M5: Facilitating Multi-user Volumetric Content Delivery with Multi-lobe Multicast over mmWave

Ding Zhang
George Mason University
dzhang13@gmu.edu

Puqi Zhou
George Mason University
pzhou@gmu.edu

Bo Han
George Mason University
bohan@gmu.edu

Parth Pathak
George Mason University
phpathak@gmu.edu

ABSTRACT

Multi-user volumetric content delivery can enable numerous appealing applications, such as online education, telehealth, multi-user AR/VR training, immersive collaborative analytics, etc. However, the bandwidth-intensive nature of volumetric video streaming makes existing systems for single-user experiences hard to scale to multi-user scenarios. To address this critical issue, in this paper, we first perform a scaling experiment on mmWave networks that offer the needed multi-Gbps throughput and identify two key challenges of streaming high-quality volumetric videos to multiple users: *frequent blockages of mmWave links and high transmission redundancy* among users. To solve these problems, we propose a first-of-its-kind, agile, and cross-layer system, dubbed M5, for improving the performance and quality of experience for multi-user volumetric video streaming. M5 utilizes the 6DoF motion prediction of users to proactively adapt mmWave beams and prefetch frames to mitigate the blockage effects. Furthermore, it takes advantage of the multicast transmission to deliver the overlapped common content within users' viewports to reduce the bandwidth requirement. Our extensive experiments on a real testbed and with a trace-driven simulator show that M5 can effectively improve the frame rate by 44.1% and volumetric video quality by 62.3% compared to the state-of-the-art system.

ACM Reference Format:
Ding Zhang, Puqi Zhou, Bo Han, and Parth Pathak. 2022. M5: Facilitating Multi-user Volumetric Content Delivery with Multi-lobe Multicast over mmWave. In *The 20th ACM Conference on Embedded Networked Sensor Systems (SenSys '22), November 6–9, 2022, Boston, MA, USA.* ACM, New York, NY, USA, 16 pages. https://doi.org/10.1145/3560905.3568540

1 INTRODUCTION

Volumetric content (e.g., point clouds and 3D meshes) empowers an immersive and interactive viewing experience by enabling users to explore it with six degrees of freedom (6DoF) motion [20, 29]. When watching a volumetric video, users can move along not only rotational dimensions (yaw, pitch, and roll), which have already been supported by 360° videos [22, 53], but also translational dimensions (X, Y, and Z). When used in augmented reality (AR) or virtual reality (VR), volumetric videos lead to numerous promising applications in entertainment, education, training, healthcare, etc. For example, students can attend classes remotely and enjoy the volumetric content of their teacher that demonstrates complex 3D models in real-time. With telesurgery, a remote surgeon can operate on wounded soldiers on battlefields via their live volumetric content feed to save their lives. Thus, volumetric videos have been deemed one of the key applications for 5G and beyond [40, 42].

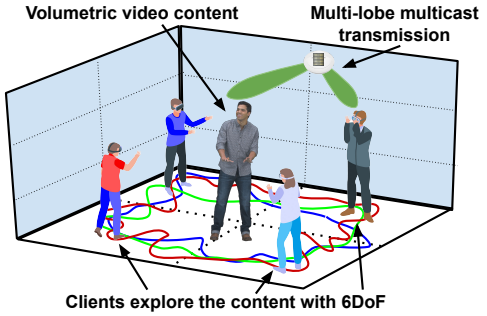


Fig. 1: Multi-user volumetric video streaming with multi-lobe multicast in mmWave WLAN

However, volumetric video streaming is extremely bandwidth intensive. Compared to 4K videos (25-40 Mbps) and 360° videos (200-300 Mbps), volumetric videos require even higher data rates (>400 Mbps) when streaming without any optimization [20, 29]. Given the data-intensive nature of volumetric video streaming, most existing research focuses on single-user experiences [20, 29, 35, 48, 83, 84], with limited attention being paid to multi-user scenarios that are naturally required in the aforementioned use cases. Among the state-of-the-art systems, even with its three visibility-aware optimizations that consider viewport, distance, and occlusion, ViVo [20] still requires a data rate of ~200 Mbps for a single user to watch a medium-quality volumetric video with ~200K points per frame. As a result, a naive extension of ViVo that streams high-quality volumetric videos to multiple users may linearly increase the network throughput and consume extremely high bandwidth.

In this paper, we first conduct a scaling experiment to understand the challenges of multi-user volumetric video systems over mmWave WLANs (wireless local area networks), whose standards such as the 802.11ad/ay [23, 24] can provide multi-gigabit per second data rates. However, despite the high data rate of mmWave WLAN, we identify two unique challenges to supporting multi-user volumetric video streaming.

(1) Mobility and blockages are known issues in mmWave wireless network and could be even more severe in multi-user volumetric video streaming, where the 6DoF motion of users requires continuous beam adaptation to deal with mobility. Furthermore, inter-user and self-body blockages can deteriorate the performance of mmWave links, increase the video stalls, and reduce the video quality, which will significantly affect the quality of experience (QoE). Our experiments with real 6DoF motion traces [14, 20] collected when consuming volumetric videos reveal that the frame rate drops sharply with the increasing number of inter-user and self blockages. Existing solutions of blockage mitigation [19, 24, 25, 68, 69, 85] mainly rely on physical layer information without exploiting the application-specific information such as users' 6DoF motion. Other solutions [76, 82], which exploit motion information, do not consider multi-user scenarios where complete blockages can be unavoidable (e.g., even reflected paths are not available). Hence, this

SenSys '22, November 6–9, 2022, Boston, MA, USA

ACM ISBN 978-1-4503-9886-2/22/11.
https://doi.org/10.1145/3560905.3568540

issue requires an application-aware, agile solution that can predict blockages and adapt beams in LoS (line of sight) and NLoS (Non-line of sight) situations for uninterrupted, high-quality volumetric video streaming.

(2) Our empirical analysis of volumetric video content being streamed over wireless links to multiple users reveals that there is a significant amount of redundancy between the content (e.g., volumetric video cells in ViVo [20]) being transmitted to users. Thus, just naively extending the state-of-the-art volumetric video streaming solutions [20] to multiple users without considering the redundancy can result in significant wastage of available bandwidth, limiting the number of users that can be supported by the system or the video quality for the given users.

In this paper, we propose M5, a first-of-its-kind system for multi-user volumetric video delivery with multi-lobe multicast over mmWave networks. M5 takes advantage of the 6DoF motion prediction of users to predict the blockages and reacts to them with proactive beam adaptation and prefetching. It also utilizes the high content similarity between multiple users to facilitate a multi-lobe multicast transmission to improve their QoE. Specifically, M5 innovates in the following aspects:

Characterizing multi-user volumetric video streaming over mmWave. While volumetric videos for AR/VR are being considered a key application for mmWave 5G networks, there is no existing study that systematically investigates the performance of volumetric video streaming over mmWave. To address this issue, we measure the performance of a state-of-the-art volumetric video streaming system ViVo [20], which was originally proposed for a single user, by extending it to support multiple users over our COTS (commercial off-the-shelf) 802.11ad mmWave testbed. Our study reveals that streaming volumetric content to seven users drops the frame rate of ViVo from 30 fps (frames per second) to 12 fps due to frequent blockages of mmWave links. Also, as much as 64.8% of video content can be redundant when streaming to five users, resulting in a large wastage of network bandwidth. Furthermore, using the real 6DoF motion traces of users, we find that 23% of the time self or inter-user blockages can result in unavoidable network outages for users where no paths are available in the mmWave channel. Given that beam switching based solutions are not helpful in such cases, intelligent content prefetching solutions are needed.

6DoF-motion-aware beam adaptation and blockage prediction. We find that users' 6DoF motion prediction can be critically helpful in mmWave beam adaptation and blockage prediction. Our key insight is that 6DoF motion predictions are not only useful for LoS beam tracking but can help us predict if and until when NLoS paths will be available while a user is consuming volumetric content and when to proactively perform beamforming for leveraging these paths to maintain uninterrupted connectivity. Furthermore, 6DoF motion prediction of multiple users can assist us in accurately predicting unavoidable blockages (all mmWave paths blocked by other users and/or user's body), triggering M5 to prefetch the affected frames before the outage occurs. We develop an adaptive prefetching scheme that finds the right balance between prefetching too much, too early with the inaccurate prediction of blockages vs. accurately predicting blockages but having too little time to prefetch. As a result, M5's adaptive prefetching with 6DoF-motion-aware beam adaptation guarantees high-quality content delivery to multiple users over mmWave links.

Multi-lobe mmWave multicast. M5 exploits the similarity of viewport content among users as an opportunity to scale the system to more users. It utilizes mmWave multicast to deliver the content that overlaps between subsets of users. However, we find it is challenging to perform multicasting over mmWave links due to the use of directional beams. Moreover, the default beams provided by vendor codebooks are not suitable for multicast. To address this issue, we develop a multi-lobe beam solution that can create beams with multiple lobes pointing towards the users of a multicast group while ensuring a high and balanced signal-to-noise ratio (SNR), leading to a high group data rate. Furthermore, we develop a scheduler that can use content-similarity based multicast, 6DoF-motion-based beam adaptation, and blockage-aware prefetching to reliably and efficiently transmit volumetric video frames to multiple users over mmWave links.

Implementation and evaluation of M5. We integrate the above-mentioned modules into a holistic system that can be used for evaluating M5. We use two 6DoF motion traces for volumetric video streaming (ViVo [20] and FHHI [14]) to test M5 with up to 5 users. M5 is evaluated over COTS mmWave 802.11ad devices with 8 patch antenna arrays to create multi-lobe beams as well as a commercial channel simulator (Remcom [58]) for evaluation with controlled multi-user 6DoF traces. Our results show the following.

(1) M5's 6DoF-motion-based beam adaption can proactively switch to available LoS and NLoS paths while reducing the beamforming overhead by 46.3% with a median SNR difference of only 0.54 dB compared to the default 802.11ad beamforming.

(2) The blockage prediction model of M5 can predict the self-blockage and inter-user blockage with 97.2% accuracy. Our blockage-aware prefetching can improve the frame rate by 72.6% and 93.4% for 5 users for ViVo and FHHI datasets, respectively, with an average prefetching accuracy of 96.8% with 533ms prediction window.

(3) We implement the multi-lobe beam design in COTS phased array on 802.11ad devices and show that our customized beam can improve SNR by 5.6 dB compared to the vendor-supplied codebook, which facilitates efficient multicast transmissions in mmWave WLAN for multi-user volumetric video streaming.

(4) In the end-to-end evaluation, when using both prefetching and multicasting, M5 can achieve 167% and 44% improvement in frame rate compared to the baseline scheme and the state-of-the-art system [20], respectively. It also achieves 109% and 62.3% improvement in video quality compared to the two schemes.

2 BACKGROUND AND MOTIVATION

We first conduct extensive measurements to understand the end-to-end performance of multi-user volumetric video streaming over mmWave wireless networks. We use the measurement study to understand the following: (i) how do blockages affect the QoE of users when a volumetric video is streamed to multiple users over a mmWave network? and (ii) how much content redundancy exists in such multi-user volumetric video transmission and how does it affect the user QoE?

Experimental setup. We utilize a server and multiple COTS laptops to measure the volumetric video streaming performance. Both the server (12 cores CPU@3.2GHz and 16GB RAM) and laptops (4 cores CPU@2.8GHz and 8GB RAM) have an 802.11ad NIC with Qualcomm QCA9500 chipset supported by wil6210 [78] driver.

We use the volumetric content (captured at 30 fps) in the 8i dynamic voxelized point cloud dataset [27] as our video source. This point cloud dataset is widely used in the volumetric video streaming research [29, 48, 84]. The original volumetric videos have more than 1M points in each frame. To represent different video quality levels (1 to 6), we resample the frames with 10%, 20%, 40%, 60%, 80%, and

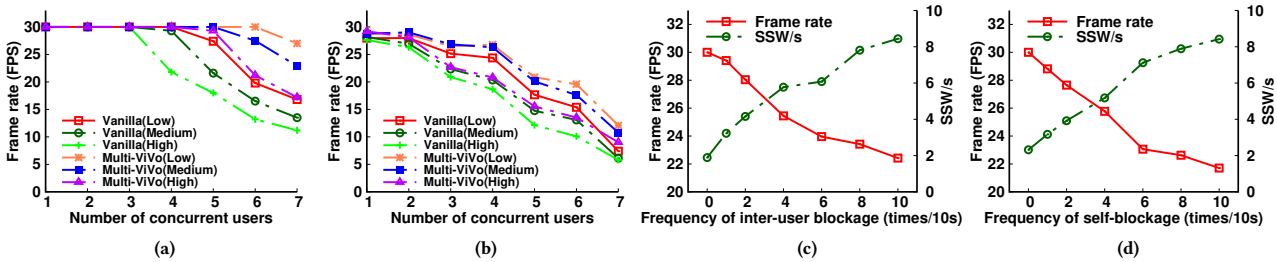


Fig. 2: Performance of multi-user volumetric video streaming with *vanilla* and ViVo [20] systems in (a) static scenario and (b) mobile scenario. (c,d) Multi-user mobility and blockages undermine the QoE of users.

Dataset	Headset Device	Cid	Num. Users	Num. Frames	Num.Virtual Content	Content Size (m^3)
ViVo	Magic Leap One	P2	16	3490	2 (close)	2×2×2
		P3	16	1941	3 (close)	2×2×2
		M2	16	1847	2 (sep.)	5×2×5
		M4	16	2612	4 (sep.)	7×2×7
FHHI	Hololens	C1	14	2001	1	1×1×2

Table 1: Two 6DoF motion datasets used in our work.

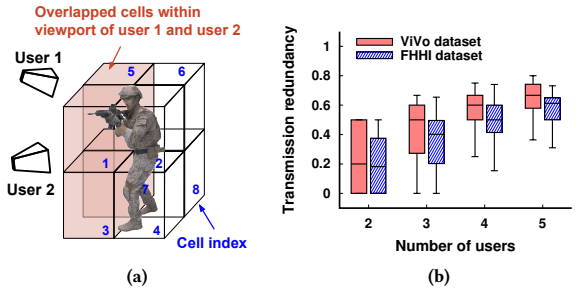


Fig. 3: (a) Viewport similarity among multiple users. (b) The transmission redundancy increases with more users.

