optimal tiling in 360-degree video streaming," in *Proc. of ACM Conf. on Multimedia*, ser. MM '17. New York, NY, USA: ACM, 2017, pp. 708–716.

[13] A. Ghosh, V. Aggarwal, and F. Qian, "A rate adaptation algorithm for tile-based 360-degree video streaming," *CoRR*, vol. abs/1704.08215, 2017.

[14] M. Chen, U. Challita *et al.*, "Machine learning for wireless networks with artificial intelligence: A tutorial on neural networks," *CoRR*, vol. abs/1710.02913, 2017.

[15] M. Cornia, L. Baraldi *et al.*, "Predicting human eye fixations via an lstm-based saliency attentive model," *IEEE Trans. Image Proc.*, vol. 27, pp. 5142–5154, 2018.

[16] Y. Bao, H. Wu *et al.*, "Shooting a moving target: Motion-prediction-based transmission for 360-degree videos," in *IEEE Int. Conf. on Big Data*, Dec 2016, pp. 1161–1170.

[17] V. Sitzmann, A. Serrano *et al.*, "Saliency in vr: How do people explore virtual environments?" *IEEE Trans. Visualization and Comp. Graphics*, vol. 24, no. 4, pp. 1633–1642, 4 2018.

[18] A. Nguyen, Z. Yan, and K. Nahrstedt, "Your attention is unique: Detecting 360-degree video saliency in head-mounted display for head movement prediction," in *Proc. ACM Int. Conf. on Multimedia*, 2018, pp. 1190–1198.

[19] A. Nguyen and Z. Yan, "A saliency dataset for 360-degree videos," in *Proc. ACM Multimedia Systems Conference*. ACM, 2019, pp. 279–284.

[20] C.-L. Fan, J. Lee *et al.*, "Fixation prediction for 360°; video streaming in head-mounted virtual reality," in *Proc. Wksp on Network and Op. Sys. Support for Dig. Audio and Video*. New York, NY, USA: ACM, 2017, pp. 67–72.

[21] C. Li, W. Zhang *et al.*, "Very long term field of view prediction for 360-degree video streaming," *CoRR*, vol. abs/1902.01439, 2019.

[22] M. Xu, Y. Song *et al.*, "Predicting head movement in panoramic video: A deep reinforcement learning approach." *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2018.

[23] Y. Zhang, P. Zhao *et al.*, "Drl360: 360-degree video streaming with deep reinforcement learning," in *IEEE Conf. on Comp. Commun. (INFOCOM 2019)*, April 2019, pp. 1252–1260.

[24] X. Hou, Y. Lu, and S. Dey, "Wireless VR/AR with edge/cloud computing," in *Int. Conf. Comp. Commun. and Netw. (ICCCN)*, July 2017, pp. 1–8.

[25] S. Mangiante, K. Guenter *et al.*, "VR is on the edge: How to deliver 360° videos in mobile networks," in *Proc. ACM SIGCOMM. Wksh. on VR/AR Network*, 2017.

[26] J. Chakareski, "VR/AR immersive communication: Caching, edge computing, and transmission trade-offs," in *Proc. ACM SIGCOMM. Wksh. on VR/AR Network*. New York, NY, USA: ACM, 2017, pp. 36–41.

[27] M. S. Elbamby, C. Perfecto *et al.*, "Edge computing meets millimeter-wave enabled VR: Paving the way to cutting the cord," in *2018 IEEE Wireless Commun. and Netw. Conf. (WCNC)*, April 2018.

[28] Y. Sun, Z. Chen *et al.*, "Communication, computing and caching for mobile VR delivery: Modeling and trade-off," in *IEEE Int. Conf. on Commun. (ICC)*, May 2018, pp. 1–6.

[29] A. Prasad, M. A. Uusitalo *et al.*, "Challenges for enabling virtual reality broadcast using 5g small cell network," in *2018 IEEE Wireless Commun. and Netw. Conf. Wksh.s (WCNCW)*, April 2018.