100% points. Then we utilize the Draco library [12] to compress the videos. After the compression, the bitrate of different qualities (1 to 6) ranges from 87.5 to 651 Mbps. We implement two video players on the client's laptops. One is the vanilla system that fetches the entire point cloud for each video frame, and the other implements the viewport, occlusion, and distance optimizations in ViVo [20] to reduce the bandwidth requirement. We select three quality levels (2 to 4) of volumetric content, which require 161.6Mbps, 296.4Mbps, and 420.7Mbps bitrate, to represent three different quality levels (low, medium, and high). The quality level 4 has, on average, 590K points per frame, which is the highest point density that can be decompressed by Draco at 30 fps on the client laptops. Next, we benchmark both the vanilla and ViVo systems in stationary and mobile scenarios for multiple users by measuring the maximum achievable frame rate over mmWave WLAN.

2.1 Impact of mobility and blockages

To understand the impact of users' mobility and blockages, we first perform measurements where the same video content [27] is simultaneously delivered to an increasing number of users.

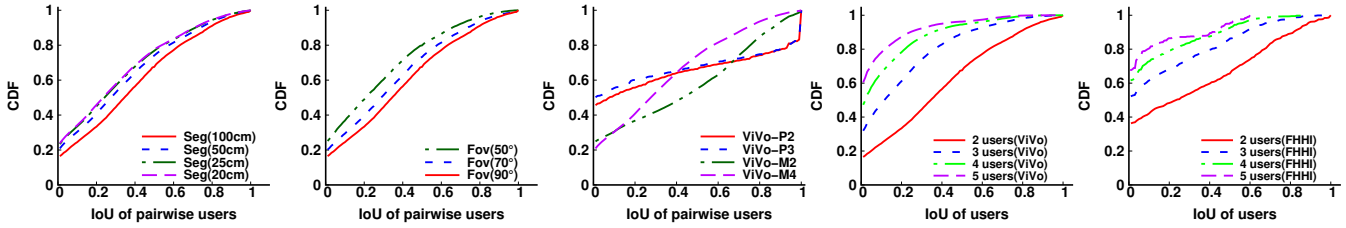
We first collect network throughput traces in a mmWave WLAN using iperf3[1] in a stationary scenario and different mobility and blockage scenarios for multiple mobile clients. In the stationary scenario, the average data rate of our mmWave testbed is 1.68 Gbps. We also collect 10 different throughput traces when different numbers of users freely move in the coverage area of a mmWave WLAN, which creates link fluctuations and blockages. The collected throughput traces are then replayed with both the vanilla and ViVo systems using the tc[2] utility to carefully imitate the same throughput variations for clients over time while different users watch the same volumetric video content.

Fig. 2a and Fig. 2b show the maximum achievable frame rate (capped at 30 fps) for stationary and mobile users, respectively, with different numbers of users and varying densities of points. In the stationary scenario, we observe that the mmWave 802.11ad WLAN can support up to three concurrent users for medium and high-quality videos and up to four users for low-quality videos. With visibility-aware optimizations, ViVo can further increase the number of users by one or two, depending on the video quality.

In the mobile scenario, the users freely move around to explore the volumetric content by holding the device. This mobility of users results in (i) self-body blockage, where the user's body blocks the signal between the access point (AP) and the device, and (ii) inter-user blockage, where one user blocks the LoS or NLoS path from the AP to other users. From Fig. 2b, we observe that compared to the static scenario, user mobility and resultant blockages significantly reduce the frame rate and undermine the QoE of users. In this case, users experience frequent video stalls even when streaming the low-quality content with visibility-aware optimizations. When increasing the number of users to 7, the frame rate can reduce to as low as 7 fps for the vanilla scheme and 12 fps for the multi-user ViVo scheme. Furthermore, high-quality videos are more likely to be affected by user mobility and blockages than low-quality videos due to the higher bandwidth requirement.

To better understand the impact of the two types of blockages, we conduct controlled experiments. We fix the positions of the AP and a client with 3 meters apart and ask another user to walk around the LoS path between them, creating blockages at different frequencies. In addition, we set another 802.11ad compatible device as a wireless monitor to sniff the on-air packets and monitor the number of sector sweeps (SSW) during the blockage events. The reason for capturing the SSW packets is to calculate the beamforming overhead triggered by the mobility and blockages. Fig. 2c shows the frame rate and the number of sector sweeps when increasing this inter-user blockage frequency while streaming low-quality content. The frame rate reduces while the number of SSW increases with the increasing number of blockages. This is because more blockages significantly reduce the LoS link's SNR, frequently triggering beamforming. Similarly, instead of asking another user to block the LoS path, we ask the user holding the device to rotate and create self-blockage scenarios. In Fig. 2d, we see a similar trend of frame rate and the number of SSW with self-blockages as those with the inter-user blockages. The key observation from these experiments is that human body blockages, including both inter-user blockages and self-blockages, are one of the most significant contributors to the poor QoE of multi-user volumetric video streaming in mmWave WLAN. Hence, we need a systematic solution to address this issue when utilizing the multi-gigabit link capacities of mmWave for volumetric video streaming, especially for the multi-user scenarios.

[1]Network measurement tool. https://en.wikipedia.org/wiki/Iperf
[2]Bandwidth control tool. https://en.wikipedia.org/wiki/Tc_(Linux)



(a) Different segmentation (b) Different FoVs (c) Different video content (d) ViVo dataset (e) FHHI dataset

Fig. 4: (a) Viewport similarity decreases when using smaller segmentation or (b) smaller FoV. (c) Viewport similarity varies when users watch different video content, but they share a significant portion of content in general. (d,e) Viewport similarity decreases when increasing concurrent users.

2.2 Multi-user transmission redundancy

In addition to the blockage problems in mmWave WLANs, the transmission redundancy among multiple users also negatively affects the performance of volumetric video streaming. Hence, we investigate the multi-user transmission redundancy with 6DoF motion traces (i.e., viewport trajectories) collected from two datasets listed in Table 1. The ViVo 6DoF motion dataset [20] includes 16 users equipped with the Magic Leap One headset where the users watch four different types of videos with different numbers of virtual content. The FHHI 6DoF motion dataset [14] includes 14 users equipped with Microsoft HoloLens headsets, and the users interacted with one virtual content. ViVo dataset has 30 fps frame rate, and we downsample the FHHI dataset also to 30 fps. Both datasets include translation (x, y, z) and rotation (yaw, pitch, roll) data for each user at different frames.

To calculate the transmission redundancy for volumetric video content between different users, we spatially partition the original virtual content into smaller subregions called *cells*, which have been used in viewport-adaptive volumetric video streaming such as ViVo [20] and GROOT [29]. Each cell is independently prefetchable and decodable. We partition the entire frame into cells of three sizes: $25\times25\times25$ cm^3, $50\times50\times50$ cm^3, and $100\times100\times100$ cm^3. After the partition, we use frustum culling [72] to determine the cells overlapping within the 3D viewport based on the 6DoF motion of the users and calculate a *visibility map* that records the visible cells for each user. We then define the *transmission redundancy* of a group of users as $(total\ transmitted\ cells - union\ of\ cells)/total\ transmitted\ cells$ of their visibility maps. For instance, as shown in Fig. 3a, if we segment the video content into 8 cells, cells 1, 3, and 5-8 are visible for User 1, and cells 1-4, 5, and 7 are visible for User 2. Thus, the total transmitted cells are 12, but the union of cells (needed cells) is 8, so the transmission redundancy here is 4/12.

We show the transmission redundancy of two different datasets in Fig. 3b. We can see that with the increasing number of concurrent users, transmission redundancy grows significantly. For instance, with 2 users in the ViVo dataset, 19.1% of the frame content is redundant on average. This rate surges to 64.8% for 5 users. A similar trend can be seen in the FHHI dataset. As we discuss next, this high transmission redundancy can stem from the similarity between users' viewports.

Viewport similarity. Bao *et al.* [8] utilized the overlapped viewport of multiple users to improve the transmission efficiency in tiled-based multi-user 360° video streaming. However, whether and how the viewports overlap in 3D space among numerous users in volumetric video streaming is still unexplored. Similar to the transmission redundancy, we investigate the viewport similarity in multi-user volumetric video streaming based on the 6DoF motion trajectories in both ViVo and FHHI datasets. We define the *viewport similarity* of a group of users as the intersection over union (IoU) of their visibility maps. For the same example in Fig. 3a, cells 1, 3, and 5-8 are visible for User 1, and cells 1-4, 5, and 7 are visible for User 2. Thus, cells 1, 3, 5, and 7 will be needed for both users, and their viewport similarity (IoU) for this frame is 4/8.

We analyze the viewport similarity among different datasets and make the following observations. Firstly, viewport similarity is affected by many factors, including the segmentation granularity, the field of view (FoV), and the video content. Fig. 4a shows the CDF of the viewport similarity (IoU) among all users for different segmentation granularity. The IoU between users decreases when the number of cells increases with finer-grained segmentation because the finer cells will describe a more accurate shape of the overlapped volume of the content. Regarding the FoV of users, as shown in Fig. 4b, a larger FoV leads to a higher IoU among users because more content is displayed within the larger FoV for each frame.

In terms of the video content, there are four different content types representing different content locations in the ViVo dataset. We calculate IoU among two users for different video content in the ViVo dataset and show the CDF in Fig. 4c. Different video content results in different IoU distributions because users might choose to explore them differently (from different places) based on their preferences and interest. However, the average IoU for them is 0.34, 0.33, 0.41, and 0.31, respectively, which shows a significant overlap of users' viewports even with different content types. We also calculate the IoU for up to 5 users for the ViVo and FHHI datasets, respectively. As shown in Fig. 4d and Fig. 4e, when calculating the viewport similarity for a group with more users, the IoU decreases as more users bring more variations in their positions and orientations.

Though viewport similarity varies with different factors, we find a significant viewport overlap for two or three users. For instance, Fig. 4a shows despite the different segmentation sizes, the average IoU among two users is from 28.8% to 36.5%, which means the users share around 30% of the same content per frame when they watch the volumetric content simultaneously. On the other hand, the average of the IoU among 2 to 5 users in the ViVo dataset drops from 38.2% to 6.9% (50cm segmentation and 90° FoV) as shown in Fig. 4d. The IoU among 2 to 5 users drops from 31.9% to 7.7% in FHHI dataset (Fig. 4e). It is because more users vary with their viewing preferences, resulting in diverse positions and orientations. These viewport similarity observations motivate us to leverage multicasting to users with high viewport similarity to utilize the available bandwidth more efficiently.

3 M5 OVERVIEW

Fig. 5 shows the overview of M5. Here, the clients update the server with their current 6DoF motion at a rate of 30 Hz [20]. The updated 6DoF motion of clients is used by the server to first perform 6DoF motion prediction. The predicted 6DoF motion of clients is then used for (i) beam adaptation, (ii) blockage prediction, and (iii) viewport and content prediction.

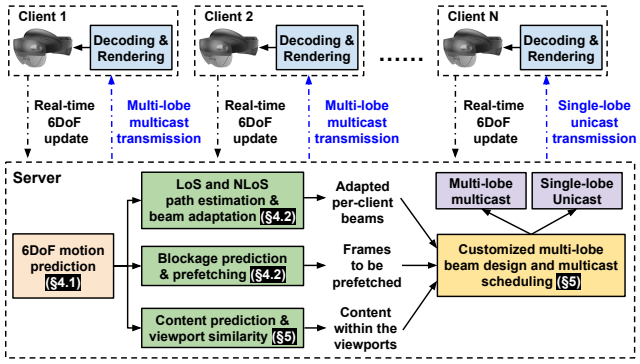
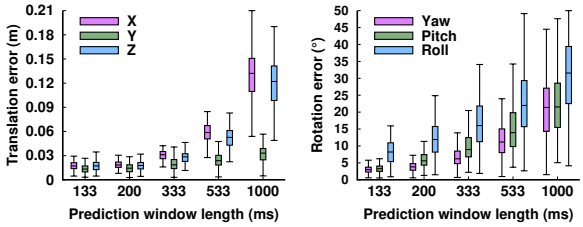


Fig. 5: M5 Overview



(a) Translation prediction error (b) Rotation prediction error

Fig. 6: 6DoF motion prediction error for linear regression

First, the 6DoF motion prediction of clients is leveraged for predicting the availability of mmWave LoS paths and adapting the beam accordingly. It is also used for determining LoS blockage prediction and to switch to a NLoS path that is found using beamforming proactively. Consequently, if the NLoS paths are not available or clients have significant 6DoF motion that renders the proactive beamforming less useful, prefetching is employed to ensure the frames that would be potentially affected by an unavoidable blockage are fetched well in advance. 6DoF motion is also exploited for predicting the viewport of clients and fetching the frames in advance based on their predicted viewport. The similarity between the predicted content of clients is also calculated. Next, the beams adapted based on LoS and NLoS predictions, frames to be prefetched due to blockages, and 6DoF motion-based content prediction and its similarity become input to the multi-lobe multicast scheduler. The scheduler determines the unicast and multicast transmissions based on the content similarity. It identifies multicast grouping between the clients that can achieve the shortest transmission time for their shared content. Based on the multicast groups, multi-lobe beams that can achieve high and balanced SNR for all users of each group are calculated. The scheduler then increases the quality of the unicast and multicast transmissions while adhering to the 30 fps frame rate constraint. The transmissions are then carried out from the mmWave AP over the adapted single and multi-lobe beams. The received content is then decoded and rendered to the clients.

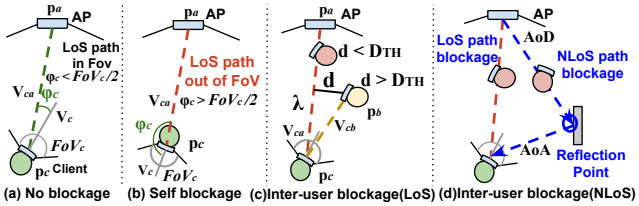
We assume that while the actual volumetric video transmissions are carried out mmWave links, there exists a 2.4 GHz control channel that can be used for reliable 6DoF motion feedback from clients to the server. M5 does not assume any prior knowledge about the environment and/or ambient reflectors for mmWave reflections.

4 BLOCKAGE PREDICTION AND MITIGATION

In this section, we first describe our 6DoF motion prediction model and the blockage prediction model. We then explain how M5 uses them for beam tracking, blockage prediction and prefetching.

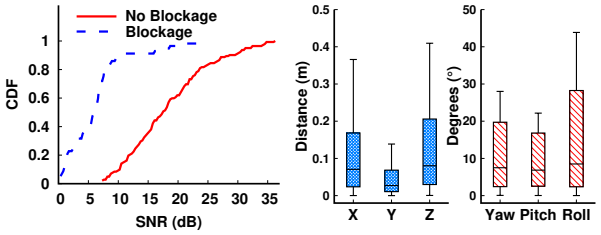
4.1 6DoF motion and blockage prediction

6DoF motion prediction. M5 utilizes an online linear regression model [20, 80] (runs on the server) to predict the 6DoF motion of



(a) No blockage (b) Self blockage (c)Inter-user blockage(LoS) (d)Inter-user blockage(NLoS)

Fig. 7: M5's self and inter-user blockage prediction model



(a) (b)

Fig. 8: (a) Accuracy of our blockage prediction model (b) Translational and rotational movement within the same mmWave sector

clients. Here, the three dimensions of position (X, Y, Z) and another three dimensions of orientation (pitch, yaw, roll) are predicted separately. At every time t, we utilize a history window of length H to predict the 6DoF motion at time $t + P$, where P is the length of the prediction window. Fig. 6 shows the prediction error for ViVo and FHHI dataset with $H = P/2$ [80]. The median error of translational movement with a $200ms$ prediction window is less than $3.5cm$, while the median rotation error is less than 8°. We also find that the regression model has similar prediction errors for translational and rotational movements in FHHI dataset [15].

Blockage prediction model. M5 utilizes a raytracing model [26, 67, 76, 77] for blockage prediction. It considers two types of blockages: self blockage and inter-user blockage. Self blockage is when a client's body itself blocks the FoV of the phased array of her device. While prior work considers the self blockage [76], M5 also predicts inter-user blockages in the multi-user volumetric video streaming scenario. An inter-user blockage occurs when other users block the paths between the AP's and the client's phased array.

Fig. 7 shows how we use raytracing to determine the two blockages. Let $p_a = [x_a, y_a, z_a, \alpha_a, \beta_a, \gamma_a]$ and $p_c = [x_c, y_c, z_c, \alpha_c, \beta_c, \gamma_c]$ denote the 6DoF pose of the AP and a client in the world coordinate system, respectively. We can calculate two vectors: (i) a lookat direction of the client as $\mathbf{V}_c = [\cos\beta_c \sin\alpha_c, \sin\beta_c, \cos\beta_c \cos\alpha_c]$, and (ii) the direction from client to AP as $\mathbf{V}_{ca} = [x_a, y_a, z_a]^\mathbf{T} - [x_c, y_c, z_c]^\mathbf{T}$. The angle ϕ_c between the vectors $\mathbf{V}_c$ and $\mathbf{V}_{ca}$ is calculated as $\phi_c = atan2(\|\mathbf{V}_c \times \mathbf{V}_{ca}\|, \mathbf{V}_c \cdot \mathbf{V}_{ca})$ where function $atan2()$ is the four quadrant inverse tangent. The angle ϕ_a between the lookat direction (norm vector) of AP and $\mathbf{V}_{ac}$ is calculated in the same way. A self blockage is detected when the condition $\phi_a > FoV_a/2$ or $\phi_c > FoV_c/2$ is true, indicating that the LoS path between AP and client is out of the FoV of AP or client (Fig. 7b).

For determining inter-user blockage, we calculate the perpendicular distance d between a potential blocker p_b and the vector $\mathbf{V}_{ca}$. We calculate the ratio $\lambda = \frac{\mathbf{V}_{cb} \cdot \mathbf{V}_{ca}}{\|\mathbf{V}_{ca}\|^2}$ and distance $d = \frac{\|\mathbf{V}_{cb} \times \mathbf{V}_{ca}\|}{\|\mathbf{V}_{ca}\|}$. λ is used to inspect if the perpendicular point is on vector $\mathbf{V}_{ca}$ while d is the distance from the blocker to the perpendicular point on the vector. If $0 < \lambda < 1$ and $d < D_{TH}$, the blocker blocks the path from AP to the client (Fig. 7c). Here, D_{TH} is a threshold to model the size of a human body and we use $0.3m$ in our model.

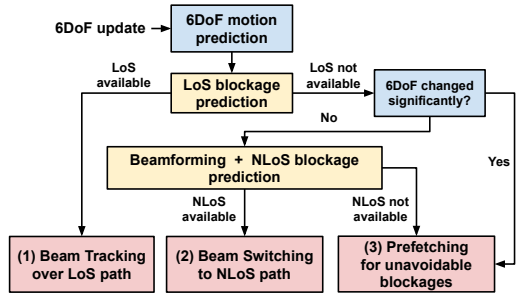


Fig. 9: Beam adaptation, blockage prediction and prefetching based on 6DoF motion prediction.

To determine the inter-user blockage of an NLoS path, we leverage the angle of arrival (AoA) and the angle of departure (AoD) of the paths found through the beamforming (proactively triggered when an LoS path is found to be unavailable). After getting the AoA and AoD of different paths, we can separate the LoS path using the 6DoF motion information of users. In terms of the remaining NLoS paths, their AoA and AoD could be matched by utilizing the method in [26, 77]. In M5, all paths including the NLoS paths are judged using SNR as in the 802.11ad protocol. We regard the midpoint of the shortest distance between the two skew lines of the AP's AoD and the client's AoA of the NLoS path as the reflection point. To determine whether a blocker's mobility will block the NLoS path of the client, we examine the distance d from the blocker to two segments (one from AP to the reflection point and another from the reflection point to the client). If the blocker appears within D_{TH} distance of either of them, the NLoS path of the client is predicted to be blocked.

To validate our blockage prediction model, we utilize the user 6DoF motion traces from the ViVo dataset to simulate the mmWave wireless channel in Remcom Wireless InSite. We then apply our blockage prediction model on the same traces to determine if LoS or NLoS path is available or client experiences a blockage (self or inter-user). We then compare these predictions with SNR values observed in Remcom for 200 instances. The results are shown in Fig. 8a. We find that the accuracy, true-positive, and false-positive rates of our predictions to be 92.9%, 92.1%, and 6.7%, respectively.

4.2 Beam adaptation and prefetching

Fig. 9 shows a flow-chart outlining how and based on which factors the beam adaptation and prefetching decisions are made in M5. When a new 6DoF motion update is available from a client at time t, M5 first runs the 6DoF motion prediction model to predict the client's 6DoF motion at each time slot from t to $t + B\Delta t$ where Δt is the time between two consecutive frames and B is the number of frames in the prediction window. The 6DoF motion prediction is carried out for all clients for inter-user blockage detection. M5 then examines the link status for each consecutive time slot and takes corresponding actions as we describe next.

(1) Beam tracking over LoS path. For each client, M5 first checks the LoS availability between the AP and the newly predicted 6DoF motion of the client using the ray-tracing based self and inter-user blockage model described earlier. This process takes as input the 6DoF motion prediction of all clients for determining potential inter-user blockages. If the LoS path is available for the client p_c at time slot $t + i\Delta t$ where $i \in [1, B]$, AP performs beam adaptation where the AP chooses to switch to a beam from its codebook that provides the maximum gain at (ϕ_a, θ_a), where ϕ_a and θ_a are the azimuth and elevation angles from AP to the client calculated using the AP's 6DoF pose and the client's predicted 6DoF motion. This means that M5 leverages the predicted 6DoF data to directly perform LoS beam tracking without conducting the time-consuming beamforming.

(2) Beam switching to NLoS path. If the LoS path between the AP and the client is not available due to a blockage, M5 tries to check if an NLoS path is available. However, the challenge is how do we determine the AoD for the NLoS path for a client at time slot $t + i\Delta t$ using its predicted 6DoF motion. Prior works such as [67, 76, 77] address this challenge by first locating the reflectors in the environment through senses. However, the sensing can not only be inaccurate but has to be repeated periodically, resulting in high overhead. Instead, M5 proactively triggers the beamforming process in such cases to find the NLoS path.

Even when relying on beamforming to find the NLoS path, beam coherence can be an outstanding problem. Here, M5 predicts at time t that LoS blockage will occur at $t + i\Delta t$ and performs beamforming to find the NLoS path at time t. However, performing the beamforming at t might not be accurate since the best beam found at time t might be different from that at time $t + i\Delta t$ due to client mobility. We must ensure beam coherence between the two-time instances. We leverage the change in 6DoF[3] to predict this beam coherence [71]. We claim that if the change in 6DoF between the time t and $t + i\Delta t$ is not significant, the available mmWave paths between the AP and the client remain the same. This means that the NLoS sector found through beamforming at time t can be used correctly at $t + i\Delta t$. Finding what can be considered an insignificant change is challenging, and we rely on empirical analysis to determine thresholds. Since the beam coherence depends on a range of factors including beamwidth, we use a change of 1 dB to indicate a possible change of beam. We choose 1 dB as the threshold since existing mmWave WLAN standards such as 802.11ad/ay use that as a threshold as default to trigger beamforming. We use Remcom Wireless InSite [58] mmWave channel simulator to calculate the channel matrix H for all users and their 6DoF traces for the ViVo dataset. Fig. 8b shows the change of 6DoF for clients when their SNR changes by no more than 1 dB. We find that median translational movements are 7.1 cm, 2.7 cm and 8.6 cm for X, Y, and Z, and the median rotational movements are 7.4°, 6.8° and 9.2° for yaw, roll, and pitch, respectively. Hence, we use these median values as thresholds to determine whether a client's 6DoF change is significant or not. We note that these values approximately correspond to 6DoF prediction errors for window of length 200ms as per Fig. 6. This means that our 6DoF prediction errors are relatively smaller compared to the movements observed within the same mmWave beam, not requiring unnecessary beamforming due to motion prediction errors.

If the 6DoF change for a client is not significant, the client is scheduled to perform beamforming in the next time slot (i.e., $t + \Delta t$) and the NLoS paths found will be leveraged at $t + i\Delta t$ when the LoS blockage is predicted to occur. This proactive beamforming helps the AP to maintain uninterrupted connectivity to the client using beamswitching. On the contrary, if we find after performing beamforming that the NLoS is unavailable or the NLoS will be blocked by other users at time $t + i\Delta t$, M5 does not employ beamswitching. Instead, M5 considers this situation as an unavoidable blockage and starts the prefetching process as we discuss next.

(3) Prefetching for unavoidable blockages. There are two situations that trigger prefetching in M5. One is when both LoS and NLoS are predicted to be blocked for $t + i\Delta t$ even after conducting

[3] In this paper, we use the terms 6DoF, 6DoF motion and 6DoF pose interchangeably.

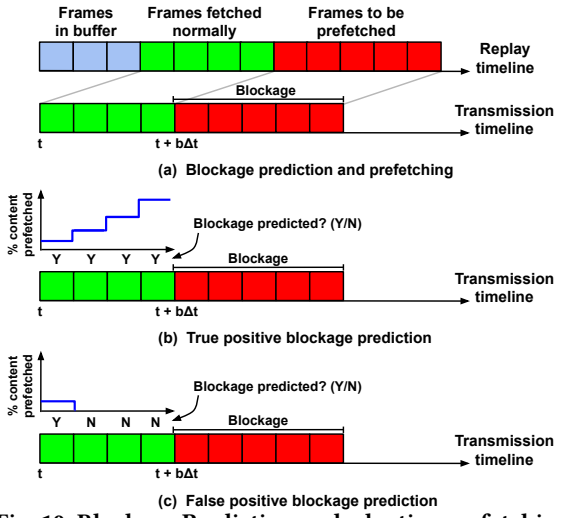


Fig. 10: Blockage Prediction and adaptive prefetching

the beamforming at time t. Another situation is when the LoS path is unavailable and the client has a significant 6DoF change between t and $t + i\Delta t$ (refer to Fig. 9). In this scenario, we cannot use the NLoS path found using beamforming at time t at $t + i\Delta t$ since the best beam found at t could be likely changed at $t + i\Delta t$ due to a relatively larger 6DoF change. Therefore, M5 also considers this as an unavoidable blockage and relies on prefetching to ensure continuous video playback without stalls.

Frames to prefetch. Fig. 10(a) shows an example of unavoidable blockage where prefetching would be necessary. Here, for a given client, while evaluating the link status for each slot in the prediction window B, we find that an unavoidable blockage event is expected to start at time $t + b\Delta t$ and the blockage duration is expected to be BLK time slots. The blockage is anticipated to affect the transmission of frames shown in red in Fig. 10(a). These frames have to be prefetched before the start of the blockage (i.e., between t and $t + b\Delta t$). These prefetched frames have to be transmitted to the client along with the normally fetched frames (green frames in Fig. 10(a)) between t and $t + b\Delta t$. M5 prefetches these frames with maximum allowable quality such that their transmission can be completed before the start of the blockage. We defer the discussion on video quality adaptation to Section 5.2 and discuss the adaptive nature of prefetching first.

Blockage and viewport prediction error. The effectiveness of prefetching relies on how accurately we are able to predict blockages as well as the content to be prefetched itself. However, there are two outstanding challenges of prefetching the blocked frames accurately and efficiently: blockage prediction errors and viewport prediction errors. If the blockage prediction is inaccurate, prefetching would lead to transmission of frames that are possibly unnecessary, adversely affecting the quality at which the normally fetched frames are transmitted. Furthermore, inaccurate 6DoF prediction can result in viewport prediction errors which can then lead to prefetch of inaccurate cells. Both blockage and viewport prediction rely on 6DoF motion prediction which has a larger error with longer prediction windows (Fig. 6). *This means that on one hand, we want to prefetch as close to the blockage start as possible for the prefetching to be more accurate, while on the other hand, we want to start prefetching reasonably before the start of blockage so that there is enough time to prefetch the blocked frames.*

Adaptive prefetching rate. To deal with this problem, M5 proposes an adaptive prefetching scheme by gradually increasing the prefetching rate as it approaches the start of the blockage. The idea here is that when the prediction time t is far away from the predicted blockage starting at $t + b\Delta t$, M5 chooses to prefetch a smaller portion of the total frames to be prefetched because of the high probability of blockage and viewport prediction errors. As it approaches closer and closer to the blockage start time, M5 has a higher probability of a more accurate blockage and viewport prediction. It then adaptively increases (or decreases) the prefetching rate depending on how likely is the blockage.