[30] A. Prasad, A. Maeder, and M. A. Uusitalo, "Optimizing over-the-air virtual reality broadcast transmissions with low-latency feedback," in *IEEE 5G World Forum (5G-WF)*, July 2018.

[31] N. Carlsson and D. Eager, "Had you looked where i'm looking: Cross-user similarities in viewing behavior for 360 video and caching implications," *arXiv preprint arXiv:1906.09779*, 2019.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCOMM.2020.2965527, IEEE Transactions on Communications

17

[32] M. Chen, W. Saad *et al.*, "Echo state transfer learning for data correlation aware resource allocation in wireless virtual reality," in *Asilomar Conf. Signals, Syst., and Comp.*, Oct. 2017.

[33] N. D. Sidiropoulos, T. N. Davidson, and Z.-Q. Luo, "Transmit beamforming for physical-layer multicasting," *IEEE Trans. Signal Process.*, vol. 54, no. 6, pp. 2239–2251, June 2006.

[34] 3GPP, "ETSI TR 138 901 V14.3.0: 5G; Study on channel model for frequencies from 0.5 to 100 GHz (3GPP TR 38.901 version 14.3.0 release 14)," European Telecommunications Standards Institute (ETSI), Tech. Rep., 2018.

[35] V. Raghavan, L. Akhoondzadeh-Asl *et al.*, "Statistical blockage modeling and robustness of beamforming in millimeter-wave systems," *IEEE Transactions on Microwave Theory and Techniques*, 2019.

[36] J. Wildman, P. H. J. Nardelli *et al.*, "On the joint impact of beamwidth and orientation error on throughput in directional wireless poisson networks," *IEEE Trans. Wireless Commun.*, vol. 13, no. 12, pp. 7072–7085, Dec. 2014.

[37] M. J. Neely, "Stochastic network optimization with application to communication and queueing systems," *Synthesis Lect. on Commun. Netw.*, vol. 3, no. 1, pp. 1–211, 2010.

[38] L. Georgiadis, M. J. Neely, and L. Tassiulas, *Resource Allocation and Cross-Layer Control in Wireless Networks*, ser. Foundations and Trends in Networking. Now Publishers Inc., 2006.

[39] A. Roth and M. Sotomayor, *Two-sided matching: A study in game-theoretic modeling and analysis*. Cambridge University Press, 1992.

[40] Y. Gu, W. Saad *et al.*, "Matching theory for future wireless networks: fundamentals and applications," *IEEE Commun. Mag.*, vol. 53, no. 5, pp. 52–59, May 2015.

[41] D. Gale and L. S. Shapley, "College admissions and the stability of marriage," *Am. Math. Mon.*, vol. 69, no. 1, pp. 9–15, 1962.

[42] Z. Zhou, K. Ota *et al.*, "Energy-efficient matching for resource allocation in d2d enabled cellular networks," *IEEE Trans. Veh. Technol.*, vol. 66, no. 6, pp. 5256–5268, June 2017.

[43] B. Di, L. Song, and Y. Li, "Sub-channel assignment, power allocation, and user scheduling for non-orthogonal multiple access networks," *IEEE Trans. Wireless Commun.*, vol. 15, no. 11, pp. 7686–7698, Nov 2016.

[44] K. Cho, B. van Merriënboer *et al.*, "Learning phrase representations using RNN encoder–decoder for statistical machine translation," in *Proc. Conf. Emp. Methods Natural Lang. Process. (EMNLP)*, Oct. 2014, pp. 1724–1734.

[45] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov 1997.

[46] J. Chung, C. Gulcehre *et al.*, "Empirical evaluation of gated recurrent neural networks on sequence modeling," in *NIPS 2014 Wksh. on Deep Learning*, 2014.

[47] F. Chollet *et al.*, "Keras," https://keras.io, 2015.

[48] P. J. Werbos, "Backpropagation through time: what it does and how to do it," *Proc. IEEE*, vol. 78, no. 10, pp. 1550–1560, Oct 1990.

[49] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014.