Time before the start of the blockage (i.e., $b\Delta t$ in s)	Percentage of total prefetched content (%)
2.0 - 1.5	11
1.5 - 1.0	32
1.0 - 0.5	66
0.5 - 0.33	85
0.33 - 0.2	97
< 0.2	100

Table 2: Adaptive prefetching rates calculated for the ViVo dataset

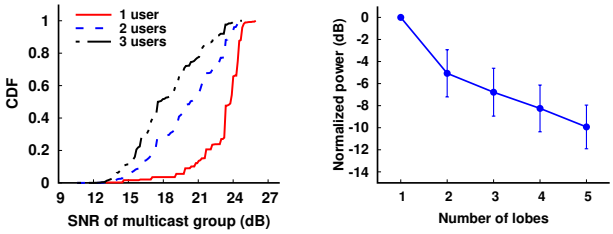


Fig. 11: (a) The default beams cannot support an efficient multicast (b) Reduction in peak power for multi-lobe beams

Table 2 shows the total amount of prefetched content (in percentage) for different time intervals ($b\Delta t$) preceding the start of the blockage in the ViVo dataset. Here, the prefetching percentage χ is calculated as $1 - (e^b_m - \min e^b_m)/(\max e^b_m - \min e^b_m)$, where e^b_m is the median of blockage prediction error for different time intervals $b\Delta t$. χ increases as we approach the start of the blockage. For example, the time before the start of the blockage is 1s and if the blockage event is predicted, 66% of the total prefetched content should be transmitted before 0.5s is remaining till the blockage. Figs. 10(b) and (c) show the true positive and false positive situations of the blockage prediction and corresponding prefetching in M5. In Fig. 10(b), the blockage is consistently predicted to occur between t to $t + b\Delta t$, resulting in an increasing amount of content being prefetched over these slots. On the other hand, in Fig. 10(c), the blockage is predicted only during the first slot $t + \Delta t$. Given that it is far from the start of the blockage, only a small amount of content is prefetched. No blockages are predicted in subsequent slots, resulting in no prefetched content in these slots.

5 CONTENT SIMILARITY BASED MULTICAST

We now develop customized multi-lobe beams that can be used for content similarity-based multicast. We then propose a scheduler for mmWave WLANs that combine the multicast and unicast transmissions while performing 6DoF prediction based beam adaptation and blockage-aware prefetching.

5.1 Customized multi-lobe beam design

While we can leverage the 6DoF prediction of users to generate beams providing high gain towards the users and increase the SNR, multicasting with mmWave requires designing beams that provide high gain in multiple directions simultaneously. We find that the default vendor-provided codebooks are not designed to provide such multi-lobe gains. Another challenge is that the multicast rate is limited by the lowest possible modulation and coding scheme (MCS)

among all concurrent users in the group. If sufficiently high SNR cannot be guaranteed for all multicast group members, multicast can perform even worse than unicast. Therefore, we not only need to design customized beams that can provide gains in multiple directions but the gains should be balanced such that a balanced SNR can be guaranteed for multicast group users.

We first investigate the effectiveness of using the default vendor-supplied codebook used in the commercial 802.11ad devices for multicasting. We use the 6DoF traces collected in the ViVo dataset and measure the SNR for different users with our mmWave WLAN testbed. Our mmWave testbed includes an 802.11ad router from Airfide [5] with 8 phased antenna array patches (shown in Fig. 13d) and multiple Acer laptops [4] with 802.11ad NIC. We modify the open-source 802.11ad driver on the laptop to extract the SNR, MCS, and beamforming information to userspace. Fig. 11a shows the CDF of maximum SNR that could be supported by the default codebook for multicast groups of different sizes. We find that using the default codebook, the median SNRs for three different group sizes are 24.2, 20.4, 16.7 dB, respectively, which can provide MCS 8, MCS 3, MCS 1 (as per 802.11ad table [24]) for multicast. However, the MCS 3 and MCS 1 can only provide 363 Mbps and 112 Mbps average data rate in our measurements, which is not sufficient for multi-user volumetric video streaming. Since the default codebook beams are not explicitly designed to support multicast, it cannot guarantee high SNR to all users in multicast groups.

To address the problem, we propose to create multi-lobe beams by weighting the antenna vectors of beams pointing to individual users while constraining the total power. Assume the optimal antenna vector used for transmitting to user i in a multicast group S is $\mathbf{w}_i$, and the corresponding SNR is snr_i. We define the combined antenna weight vector for the new beam as

$$\mathbf{w} = \frac{\sum_i^{|S|} snr_i \cdot \mathbf{w}_i}{\sum_i^{|S|} snr_i} \tag{1}$$

The combined weight vector keeps the lobe direction of the individual beams while re-assigning the power of each lobe to provide a balanced SNR for users in the multicast group. Therefore, it can not only cover users at different locations but also provide a high common MCS to them. Our custom beam design shares the idea with the coherent phase alignment [25, 74], but it requires just the SNR value of single-lobe beams to normalize the antenna vectors. Since different users have separate RF chains, a detailed channel state information (CSI) is not needed in our case, making the multi-lobe design more efficient in terms of measurement overhead.

When increasing the number of lobes in a customized beam, the power assigned to each user decreases, given that the total transmission power is restricted by FCC Maximum Effective Isotropic Radiated Power (EIRP) limit. To understand this impact, we measure the average of peak power of all lobes in multi-lobe beams. Fig. 11b shows the average of peak power as the number of lobes increases when using the combined vector for the weight calculation with a 6×6 uniform rectangular array (URA). We find that compared to the single lobe, the two and three-lobe power is lower by 4.9 dB and 6.8 dB, respectively. Increasing the number of lobes to four reduces the power by 8.3 dB. Therefore, we limit the number of users in a multicast group to no more than 3 in M5.

5.2 Multi-lobe multicast scheduling

In the multi-user multicast volumetric video streaming scenario, we can formalize the scheduling problem as an optimization problem that maximizes the video quality for the requested volumetric cells subject to the constraint of meeting the required frame rate. However, different user groups of multicast transmission lead to distinct viewport similarities (within the required cells) and different SNR values in mmWave WLAN, which results in different transmission latency for users. Therefore, we must schedule the multicast transmissions intelligently by selecting multicast groups with low transmission latency. In M5, we choose multicast groups based on the similarity of the users' viewports and the transmission efficiency (achievable SNR) of the corresponding multi-lobe beams.

Let us assume that the total cell size of a requested frame for user i at video quality q is F_i^q. In terms of the point cloud volumetric frame, M5 considers q = 1 to 6 to represent different points density (10%, 20%, 40%, 60%, 80%, and 100%) of the original frame. The corresponding data rate at the current time slot for user i is r_i after the beam adaptation. Then we can estimate the transmission time for user i and video quality q as $t_i^q = F_i^q / r_i$. Suppose we use unicast to transmit the frame to all concurrent users. In that case, the total time for transmitting the frame is the summation of the transmission time for all users, including transmitting a large amount of redundant content, limiting the highest video quality users can watch.

In terms of the multicast, we can estimate the transmission time of a multicast group S being served with video quality q as

$$T^q(S) = F_S^q / r_S + \sum_i^{|S|} (F_i^q - F_S^q) / r_i \tag{2}$$

where F_S^q is the size of the overlapped cells for user group S being requested at video quality q and r_S is the multicast data rate determined by the multi-lobe beams for the multicast group S. In above equation, the first part is the time to transmit the overlapped cells using multicast transmission, while the second part is the transmission time to stream the remaining requested cells using unicast to each user.

It is clear that multicast group selection affects both the multicast content and the data rate. Among a set of users U, we would like to find the set of groups S_{min} that has the minimum total transmission time $T = \sum T^q(S)$ where $S \in S_{min}$. We then increase the video quality q until the total transmission time $T > 1/FPS$ where FPS is the required frame rate, and the constraint $\sum T^q(S) <= 1/FPS$ guarantees that there is no stall for all multicast groups in the group set S_{min}. The searching space for finding the optimal grouping to minimize the transmission time is bounded by $O(N^N)$ where N is the total number of users.

As discussed in Section 5.1 and Fig. 11b, we restrict the number of lobes to 3 in multi-lobe design, resulting in the maximum multicast group size being 3 as well. Hence, the complexity of finding the optimal group that can maximize the video quality while meeting the frame rate is bounded by $O(N^3)$. To achieve efficient multicast group selection, we propose a scheduler to select the multicast group based on their viewport similarity and gradually improve the video quality while adhering to the frame-rate limitation.

M5's scheduler. Our scheduler schedules the transmissions at each time slot (every 33 ms at a frame rate of 30 Hz). At each time slot, M5 scheduler first determines the set of pending frames $\mathbb{F}$ (normally fetched for buffering as well as prefetched) based on the 6DoF motion prediction and prefetching schemes discussed earlier. These set of frames $\mathbb{F}$ with a set of users $\mathbb{U}$ becomes input to our scheduler. The output of our scheduler is a group of transmissions for the pending frames, along with its quality and the set of beams (single lobe for unicast and multi-lobe for multicast) used to deliver these frames to the users. The scheduler tries to form multicast

groups when multiple users request the same content in order to reduce the transmission redundancy and improve the video quality within the time slot (i.e., the required frame rate is maintained). Here, for each frame that is being requested by multiple users, all possible multicast groups are first calculated, and then the group that achieves the highest SNR (for all users in the group based on the multi-lobe beam pattern created for that group) is chosen for the multicast transmission. A user can become part of a different multicast group at a different time depending on the frame(s) being requested at that time. The intuition here is that if we select the group that achieves the highest SNR, the resultant multicast transmission will take the smallest transmission time, reducing the video stall. To achieve this, our scheduler runs the following steps every time slot.

(1) Sort the pending frames in ascending order of the frame index. For each pending frame $F_j \in \mathbb{F}$, find the set of users $U_j \subset \mathbb{U}$ requesting the same frame. If U_j only has one user, go to Step 2, otherwise go to Step 3.

(2) Schedule a unicast transmission with the default beam (found using the 6DoF-based beam adaptation) at the lowest quality ($q = 1$) for the user frame F_j. Estimate the transmission time T_j. Go to Step 1 if there are more pending frames in $\mathbb{F}$ or else go to Step 5.

(3) Schedule a multicast transmission for the set of users U_j. The total number of possible multicast groups equals the number of ways one can partition U_j into subsets of size x where $x \in [1, |U_j|]$ and $x \leq L$ where L is the maximum number of users allowed in a multicast group. As stated earlier, we utilize $L \leq 3$ in our experiments due to decreasing similarity among users and lower SNR of multi-lobe beams with increasing group size. We obtain a set of candidate partitions $\mathbb{S}$ by calculating the Stirling numbers of the second kind for the set U_j. For each candidate partition $S_c \in \mathbb{S}$, go over all groups $S \in S_c$ in the partition. Then for each group S, calculate the overlapped volumetric content F_S^q at the lowest quality $q = 1$. Note that the partition S_c can be a combination of multi-user groups and single user groups which are served using multicast and unicast respectively.

(4) For each multi-user group $S \in S_c$, find the corresponding multi-lobe beams b_S using Equ. 1 by combining the individual weight vectors for users in S. By probing the multi-lobe beam, we can get the SNR value and estimate data rate r_S. Then we estimate the transmission time $T^q(S)$ for this group S based on the Equ. 2. The total transmission time for the candidate partition S_c is the summation of the transmission time of all groups belonging to it. Among all candidate partitions $S_c \in \mathbb{S}$, select the partition S_{min} with the *minimum total transmission time* T_j for multicasting the current frame F_j. Go to Step 1 if there are more pending frames in $\mathbb{F}$ or else go to Step 5.

(5) Let $T = \sum_j^{|\mathbb{F}|} T_j$ be the total transmission time for all pending frames in $\mathbb{F}$. If T is less than the frame rate constraint $1/FPS$, keep increasing q to the next video quality level until $T > 1/FPS$.

(6) Lastly, transmit with multicast for multi-user and unicast for a single user for each pending frame $F_j \in \mathbb{F}$ with quality q_j. Go to Step 1 to scheduling the next time slot.

6 PERFORMANCE EVALUATION

We first utilize 6DoF motion traces from two datasets, ViVo [20] and FHHI [14] (Table 1, the same datasets used in motivation experiments in Section 2), to analyze the performance of our blockage prediction model in Section 6.1. We then implement and evaluate our multi-lobe beam design on our COTS 802.11ad 60 GHz mmWave testbed in Section 6.2. Lastly, using trace-driven simulations (6DoF traces and mmWave channel traces), we evaluate M5 's blockage mitigation scheme including beam adaptation and prefetching in Section 6.3 and end-to-end performance in terms of video frame rate and quality in Section 6.4.

Our mmWave experimental and trace-driven evaluation involves the following steps. We simulate the Airfide AP (shown in Fig. 13d) equipped with eight patches of phased antenna arrays from our testbed in Remcom Wireless InSite mmWave channel simulator [58]. Here, we place the AP in a $10m \times 10m$ empty room at a height of $2m$. We then use the Remcom to simulate fine-grained channel matrices for 6DoF motion traces of users. We use an AWS server (3.6GHz 18-cores with 32 GB RAM) for approximately 16 days to get more than 320K channel instances for 30 different users. We then use the AP in our testbed to create single and multi-lobe beam codebooks. These codebook weight vectors are then applied to the channel matrices calculated from Remcom for different 6DoF traces to calculate the SNR values. The SNR values of the traces are then used for all modules of M5, including beam adaptation, blockage prediction, prefetching and multicast schedule. The benefit of this methodology is that it enables us to evaluate different variations of our proposed schemes as well as other alternative schemes on the same set of 6DoF traces in a more reproducible manner.

6.1 Blockage prediction

Characterizing blockages with 6DoF motion traces. We first characterize the viewport of two real world 6DoF motion datasets (ViVo and FHHI) in terms of the availability of LoS and NLoS paths along with the probability of blockages in mmWave WLANs. Depending on users' 6DoF mobility, there can be self blockages as well as inter-user blockages, resulting in different availability of the LoS path and NLoS paths.

Based on our path estimation and blockage prediction models presented in Section 4.1, we define three probabilities to characterize the blockage events. Each probability is defined as $\frac{\sum_1^F \sum_1^N \mathcal{I}}{F \times N}$ where F is the number of frames of the video, N is the number of concurrent users watching the video together, and $\mathcal{I}$ is a binary variable presenting different events. The LoS probability is defined as the probability where $\mathcal{I} = 1$ represents the LoS path is available for the corresponding user. When the LoS path is available, M5 tracks this LoS path with the predicted 6DoF data and directly adapts the mmWave beam accordingly. Similarly, the NLoS probability is defined as the probability where $\mathcal{I} = 1$ represents the LoS is not available but at least one of the NLoS paths is available for the corresponding user. To utilize the NLoS path, M5 examines the channel stability using 6DoF changes and conducts beamforming to find the NLoS path across the blockage proactively. The last is the blockage probability where $\mathcal{I} = 1$ represents that none of the LoS and NLoS paths are available for the corresponding user. M5 tackles this complete blockage using adaptive prefetching.

Figs. 12(a)-(c) show the CDF of the three probabilities for the ViVo dataset and FHHI dataset, respectively. We observe that the LoS probability decreases with the increasing number of concurrent users due to the increasing number of blockages. For instance, for five concurrent users, the average LoS probability is 54% and 31% for ViVo dataset and FHHI dataset, which shows that only 54% and 31% of the total 6DoF poses can be transmitted using the default LoS path in the five users scenario. Fig. 12b shows the probability of the NLoS path which can be used to proactively switch to. On average, 36% and 45% of the 6DoF pose for five users in ViVo and FHHI dataset could use a NLoS path. Lastly, there is a large number of 6DoF poses (up to 21% and 32% for five users as shown in Fig.12c)

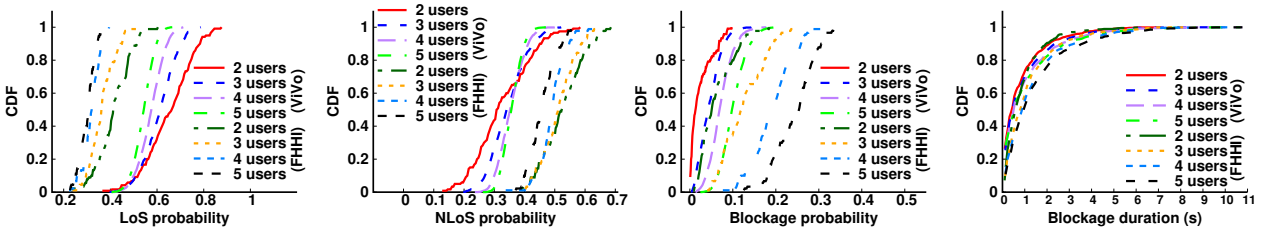


Fig. 12: Characterization of LoS, NLoS, blockage probability and blockage duration in multi-user scenario

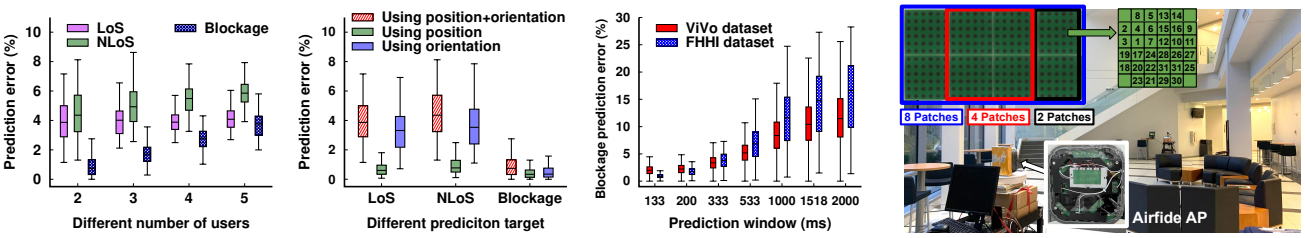


Fig. 13: (a-c)M5 **blockage prediction performance in multi-user scenario; (d)Airfide AP with uniform rectangular array (URA)**

where none of the LoS and NLoS paths are available for transmitting when users watch the volumetric frames. These frames need to be prefetched to reduce the video stall.

We also characterize the blockage duration with ViVo and FHHI 6DoF traces in Fig. 12d. We observe that an increasing the number of users will result in some increase in the blockage duration. Increasing the users from 2 to 5, the 75th percentile of the blockage duration increases from 1089 ms to 1353 ms for ViVo dataset and 1122 ms to 2079 ms in the FHHI dataset.

Performance of blockage prediction. M5 's path estimation and the blockage prediction model are based on the 6DoF motion prediction. To evaluate the performance, we run the model on ground truth 6DoF traces and predicted 6DoF traces separately and calculate the prediction error for LoS, NLoS, and blockages. The prediction error is calculated as (false position + false negative) / total number of predictions. Fig. 13a shows the prediction error (using 200*ms* prediction window) with the increasing number of concurrent users for LoS, NLoS, and blockage prediction. We observe that the LoS prediction error does not rise when increasing the number of users, while both NLoS and blockage prediction errors increase with the increasing number of users. It is because, with the increasing number of users, the NLoS and blockage events increase, lowering the probability of accurate prediction. However, our LoS, NLoS, and blockage prediction are accurate with a median prediction error for all three being 3.8%, 4.3%, 0.8% and 4.1%, 5.8%, 3.6% for groups of two users and five users, respectively.

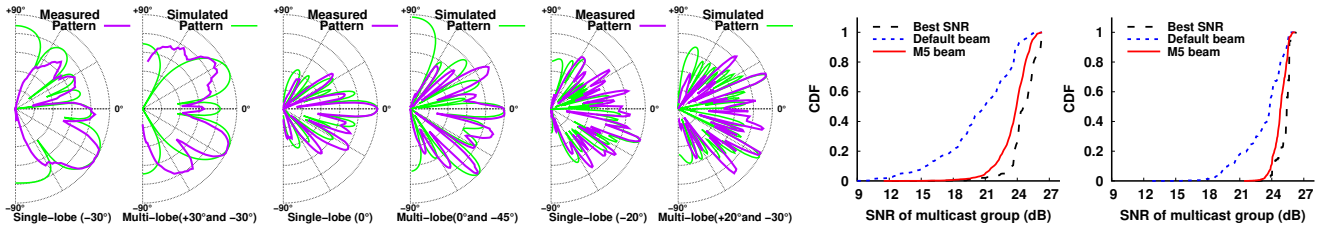
Since volumetric videos enable the 6DoF movement of users, the prediction of each dimension might have different impacts on the path estimation and blockage prediction. We separately investigate the effect of the translational (X, Y, Z) and rotational (yaw, pitch, roll) movement on the prediction error. We show the result of LoS, NLoS, and blockage prediction error when only using position, rotation, and using both in Fig. 13b. We observe that prediction errors in orientation play a more important role in terms of path estimation. It shows that compared to the small translational error, the small rotational error causes larger changes to the path estimation in the mmWave WLAN. While in terms of the blockage prediction, the position and orientation errors negatively affect the prediction.

Lastly, we show the blockage prediction error with different lengths of prediction windows for both ViVo and FHHI datasets in Fig. 13c. As expected, the blockage prediction error increases with a longer prediction window. For example, with the 200*ms* prediction window size, ViVo and FHHI datasets have less than 2.7% and 1.8% median blockage prediction error. When the prediction window size increases to 2*s*, both error rates rise to 14.5% and 16.6%. As mentioned in Section 4.2, we utilize this observation to facilitate the adaptive prefetching scheme only to increase the prefetching rate when we gradually approach the start of blockage (i.e., shorter prediction window and hence a more accurate prediction).

6.2 Multi-lobe design

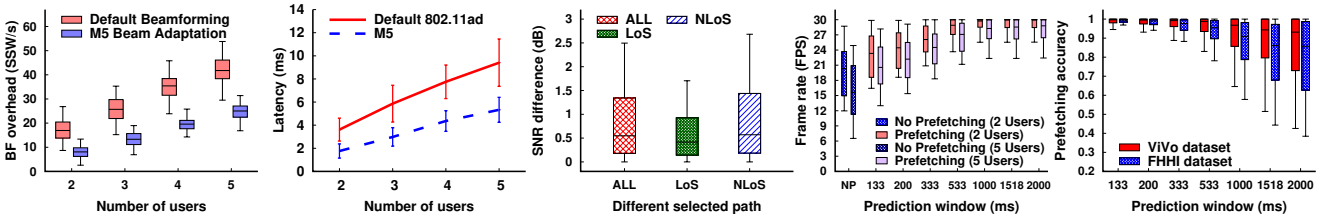
Multi-lobe beam implementation on COTS phased arrays. To evaluate M5 's multi-lobe design, we utilize a COTS 802.11ad mmWave AP from Airfide [5] which is equipped with Qualcomm QCA9500 chipset (QCA6335 baseband and QCA6310 RF transceiver) and eight 6×6 uniform rectangular array (URA) patches as shown in Fig.13d. The control structure of the Airfide URA is similar to the one shown in [47, 92]. Each element of the URA is controlled using a 2-bit phase-shifter and a 1-bit amplitude switch (enabling/disabling the antenna element). To generate single-lobe beams pointing in different directions and then combine them into multi-lobe beams used for multicast in M5, we first simulate the beams in Matlab (using antenna toolbox [70]) and map the corresponding weight vectors on the Airfide URA. Through extensive calibration measurements, we first determine the index of URA antenna elements) one by one and then apply the antenna weight vectors. The measured element indices are shown in Fig. 13d.

Fig. 14(a-c) show the simulated as well as measured single and multi-lobe patterns for 2, 4 and 8 patches. To measure the Airfide URA gain patterns, we use a Vubiq RF frontend [73] equipped with a 1.5° horn antenna and connected it to a baseband signal analyzer. We then rotate the Airfide AP using an automatic rotator and measure signal strength at 1° resolution. We repeat the process for each antenna gain pattern (single and multi-lobe) created through the different antenna patches. As shown in Fig. 14(a-c), the effective beams have 34°, 17°, and 8.5° main-lobe half-power beamwidth (HPBW) for 2 patches (6 elements on the horizontal plane), 4 patches (12 elements on the horizontal plane), and 8 patches (24 elements on the horizontal plane), respectively. We can observe that the main lobes of the simulated and measured antenna patterns match well, and the patterns have a strong similarity. The differences can



(a) HPBW ≈ 34° (b) HPBW ≈ 17° (c) HPBW ≈ 8.5° (d) ViVo dataset (e) FHHI dataset

Fig. 14: (a-c) M5 single-lobe and multi-lobe patterns with different HPBW. (d,e)M5's multi-lobe beams improve SNR for multicast.



(a) (b) (c) (d) (e)

Fig. 15: M5's blockage mitigation (a) BF overhead, (b) latency, (c) SNR difference, (d) frame rate, and (e) prefetching accuracy.

be mostly attributed to phase quantization as the Airfide AP can only provide four phase values (0, $\pi/2$, π, and $3\pi/2$) compared to continuous phase values used in the simulation.

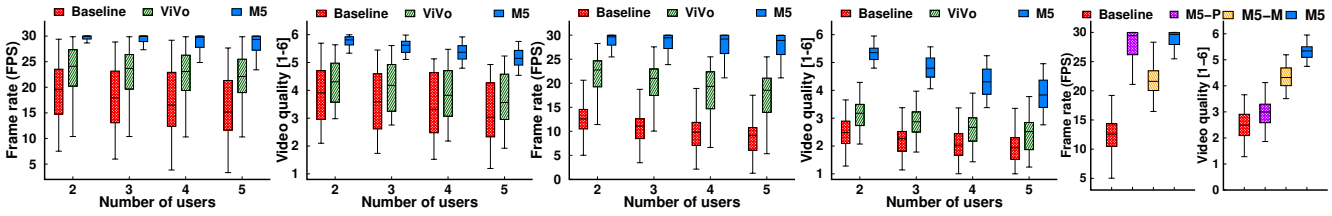
SNR improvement of multi-lobe beams for multicast groups. We next evaluate how effectively the multi-lobe beams can support the users within different multicast groups. To this end, we put different multicast groups of two users at different positions and orientations corresponding to the 6DoF data in ViVo and FHHI datasets. The users utilize Acer laptops [4] (equipped with similar 802.11ad NIC as Airfide but different phased array NGFF595A-L-Ant [43]) as the receivers. We then extract the SNR, MCS and sector information from the wil6210 [78] driver to user space. On the AP side, we use Airfide with our customized multi-lobe patterns to generate beams pointing to the users. In Fig. 14d and 14e, we show the CDF of SNR values with all tested 6DoF combinations with the default beams provided by the 802.11ad vendor codebook (not optimized for multicast) and our M5 's multi-lobe beams. We also show the best SNR of the single user in the multicast group. Exploiting the M5 's customized beams, the users can achieve an average 5.6 dB and 3.4 dB gain in the low SNR range (bottom 30% SNR) compared with the default beams when using 6DoF poses from ViVo and FHHI datasets in our testbed. In addition, the multi-lobe beams can achieve comparable SNR as the single user's best SNR case (0.6 dB and 0.5 dB SNR difference on average) to support multi-user in the multicast group. These higher SNR values can lead to higher common MCS and throughput in mmWave WLANs.

6.3 Cross-layer blockage mitigation

Beam adaptation. Current commercial 802.11ad/ay devices (including from Airfide [5]) reactively triggers the beamforming when the SNR value of the current beam changes by 1 dB as default. On the other hand, M5 utilizes the 6DoF prediction to find the LoS path and proactively adapt the beam to the LoS path without the time-consuming beamforming. In the case of NLoS, when the predicted 6DoF has not changed significantly, M5 conducts beamforming and NLoS blockage prediction to find the available NLoS path. M5 also needs to probe the custom-designed multi-lobe beam before initiating the multicast transmission. However, this probing overhead is much smaller (probing one beam compared to all beams of the codebook in typical SSW). Fig. 15a shows the beamforming overhead collected on our 802.11ad COTS testbed in terms of the number of sector sweeps per second (SSW/s) with increasing users. As shown in Fig.15a, M5 reduces beamforming overhead from 16.9 SSW/s to 8.1 SSW/s for two users, and from 41.8 SSW/s to 25 SSW/s for five users. On average, M5 results in 46.3% beamforming overhead reduction for different numbers of users. After the SSW, we switch to the sector with the best SNR value via the *wmi* command in real-time.