[50] W.-C. Lo, C.-L. Fan *et al.*, "360° video viewing dataset in head-mounted virtual reality," in *Proc. ACM Conf. on Multimedia Syst.*, 2017, pp. 211–216.

**Mohammed S. Elbamby** received the B.Sc. degree (Hons.) in Electronics and Communications Engineering from the Institute of Aviation Engineering and Technology, Egypt, in 2010, the M.Sc. degree in Communications Engineering from Cairo University, Egypt, in 2013, and the Dr.Sc. Degree (with distinction) in Communications Engineering from the University of Oulu, Finland, in 2019. He is currently with Nokia Bell Labs in Espoo, Finland. Previously, he held research positions at the University of Oulu, Cairo University, and the American University in Cairo. His research interests span resource optimization, network management, and machine learning in wireless cellular networks. He received the Best Student Paper Award from the European Conference on Networks and Communications (EuCNC'2017).
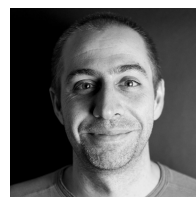


**Javier Del Ser** (M'07-SM'12) received his first Ph.D. degree (cum laude) in Electrical Engineering from the University of Navarra (Spain) in 2006, and a second Ph.D. degree (cum laude, extraordinary Ph.D. prize) in Computational Intelligence from the University of Alcala (Spain) in 2013. He is currently a Research Professor in Artificial Intelligence at TECNALIA, Spain. He is also an adjunct professor at the University of the Basque Country (UPV/EHU), an invited research fellow at the Basque Center for Applied Mathematics (BCAM), an a senior AI advisor at the technological startup SHERPA.AI. He is also the coordinator of the Joint Research Lab between TECNALIA, UPV/EHU and BCAM (http://jrlab.science), and the director of the TECNALIA Chair in Artificial Intelligence implemented at the University of Granada (Spain). His research interests gravitate on the design of artificial intelligence methods for data mining and optimization problems emerging from Industry 4.0, intelligent transportation systems, smart mobility, logistics and health, among other fields of application. He has published more than 290 scientific articles, co-supervised 10 Ph.D. theses, edited 7 books, co-authored 9 patents and participated/led more than 40 research projects. He is an Associate Editor of tier-one journals from areas related to data science and artificial intelligence such as Information Fusion, Swarm and Evolutionary Computation and Cognitive Computation.



**Mehdi Bennis** (S'07-AM'08-SM'15) is an Associate Professor at the Centre for Wireless Communications (CWC), University of Oulu, Finland, an Academy of Finland Research Fellow and head of the Intelligent COnnectivity and Networks/systems group (ICON). His main research interests are in radio resource management, heterogeneous networks, game theory and machine learning in 5G networks and beyond. He has co-authored one book and published more than 200 research papers in international conferences, journals and book chapters. He has been the recipient of several prestigious awards including the 2015 Fred W. Ellersick Prize from the IEEE Communications Society, the 2016 Best Tutorial Prize from the IEEE Communications Society, the 2017 EURASIP Best paper Award for the Journal of Wireless Communications and Networks, the all-University of Oulu award for research and the 2019 IEEE ComSoc Radio Communications Committee Early Achievement Award. Dr Bennis is an editor for the IEEE TRANSACTIONS ON COMMUNICATIONS.



**Cristina Perfecto** (S'15-AM'20) received her M.Sc. in Telecommunication Engineering and Ph.D. (with distinction) on Information and Communication Technologies for Mobile Networks from the University of the Basque Country (UPV/EHU) in 2000 and 2019, respectively. She is currently an Associate Professor with the Department of Communications Engineering at this same University. During 2016, 2017 and 2018 she was a Visiting Researcher at the Centre for Wireless Communications (CWC), University of Oulu, Finland where she worked on the application of multidisciplinary computational intelligence techniques in millimeter wave communications radio resource management. Her current research interests lie on the use of machine learning and data analytics, including different fields such as metaheuristics and bio-inspired computation, for resource optimization in 5G networks and beyond.