We now evaluate the latency of M5. The dominant component affecting the latency is the overhead of beamforming. In M5, beamforming is required for finding the NLoS paths to go across predicted blockages and for evaluating the SNR of single-lobe and multi-lobe beams. The beamforming overhead increases with the number of users as it has to be performed in sequence for each user. On our testbed, each SSW takes 2.41 ms for one user considering 64 sectors to be searched on the AP. We find that other components of M5 such as the blockage prediction calculation and multicast scheduler incur very small latency (< 1ms on typical off-the-shelf laptop [4] with 2 cores). Fig. 15b compares the beamforming latency of M5 and the default 802.11ad protocol. Given that the time consumed by an AP for performing beamforming results in packets being queued (increasing queuing delay on the AP), fewer sector sweeps translate to lower link latency. M5 utilizes 6DoF motion prediction to reduce the number of SSWs, resulting in lower latency of the system. In comparison, 802.11ad requires much more frequent beamforming, increasing the latency. Fig. 15b shows that M5 can reduce the latency by 51.2% and 43.4% for two users and five users, respectively. We also note that the latency can be further reduced by searching fewer sectors in each SSW based on the predicted 6DoF motion.

Another important factor in evaluating 6DoF based beam adaptation is that when the predicted 6DoF has errors, it might result in a selection of a suboptimal beam. To understand this effect, we first compare the best sector selected by M5 based on the ground truth and predicted 6DoF motion. We find that, on average, in 93.2% of the instances, the best sector selected by M5 using ground truth and the predicted 6DoF motions are the same. For the remaining instances when the chosen best sectors are different, we calculate the SNR difference between the best sector selected based on the ground truth motion and the suboptimal sector selected based on the predicted 6DoF motion. Fig. 15c shows the SNR difference. We find that the median SNR difference is 0.54 dB, where the difference for LoS and NLoS paths is 0.41 dB and 0.57 dB, respectively. The higher SNR difference of the NLoS paths is due to the higher NLoS



(a) Frame rate (ViVo dataset) (b) Video quality (ViVo dataset) (c) Frame rate (FHHI dataset) (d) Video quality (FHHI dataset) (e) FHHI dataset

Fig. 16: M5 can effectively improve the frame rate and video quality

prediction error, as shown in Fig. 13a. Overall, we find that the 6DoF-based beam adaptation can reduce the beamforming overhead and guarantee a comparable SNR.

Prefetching. In the situations of unavoidable blockages, M5 adaptively prefetches frames ahead of the predicted blockage, as described in Section 4.2. Fig. 15d shows the frame rate (capped at 30 fps) for two and five concurrent users scenario without and with prefetching as we increase the prediction window. We observe that our prefetching scheme significantly improves the frame rate by prefetching the frames that would be affected by blockages. We find that the median frame rate of two users is 19.8 fps without utilizing the prefetching, while with prefetching, the frame rate increases to 24.4 fps, 28.7 fps and 29.8 fps for the prediction window of 200 ms, 533 ms, and 1000 ms, respectively. Since a longer prediction window provides more time for prefetching, the frame rate increases correspondingly. While increasing the number of users reduces the frame rate, Fig. 15d shows that the prefetching scheme can still improve the frame rate by 41.1%, 72.6% and 80.2% for 200 ms, 533 ms, and 1000 ms prediction windows, respectively in five users case. We note that using content similarity-based multicast is another key way of increasing frame rate with more users. We discuss these results later in the end-to-end evaluation.

While a longer prediction window helps to prefetch more volumetric content in time, it also results in higher error in content prediction (i.e., what volumetric cells in frames are being prefetched). We calculate the prefetching accuracy as the number of cells correctly prefetched and correctly not prefetched over the total number of predicted cells in each frame and show M5 's prefetching accuracy with different sizes of the prediction window within two datasets. As shown in Fig.15e, the longer prediction window leads to a higher 6DoF prediction error and lower prefetching accuracy. The median prefetching accuracy is 97.1% and 91.1% for 1*s* and 2*s* prediction window within ViVo datasets. FHHI dataset has lower prefetching accuracy, which is 94.2% and 85.9% for 1*s* and 2*s* prediction window.

6.4 End-to-end performance of M5

To evaluate the end-to-end performance of M5 in volumetric video streaming scenarios, we investigate the frame rate and video quality using trace-driven simulation for different schemes. (1) The **Baseline scheme** utilizes the default 802.11ad beamforming to find the best SNR beam for all users and transmit the required volumetric content to all users with the unicast transmission, (2) **ViVo**: it uses the viewport, distance, and occlusion optimization to reduce the required content size compared to the baseline scheme. We implement ViVo for multiple users with unicast transmissions for our comparison. (3) Our proposed scheme **M5** uses 6DoF-based beam adaptation, blockage-aware prefetching, and content similarity based multicast along with ViVo's volumetric content optimization.

Fig. 16a and Fig. 16c show the frame rate for ViVo and FHHI datasets for the three schemes. We find that M5 significantly improves the frame rate compared to the baseline unicast and ViVo. The median frame rates for the two user cases are 19.5 fps, 24.1 fps, and 30 fps in ViVo dataset, and 12.6 fps, 22.7 fps and 29.6 fps in FHHI dataset. The ViVo scheme improves the frame rate by significantly reducing the required bandwidth. Hence, it can support more frames when users experience low SNR value. However, it cannot predict and mitigate the blockage effects, so the blockages in the multi-user scenarios still deteriorate the frame rate for ViVo. On the contrary, M5 can predict the blockage events and prefetch frames to mitigate the blockage events. In addition, it further reduces the required bandwidth when multiple users share the overlapping volumetric content. As a result, M5 can keep the high frame rate for all users. With the increasing number of users, all schemes experience frame rate reduction because of the growing size of required content and the frequent blockages among users. However, M5 can still provide a reasonably high frame rate.

Figs. 16b and 16d show the video quality. We observe that M5 outperforms the ViVo and baseline scheme. Specifically, M5 can achieve on average 59.4% and 109.3% quality improvement in ViVo and FHHI dataset, respectively. M5 provides higher gains for both frame rate and video quality with FHHI dataset than the ViVo dataset because of the higher blockages in FHHI dataset (Fig. 12c). We note that the maximum number of users supported by M5 depends on various factors including mmWave link bandwidth, inter-user interference, design of phased antenna array (for creating multi-lobe beams), users' expected video content quality, and even their 6DoF motion. While our results indicate that M5 can serve up to 5 users with high-resolution video without any stalls, if the required quality is lower or viewport similarity is significantly higher, more users can be supported. For instance, among the ViVo dataset user traces, if all users are stationary without any blockages and just need the low-resolution video quality, M5 can support up to 10 users on average without video stalls.

Lastly, we dissect M5 by separately considering the performance gain provided by prefetching (M5-P) and multicast (M5-M). As shown in Fig. 16e, compared to the frame rate improvement achieved through multicast (from 12.6 to 21.1 fps), the prefetching scheme can significantly improve the frame rate (from 12.6 to 29.3 fps). However, prefetching alone achieves a relatively lower gain in video quality (21%) compared to multicast. It is because prefetching can mitigate the blockage effect and reduce the video stall times but it does so with the low-quality content most of the time. On the other hand, multicast can achieve a higher video quality gain (74%) compared to prefetching because it effectively reduces the total content to be transmitted. In conclusion, M5 can provide significant performance improvement for users' frame rate and video qualities compared to the state-of-the-art system.

7 DISCUSSION AND FUTURE WORK

Multi-AP scenario. While M5 utilizes a single AP, the underlying concepts of viewport similarity-based multicast and 6DoF-based blockage prediction can be extended to the multiple-AP scenario. While the added spatial diversity can increase blockage resilience,

extending M5 to multiple APs will require us to address the following two problems. First, depending on the blockages observed by the users of a multicast group, an intelligent AP selection scheme is necessary to provide the highest throughput for the multicast group. However, the computational complexity of AP selection, along with group selection, could be higher in comparison. Second, the beamforming overhead also proportionally increases with the number of APs. This means that the beam adaptation schemes proposed for M5 need to be carefully adapted to reduce this overhead while ensuring high-gain beams from all APs.

Prolonged blockages. In a multi-user scenario, it is possible that all paths of a user are blocked by other users for a long duration. The duration can be long enough such that prefetching cannot be used for the entire duration and video stall becomes unavoidable. A potential solution to the problem could be providing visual hints to the blocked user (through the AR/VR headset) and guiding them to potential alternatives in terms of 6DoF motion that can help them avoid the blockages. We also believe that the use of multiple APs and careful deployments can further alleviate such situations of prolonged blockages.

Multicast implementation. After micro-benchmarking individual components of M5 on our testbed, we utilized a trace-driven simulation to evaluate the end-to-end performance. This is because currently there is only one open-source 802.11ad driver (*wil6210* [78]) in the Linux kernel. This driver, which is compatible with the current 802.11ad chipset (Qualcomm QCA9500 [51]), utilizes *Hard-MAC* where MAC layer functions are implemented in proprietary firmware, impeding multicast implementation on the testbed. In the future, we plan to collaborate with device vendors to develop a complete MAC layer implementation of M5.

Blockage and motion prediction. mmWave links in practice can exhibit partial blockages (reduction in signal strength but not complete outage of the link). M5 uses a binary model for blockage prediction which is relatively conservative in predicting blockages. This is because accurately predicting partial blockages requires exhaustive ray-tracing, which is known to be computationally expensive and time-consuming [28], making it difficult to be adopted in M5 for real-time operations. In the future, we plan to leverage recent advances in computationally efficient ray-tracing models [18, 30, 91] for simulating partial blockages.

8 RELATED WORK

mmWave Communications. There exists a plethora of work on characterizing mmWave links and networks [56, 57, 59, 63, 68, 79, 87, 95] and on beam searching and adaptation for mobility [31, 45, 47, 64, 76, 93], interference [26, 37, 44, 62], and their combinations [19, 21, 60, 67, 74, 85]. Blockage mitigation in mmWave WLANs has been extensively studied with the use of multiple beams [25, 75], multiple APs [85], out-of-band communication [46, 66], reflected paths [2, 3, 69, 77, 94, 96], wider beams [19, 68, 82] and proactive network deployments [81]. However, these solutions only consider the PHY layer information for improving link resiliency. In comparison, we investigate application-aware and cross-layer resiliency solutions that explore upper layer information at MAC/PHY layer (e.g., motion-aware multi-beam adaptation) and vice versa (e.g., blockage-aware prefetching) in M5.

Multimedia Content Delivery over mmWave. Most works used mmWave "as-is" to deliver HD videos [7, 54, 61], 360° videos [49, 65], VR content [2, 3, 33]. MoVR [2, 3] utilizes special antennas and reflectors for VR over mmWave, but it is incompatible with off-the-shelf devices. Pia [76] takes advantage of poses to help users proactively switch to alternative APs in mmWave WLAN to mitigate self-blockages. While M5 not only further considers inter-user blockages in a multi-user scenario, but also focuses on volumetric video streaming by considering the upper layer information such as viewport similarity for multicast transmission.

Multicasting multimedia content over legacy WLAN has been explored [6, 13, 39]. However, directional communication in mmWave WLANs makes multicast a challenging problem, requiring us to develop a multi-lobe multicast solution. Similarly, mmWave multicast has been investigated in prior works [1, 41]. Authors in [1] propose a joint design of hybrid transmit precoders and receive combiners for mmWave multi-group multicasting. Authors in [41] propose to use multi-level codebook for mmWave multicasting. These works are oblivious to upper-layer applications compared to M5 which leverages application specific information (e.g., viewport similarity) to form the multicast groups and serve them over mmWave. Furthermore, M5 is the first system to demonstrate the use of multi-lobe beams for multicasting on COTS mmWave devices. Viewport-based video content prefetching has been widely explored in 360° videos [11, 34, 53, 80, 90]. In comparison, M5 leverages prefetching for volumetric videos. A key difference compared to the prior work is that prefetching in M5 is not only dependent on the user's motion but also on predicted blockages. Such a prefetching facilitated through M5's cross-layer design for volumetric video streaming over blockage-prone mmWave links is still unexplored.

Volumetric Video Streaming. Only a few studies exist on volumetric video streaming [15, 16, 20, 29, 48, 50, 52], given that the research in this area is still in its infancy. Existing work either directly streams encoded point cloud data (e.g., ViVo [20] and GROOT [29]) or uses 3D to 2D content transcoding [15, 16, 52] for a single user. In contrast to the above work, we study the technical challenges of enabling multi-user volumetric content delivery over mmWave. Volumetric video streaming over mmWave was recently explored in [86]. However, M5 aims at creating a holistic system with various novel components, including 6DoF-motion-based beam adaptation, blockage-aware prefetching, multicast multi-lobe scheduling, and end-to-end system evaluation on real 6DoF motion traces.

Multi-user Multimedia Applications. Recent work has started to support multiple users for AR [36, 55, 88, 89], VR [17, 32, 38], and 360° video streaming [8–10, 49]. While these works improve the system performance by focusing purely on the application layer, we propose cross-layer optimizations for multi-user immersive volumetric content delivery and employ motion prediction to facilitate beam adaptation, blockage prediction and multicast grouping.

9 CONCLUSION

In this paper, we propose a holistic system, M5, to stream volumetric videos for multiple users with multi-lobe multicast in mmWave WLAN. Based on the 6DoF motion prediction of multiple users, M5 predicts the blockages and proactively adapts beams if the blockage is avoidable and prefetches frames otherwise. It uses multicast for users with similar content through a customized multi-lobe beam pattern to reduce the required bandwidth. Our experimental and trace-driven simulation results show that M5 can effectively improve the frame rate and video quality compared to the state-of-the-art system.

ACKNOWLEDGMENTS

We thank the anonymous reviewers and the shepherd for their valuable feedback. This research is supported by NSF grants CNS-1815945, CNS-1730083, CNS-2045885, and CNS-2212296.

REFERENCES

[1] Luis F. Abanto-Leon, Matthias Hollick, and Gek Hong Sim. 2019. Hybrid Precoding for Multi-Group Multicasting in mmWave Systems. In *2019 IEEE Global Communications Conference (GLOBECOM)*. 1–7. https://doi.org/10.1109/GLOBECOM38437.2019.9014050

[2] Omid Abari, Dinesh Bharadia, Austin Duffield, and Dina Katabi. 2016. Cutting the Cord in Virtual Reality. In *Proceedings of ACM Workshop on Hot Topics in Networks (HotNets '16)*.

[3] Omid Abari, Dinesh Bharadia, Austin Duffield, and Dina Katabi. 2017. Enabling High-Quality Untethered Virtual Reality. In *Proceedings of USENIX Symposium on Networked Systems Design and Implementation (NSDI '17)*.

[4] Acer TravelMate P648 2016. Acer TravelMate P648. https://www.acer.com/ac/en/US/press/2016/175243.

[5] Airfide 2020. AFN2200. https://airfidenet.com/.

[6] Ehsan Aryafar, Mohammad Khojastepour, Karthikeyan Sundaresan, Sampath Rangarajan, and Edward W. Knightly. 2012. ADAM: An adaptive beamforming system for multicasting in wireless LANs. In *2012 Proceedings IEEE INFOCOM*. 1467–1475. https://doi.org/10.1109/INFCOM.2012.6195513

[7] Ghufran Baig, Jian He, Mubashir Adnan Qureshi, Lili Qiu, Guohai Chen, Peng Chen, and Yinliang Hu. 2019. Jigsaw: Robust live 4k video streaming. In *Proceedings of ACM International Conference on Mobile Computing and Networking (MobiCom)*.

[8] Yanan Bao, Tianxiao Zhang, Amit Pande, Huasen Wu, and Xin Liu. 2017. Motion-Prediction-Based Multicast for 360-Degree Video Transmissions. In *Proceedings of IEEE International Conference on Sensing, Communication, and Networking (SECON)*. https://doi.org/10.1109/SAHCN.2017.7964928

[9] Jacob Chakareski. 2020. Viewport-Adaptive Scalable Multi-User Virtual Reality Mobile-Edge Streaming. *IEEE Transactions on Image Processing* 29 (2020), 6330–6342. https://doi.org/10.1109/TIP.2020.2986547

[10] Jiangong Chen, Xudong Qin, Guangyu Zhu, Bo Ji, and Bin Li. 2021. Motion-Prediction-based Wireless Scheduling for Multi-User Panoramic Video Streaming. In *Proceedings of IEEE International Conference on Computer Communications (INFOCOM)*.

[11] Lovish Chopra, Sarthak Chakraborty, Abhijit Mondal, and Sandip Chakraborty. 2021. PARIMA: Viewport Adaptive 360-Degree Video Streaming. In *Proceedings of the Web Conference 2021* (Ljubljana, Slovenia) *(WWW '21)*. Association for Computing Machinery, New York, NY, USA, 2379–2391. https://doi.org/10.1145/3442381.3450070

[12] Google Draco library 2020. . https://github.com/google/draco

[13] Francesco Gringoli, Pablo Serrano, Iñaki Ucar, Nicolò Facchi, and Arturo Azcorra. 2019. Experimental QoE Evaluation of Multicast Video Delivery over IEEE 802.11aa WLANs. *IEEE Transactions on Mobile Computing* 18, 11 (2019), 2549–2561. https://doi.org/10.1109/TMC.2018.2876000

[14] Serhan Gül, Sebastian Bosse, Dimitri Podborski, Thomas Schierl, and Cornelius Hellge. 2020. *Kalman Filter-Based Head Motion Prediction for Cloud-Based Mixed Reality*. Association for Computing Machinery, New York, NY, USA, 3632–3641. https://doi.org/10.1145/3394171.3413699

[15] Serhan Gül, Dimitri Podborski, Thomas Buchholz, Thomas Schierl, and Cornelius Hellge. 2020. Low-latency cloud-based volumetric video streaming using head motion prediction. In *Proceedings of ACM Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV)*.

[16] Serhan Gül, Dimitri Podborski, Jangwoo Son, Gurdeep Singh Bhullar, Thomas Buchholz, Thomas Schierl, and Cornelius Hellge. 2020. Cloud Rendering-based Volumetric Video Streaming System for Mixed Reality Services. In *Proceedings of ACM Multimedia Systems Conference (MMSys)*.

[17] Simon N. B. Gunkel, Hans M. Stokking, Martin J. Prins, Nanda van der Stap, Frank B. ter Haar, and Omar A. Niamut. 2018. Virtual Reality Conferencing: Multi-User Immersive VR Experiences on the Web. In *Proceedings of ACM Multimedia Systems Conference (MMSys)*. https://doi.org/10.1145/3204949.3208115

[18] Ankit Gupta, Jinfeng Du, Dmitry Chizhik, Reinaldo A. Valenzuela, and Mathini Sellathurai. 2022. Machine Learning-Based Urban Canyon Path Loss Prediction Using 28 GHz Manhattan Measurements. *IEEE Transactions on Antennas and Propagation* 70, 6 (2022), 4096–4111. https://doi.org/10.1109/TAP.2022.3152776

[19] Muhammad Kumail Haider and Edward W. Knightly. 2016. Mobility Resilience and Overhead Constrained Adaptation in Directional 60 GHz WLANs: Protocol Design and System Implementation. In *Proceedings of the 17th ACM International Symposium on Mobile Ad Hoc Networking and Computing* (Paderborn, Germany) *(MobiHoc '16)*. Association for Computing Machinery, New York, NY, USA, 61–70. https://doi.org/10.1145/2942358.2942380

[20] B. Han, Y. Liu, and F. Qian. 2020. ViVo: Visibility-Aware Mobile Volumetric Video Streaming. In *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking* (London, United Kingdom) *(MobiCom '20)*.

[21] Haitham Hassanieh, Omid Abari, Michael Rodriguez, Mohammed Abdelghany, Dina Katabi, and Piotr Indyk. 2018. Fast Millimeter Wave Beam Alignment. In *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication* (Budapest, Hungary) *(SIGCOMM '18)*. ACM, New York, NY, USA, 432–445. https://doi.org/10.1145/3230543.3230581

[22] Jian He, Mubashir Adnan Qureshi, Lili Qiu, Jin Li, Feng Li, and Lei Han. 2018. Rubiks: Practical 360° Streaming for Smartphones. In *Proceedings of ACM International Conference on Mobile Systems, Applications, and Services (MobiSys)*.

[23] IEEE 802.11ay: Enhanced Throughput for Operation in Bands above 45 GHz. [n. d.]. http://www.ieee802.org/11/Reports/tgay_update.htm.

[24] IEEE P802.11adTM/D4.0. 2012. Part 11: WLAN MAC/PHY Enhancements for Very High Throughput in the 60 GHz Band. (2012).

[25] Ish Kumar Jain, Raghav Subbaraman, and Dinesh Bharadia. 2021. Two Beams Are Better than One: Towards Reliable and High Throughput MmWave Links. In *Proceedings of the 2021 ACM SIGCOMM 2021 Conference* (Virtual Event, USA) *(SIGCOMM '21)*. Association for Computing Machinery, New York, NY, USA, 488–502. https://doi.org/10.1145/3452296.3472924

[26] Suraj Jog, Jiaming Wang, Junfeng Guan, Thomas Moon, Haitham Hassanieh, and Romit Roy Choudhury. 2019. Many-to-Many Beam Alignment in Millimeter Wave Networks. In *16th USENIX Symposium on Networked Systems Design and Implementation (NSDI 19)*. USENIX Association, Boston, MA, 783–800. https://www.usenix.org/conference/nsdi19/presentation/jog

[27] JPEG Pleno Database: 8i Voxelized Full Bodies - A Dynamic Voxelized Point Cloud Dataset 2019. . http://plenodb.jpeg.org/pc/8ilabs

[28] Mattia Lecci, Paolo Testolina, Michele Polese, Marco Giordani, and Michele Zorzi. 2021. Accuracy Versus Complexity for mmWave Ray-Tracing: A Full Stack Perspective. *IEEE Transactions on Wireless Communications* 20, 12 (2021), 7826–7841. https://doi.org/10.1109/TWC.2021.3088349

[29] K. Lee, J. Yi, Y. Lee, S. Choi, and Y. Kim. 2020. GROOT: A Real-Time Streaming System of High-Fidelity Volumetric Videos. In *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking* (London, United Kingdom) *(MobiCom '20)*. Article 57, 14 pages.

[30] Ron Levie, Çağkan Yapar, Gitta Kutyniok, and Giuseppe Caire. 2021. RadioUNet: Fast Radio Map Estimation With Convolutional Neural Networks. *IEEE Transactions on Wireless Communications* 20, 6 (2021), 4001–4015. https://doi.org/10.1109/TWC.2021.3054977

[31] Bin Li, Zheng Zhou, Weixia Zou, Xuebin Sun, and Guanglong Du. 2013. On the efficient beam-forming training for 60GHz wireless personal area networks. *IEEE Transactions on Wireless Communications* 12, 2 (2013), 504–515.

[32] Yong Li and Wei Gao. 2018. MUVR: Supporting Multi-User Mobile Virtual Reality with Resource Constrained Edge Cloud. In *Proceedings of IEEE/ACM Symposium on Edge Computing (SEC)*. https://doi.org/10.1109/SEC.2018.00008

[33] Luyang Liu, Ruiguang Zhong, Wuyang Zhang, Yunxin Liu, Jiansong Zhang, Lintao Zhang, and Marco Gruteser. 2018. Cutting the Cord: Designing a High-quality Untethered VR System with Low Latency Remote Rendering. In *Proceedings of ACM International Conference on Mobile Systems, Applications, and Services (MobiSys)*.

[34] Xing Liu, Bo Han, Feng Qian, and Matteo Varvello. 2019. LIME: Understanding Commercial 360° Live Video Streaming Services. In *Proceedings of the 10th ACM Multimedia Systems Conference* (Amherst, Massachusetts) *(MMSys '19)*. Association for Computing Machinery, New York, NY, USA, 154–164. https://doi.org/10.1145/3304109.3306220

[35] Yu Liu, Bo Han, Feng Qian, Arvind Narayanan, and Zhi-Li Zhang. 2022. Vues: Practical Mobile Volumetric Video Streaming Through Multiview Transcoding. In *Proceedings of ACM International Conference on Mobile Computing and Networking (MobiCom)*.

[36] Zida Liu, Guohao Lan, Jovan Stojkovic, Yunfan Zhang, Carlee Joe-Wong, and Maria Gorlatova. 2020. CollabAR: Edge-assisted Collaborative Image Recognition for Mobile Augmented Reality. In *Proceedings of ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*. https://doi.org/10.1109/IPSN48710.2020.00-26

[37] Zhinus Marzi, Upamanyu Madhow, and Haitao Zheng. 2015. Interference analysis for mm-wave picocells. In *Global Communications Conference (GLOBECOM), 2015 IEEE*. IEEE, 1–6.

[38] Jiayi Meng, Sibendu Paul, and Charlie Y. Hu. 2020. Coterie: Exploiting Frame Similarity to Enable High-Quality Multiplayer VR on Commodity Mobile Devices. In *Proceedings of International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*. https://doi.org/10.1145/3373376.3378516

[39] Raheeb Muzaffar, Evsen Yanmaz, Christian Bettstetter, and Andrea Cavallaro. 2016. Application-Layer Rate-Adaptive Multicast Video Streaming over 802.11 for Mobile Devices. In *Proceedings of the 24th ACM International Conference on Multimedia* (Amsterdam, The Netherlands) *(MM '16)*. Association for Computing Machinery, New York, NY, USA, 506–510. https://doi.org/10.1145/2964284.2967272

[40] A. Narayanan, E. Ramadan, J. Carpenter, Q. Liu, Y. Liu, F. Qian, and Z. Zhang. 2020. A First Look at Commercial 5G Performance on Smartphones. In *Proceedings of The Web Conference 2020* (Taipei, Taiwan) *(WWW '20)*. 894–905.

[41] Sharan Naribole and Edward Knightly. 2016. Scalable Multicast in Highly-Directional 60 GHz WLANs. In *2016 13th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*. 1–9. https://doi.org/10.1109/SAHCN.2016.7733014

[42] Next generation display media in 6G 2021. . https://research.samsung.com/next-generation-display-media

[43] Qualcomm Atheros QCA6310-based 802.11ad WiGig RF Antenna Module NGFF595A-L-Ant. [n. d.]. https://fairwayelectronic.com/shop/qualcomm-atheros/ngff595a-l-ant-qualcomm-atheros-qca6310-based-802-11ad-wigig-rf-antenna-module/.

[44] Thomas Nitsche, Guillermo Bielsa, Irene Tejado, Adrian Loch, and Joerg Widmer. 2015. Boon and bane of 60 GHz networks: practical insights into beamforming, interference, and frame level operation. In *Proceedings of the 11th ACM Conference on Emerging Networking Experiments and Technologies*. ACM, 17.

[45] Thomas Nitsche, Adriana B Flores, Edward W Knightly, and Joerg Widmer. 2015. Steering with eyes closed: mm-wave beam steering without in-band measurement.

In *Computer Communications (INFOCOM), 2015 IEEE Conference on*. IEEE, 2416–2424.

[46] T. Nitsche, A. B. Flores, E. W. Knightly, and J. Widmer. 2015. Steering with eyes closed: Mm-Wave beam steering without in-band measurement. In *2015 IEEE Conference on Computer Communications (INFOCOM)*. 2416–2424. https://doi.org/10.1109/INFOCOM.2015.7218630

[47] Joan Palacios, Daniel Steinmetzer, Adrian Loch, Matthias Hollick, and Joerg Widmer. 2018. Adaptive Codebook Optimization for Beam Training on Off-the-Shelf IEEE 802.11Ad Devices. In *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking* (New Delhi, India) *(MobiCom '18)*. ACM, New York, NY, USA, 241–255. https://doi.org/10.1145/3241539.3241576

[48] Jounsup Park, Philip A. Chou, and Jenq-Neng Hwang. 2019. Rate-Utility Optimized Streaming of Volumetric Media for Augmented Reality. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems* 9, 1 (2019), 149–162.

[49] Cristina Perfecto, Mohammed S. Elbamby, Javier Del Ser, and Mehdi Bennis. 2020. Taming the Latency in Multi-User VR 360°: A QoE-Aware Deep Learning-Aided Multicast Framework. *IEEE Transactions on Communications* 68, 4 (2020), 2491–2508. https://doi.org/10.1109/TCOMM.2020.2965527

[50] Dimitri Podborski, Serhan Gül, Jangwoo Son, Gurdeep Singh Bhullar, Robert Skupin, Yago Sanchez, Thomas Schierl, and Cornelius Hellge. 2020. Interactive Low Latency Video Streaming of Volumetric Content. In *Proceedings of International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*.

[51] Qualcomm QCA9500. 2020. https://www.qualcomm.com/products/technology/wi-fi/qca9500.

[52] Feng Qian, Bo Han, Jarrell Pair, and Vijay Gopalakrishnan. 2019. Toward Practical Volumetric Video Streaming On Commodity Smartphones. In *Proceedings of International Conference on Mobile Computing and Networking (MobiCom)*.

[53] Feng Qian, Bo Han, Qingyang Xiao, and Vijay Gopalakrishnan. 2018. Flare: Practical Viewport-Adaptive 360-Degree Video Streaming for Mobile Devices. In *Proceedings of ACM International Conference on Mobile Computing and Networking (MobiCom)*.

[54] Jian Qiao, Yejun He, and Xuemin Sherman Shen. 2016. Proactive Caching for Mobile Video Streaming in Millimeter Wave 5G Networks. *IEEE Transactions on Wireless Communications* 15, 10 (2016), 7187–7198. https://doi.org/10.1109/TWC.2016.2598748

[55] Xukan Ran, Carter Slocum, Yi-Zhen Tsai, Kittipat Apicharttrisorn, Maria Gorlatova, and Jiasi Chen. 2020. Multi-User Augmented Reality with Communication Efficient and Spatially Consistent Virtual Objects. In *Proceedings of ACM International Conference on emerging Networking EXperiments and Technologies (CoNEXT)*. https://doi.org/10.1145/3386367.3431312

[56] Theodore S Rappaport, Felix Gutierrez, Eshar Ben-Dor, James N Murdock, Yijun Qiao, and Jonathan I Tamir. 2013. Broadband millimeter-wave propagation measurements and models using adaptive-beam antennas for outdoor urban cellular communications. *IEEE transactions on antennas and propagation* 61, 4 (2013), 1850–1859.

[57] Theodore S Rappaport, Shu Sun, Rimma Mayzus, Hang Zhao, Yaniv Azar, Kevin Wang, George N Wong, Jocelyn K Schulz, Mathew Samimi, and Felix Gutierrez. 2013. Millimeter wave mobile communications for 5G cellular: It will work! *IEEE access* 1 (2013), 335–349.

[58] Remcom Wireless InSite 3D Wireless Prediction Software 2021. . https://www.remcom.com/wireless-insite-em-propagation-software

[59] Swetank Saha, Shivang Aggarwal, Hany Assasa, Adrian Loch, Naveen Muralidhar Prakash, Roshan Shyamsunder, Daniel Steinmetzer, Dimitrios Koutsonikolas, Joerg Widmer, and Matthias Hollick. 2020. Performance and Pitfalls of 60 GHz WLANs Based on Consumer-Grade Hardware. *IEEE Transactions on Mobile Computing* (2020).

[60] Swetank Kumar Saha, Shivang Aggarwal, Rohan Pathak, Dimitrios Koutsonikolas, and Joerg Widmer. 2019. MuSher: An Agile Multipath-TCP Scheduler for Dual-Band 802.11 ad/ac Wireless LANs. In *Proceedings of the 25th Annual International Conference on Mobile Computing and Networking*. 1–16.

[61] Harkirat Singh, Jisung Oh, Changyeul Kweon, Xiangping Qin, Huai-Rong Shao, and Chiu Ngo. 2008. A 60 GHz wireless network for enabling uncompressed video communication. *IEEE Communications Magazine* 46, 12 (2008), 71–78. https://doi.org/10.1109/MCOM.2008.4689210

[62] Sumit Singh, Raghuraman Mudumbai, and Upamanyu Madhow. 2011. Interference analysis for highly directional 60-GHz mesh networks: The case for rethinking medium access control. *IEEE/ACM Transactions on Networking (TON)* 19, 5 (2011), 1513–1527.

[63] Peter FM Smulders. 2009. Statistical characterization of 60-GHz indoor radio channels. *IEEE Transactions on Antennas and Propagation* 57, 10 (2009), 2820–2829.

[64] Daniel Steinmetzer, Daniel Wegemer, Matthias Schulz, Joerg Widmer, and Matthias Hollick. 2017. Compressive Millimeter-Wave Sector Selection in Off-the-Shelf IEEE 802.11Ad Devices. In *Proceedings of the 13th International Conference on Emerging Networking EXperiments and Technologies* (Incheon, Republic of Korea) *(CoNEXT '17)*. ACM, New York, NY, USA, 414–425. https://doi.org/10.1145/3143361.3143384

[65] Liyang Sun, Fanyi Duanmu, Yong Liu, Yao Wang, Yinghua Ye, Hang Shi, and David Dai. 2018. Multi-path multi-tier 360-degree video streaming in 5G networks. In *Proceedings of ACM Multimedia Systems Conference (MMSys)*. https://doi.org/10.1145/3204949.3204978

[66] Sanjib Sur, Ioannis Pefkianakis, Xinyu Zhang, and Kyu-Han Kim. 2017. WiFi-Assisted 60 GHz Wireless Networks. In *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking* (Snowbird, Utah, USA) *(MobiCom '17)*. Association for Computing Machinery, New York, NY, USA, 28–41. https://doi.org/10.1145/3117811.3117817

[67] Sanjib Sur, Ioannis Pefkianakis, Xinyu Zhang, and Kyu-Han Kim. 2018. Towards Scalable and Ubiquitous Millimeter-Wave Wireless Networks. In *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking* (New Delhi, India) *(MobiCom '18)*. ACM, New York, NY, USA, 257–271. https://doi.org/10.1145/3241539.3241579

[68] Sanjib Sur, Vignesh Venkateswaran, Xinyu Zhang, and Parmesh Ramanathan. 2015. 60 GHz Indoor Networking through Flexible Beams: A Link-Level Profiling. In *Proceedings of the 2015 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems* (Portland, Oregon, USA) *(SIGMETRICS '15)*. Association for Computing Machinery, New York, NY, USA, 71–84. https://doi.org/10.1145/2745844.2745858

[69] S. Sur, X. Zhang, P. Ramanathan, and R. Chandra. 2016. BeamSpy: Enabling Robust 60 GHz Links under Blockage. In *Proceedings of the 13th Usenix Conference on Networked Systems Design and Implementation* (Santa Clara, CA) *(NSDI'16)*. 193–206.

[70] MATLAB Antenna Toolbox. [n. d.]. https://www.mathworks.com/products/antenna.html.

[71] Vutha Va and Robert W. Heath. 2015. Basic Relationship between Channel Coherence Time and Beamwidth in Vehicular Channels. In *2015 IEEE 82nd Vehicular Technology Conference (VTC2015-Fall)*. 1–5. https://doi.org/10.1109/VTCFall.2015.7390852

[72] View Frustum Culling 2015. . http://www.lighthouse3d.com/tutorials/view-frustumculling/

[73] Vubiq, Irvine, CA, USA. [n. d.]. *60 GHz Systems and Modules*. https://www.pasternack.com/60-ghz-systems-and-modules-category.aspx

[74] Song Wang, Jingqi Huang, Xinyu Zhang, Hyoil Kim, and Sujit Dey. 2020. X-Array: Approximating Omnidirectional Millimeter-Wave Coverage Using an Array of Phased Arrays. In *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking* (London, United Kingdom) *(MobiCom '20)*. Association for Computing Machinery, New York, NY, USA, Article 5, 14 pages. https://doi.org/10.1145/3372224.3380882

[75] Song Wang, Jingqi Huang, Xinyu Zhang, Hyoil Kim, and Sujit Dey. 2020. X-Array: Approximating Omnidirectional Millimeter-Wave Coverage Using an Array of Phased Arrays. In *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking* (London, United Kingdom) *(MobiCom '20)*. Association for Computing Machinery, New York, NY, USA, Article 5, 14 pages. https://doi.org/10.1145/3372224.3380882

[76] Teng Wei and Xinyu Zhang. 2017. Pose Information Assisted 60 GHz Networks: Towards Seamless Coverage and Mobility Support. In *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking* (Snowbird, Utah, USA) *(MobiCom '17)*. 42–55.

[77] Teng Wei, Anfu Zhou, and Xinyu Zhang. 2017. Facilitating Robust 60 GHz Network Deployment by Sensing Ambient Reflectors. In *Proceedings of the 14th USENIX Conference on Networked Systems Design and Implementation* (Boston, MA, USA) *(NSDI'17)*. 213–226.

[78] wil6210 module. 2021. https://git.kernel.org/pub/scm/linux/kernel/git/kvalo/ath.git/tree/drivers/net/wireless/ath/wil6210.

[79] Hao Xu, Vikas Kukshya, and Theodore S Rappaport. 2002. Spatial and temporal characteristics of 60-GHz indoor channels. *IEEE Journal on selected areas in communications* 20, 3 (2002), 620–630.

[80] T. Xu, B. Han, and F. Qian. 2019. Analyzing Viewport Prediction under Different VR Interactions. In *Proceedings of the 15th International Conference on Emerging Networking Experiments And Technologies* (Orlando, Florida) *(CoNEXT '19)*. 165–171.

[81] Z. Yang, P. H. Pathak, J. Pan, M. Sha, and P. Mohapatra. 2018. Sense and Deploy: Blockage-Aware Deployment of Reliable 60 GHz mmWave WLANs. In *2018 IEEE 15th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*. 397–405. https://doi.org/10.1109/MASS.2018.00063

[82] Zhicheng Yang, Parth H. Pathak, Yunze Zeng, and Prasant Mohapatra. 2015. Sensor-Assisted Codebook-Based Beamforming for Mobility Management in 60 GHz WLANs. In *2015 IEEE 12th International Conference on Mobile Ad Hoc and Sensor Systems*. 333–341. https://doi.org/10.1109/MASS.2015.58

[83] Anlan Zhang, Chendong Wang, Bo Han, and Feng Qian. 2021. Efficient Volumetric Video Streaming Through Super Resolution. In *Proceedings of ACM HotMobile*.

[84] Anlan Zhang, Chendong Wang, Bo Han, and Feng Qian. 2022. YuZu: Neural-Enhanced Volumetric Video Streaming. In *Proceedings of USENIX Symposium on Networked Systems Design and Implementation (NSDI)*.

[85] Ding Zhang, Mihir Garude, and Parth H. Pathak. 2018. MmChoir: Exploiting Joint Transmissions for Reliable 60GHz MmWave WLANs. In *Proceedings of the Eighteenth ACM International Symposium on Mobile Ad Hoc Networking and Computing* (Los Angeles, CA, USA) *(Mobihoc '18)*. Association for Computing Machinery, New York, NY, USA, 251–260. https://doi.org/10.1145/3209582.3209608

[86] Ding Zhang, Bo Han, Parth Pathak, and Haoliang Wang. 2021. Innovating Multi-User Volumetric Video Streaming through Cross-Layer Design. In *Proceedings of the Twentieth ACM Workshop on Hot Topics in Networks* (Virtual Event, United Kingdom) *(HotNets '21)*. Association for Computing Machinery, New York, NY, USA, 16–22. https://doi.org/10.1145/3484266.3487396

[87] Ding Zhang, Panneer Selvam Santhalingam, Parth Pathak, and Zizhan Zheng. 2019. Characterizing Interference Mitigation Techniques in Dense 60 GHz mmWave WLANs. In *International Conference on Computer Communications and Networks* (Valencia, Spain) *(ICCCN '19)*.

[88] Wenxiao Zhang, Bo Han, and Pan Hui. 2022. SEAR: Scaling Experiences in Multi-user Augmented Reality. In *Proceedings of IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*.

[89] Wenxiao Zhang, Bo Han, Pan Hui, Vijay Gopalakrishnan, Eric Zavesky, and Feng Qian. 2018. CARS: Collaborative Augmented Reality for Socialization. In *Proceedings of ACM International Workshop on Mobile Computing Systems and Applications (HotMobile)*.

[90] Wenxiao Zhang, Feng Qian, Bo Han, and Pan Hui. 2021. DeepVista: 16K Panoramic Cinema on Your Mobile Device. In *Proceedings of the Web Conference 2021* (Ljubljana, Slovenia) *(WWW '21)*. Association for Computing Machinery, New York, NY, USA, 2232–2244. https://doi.org/10.1145/3442381.3449829

[91] Xin Zhang, Xiujun Shu, Bingwen Zhang, Jie Ren, Lizhou Zhou, and Xin Chen. 2020. Cellular Network Radio Propagation Modeling with Deep Convolutional Neural Networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Virtual Event, CA, USA) *(KDD '20)*. Association for Computing Machinery, New York, NY, USA, 2378–2386. https://doi.org/10.1145/3394486.3403287

[92] Renjie Zhao, Timothy Woodford, Teng Wei, Kun Qian, and Xinyu Zhang. 2020. M-cube: A millimeter-wave massive MIMO software radio. In *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*. 1–14.

[93] Anfu Zhou, Leilei Wu, Shaoqing Xu, Huadong Ma, Teng Wei, and Xinyu Zhang. 2018. Following the shadow: Agile 3-D beam-steering for 60 GHz wireless networks. In *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*. IEEE, 2375–2383.

[94] Xia Zhou, Zengbin Zhang, Yibo Zhu, Yubo Li, Saipriya Kumar, Amin Vahdat, Ben Y Zhao, and Haitao Zheng. 2012. Mirror mirror on the ceiling: Flexible wireless links for data centers. *ACM SIGCOMM Computer Communication Review* 42, 4 (2012), 443–454.

[95] Yibo Zhu, Zengbin Zhang, Zhinus Marzi, Chris Nelson, Upamanyu Madhow, Ben Y Zhao, and Haitao Zheng. 2014. Demystifying 60GHz outdoor picocells. In *Proceedings of the 20th annual international conference on Mobile computing and networking*. ACM, 5–16.

[96] Yibo Zhu, Xia Zhou, Zengbin Zhang, Lin Zhou, Amin Vahdat, Ben Y Zhao, and Haitao Zheng. 2014. Cutting the cord: a robust wireless facilities network for data centers. In *Proceedings of the 20th annual international conference on Mobile computing and networking*. ACM, 581–592.