An End-to-End Programmable Testbed for the Experimental Evaluation of Video Streaming at mmWaves

Neagin Neasamoni Santhi, Michele Polese, Tommaso Melodia
Institute for the Wireless Internet of Things, Northeastern University, Boston, MA, USA

2022 IEEE Globecom Workshops (GC Wkshps) | 978-1-6654-5975-4/22/$31.00 ©2022 IEEE | DOI: 10.1109/GCWkshps56602.2022.10008612

***Abstract*—The harsh propagation environment in the millimeter wave (mmWave) band impacts all the layers of the protocol stack. This calls for full-stack, end-to-end performance evaluation platforms, with programmable lower layers, to enable cross-layer approaches, and with the support for application data traffic and transport protocols. So far, most full-stack mmWave studies have relied on commercial mmWave devices, which have limited insights and programmability at the link level, or on simulations. This paper introduces a fully programmable, software-defined platform for the design, prototyping, and evaluation of the end-to-end application performance at 60 GHz. It extends the NI mmWave Transceiver System (MTS) with real-time video streaming capabilities and a reliable retransmission-based Medium Access Control (MAC) layer. This platform establishes a framework that can be used for the development and evaluation of cross-layer optimization at mmWaves. We evaluate the performance of a video streaming use case with different video bitrates, Modulation and Coding Schemes (MCSs), and link configurations, to showcase the end-to-end, full-stack capabilities of the platform, and discuss the challenges for the support of real-time application traffic over a link with 2 GHz of bandwidth.**

***Index Terms*—mmWave, 60 GHz, testbed, video streaming**

I. Introduction

The mobile experience associated with data-rate-hungry multimedia applications and Virtual Reality (VR) is one of the main drivers behind the adoption of higher frequency bands [1]. The 5th generation (5G) and 6th generation (6G) of cellular networks will indeed include communications in the millimeter wave (mmWave) spectrum. Nonetheless, mobile access at mmWave comes with several challenges related to the harsh propagation environment at these frequency bands [2]. This comprise (i) a higher isotropic propagation loss at sub-6 GHz, partially compensated by directional transmissions; (ii) additional molecular absorption in specific bands (e.g., at 60 GHz, with a distance-dependent loss of up to 15 dB/km); and (iii) blockage from obstacles such as vehicles, the human body, brick and mortar, among others.

These challenges have been widely studied from a Physical (PHY) and Medium Access Control (MAC) layer perspective [3]. The design of antenna arrays and beamforming schemes (either analog, hybrid, or digital) has led to integrated packaging and multi-beam solutions that are now part of commercial user devices and base stations [4]. At the MAC layer, the technical specifications for 3GPP NR, one of the Radio Access Network (RAN) designs for 5G networks, include beam management procedures [5], and several strategies to cope with alignment between the endpoints of a communication link.

MmWave PHY and MAC research includes studies with theoretical, simulation, and experimental results, enabled, for example, by testbeds based on Field Programmable Gate Arrays (FPGAs) like X60 [6], mMobile [7], MATE [8], and others [9], [10]. These feature programmable PHY and MAC layers, and have been extensively used for beamforming and Multiple Input, Multiple Output (MIMO) research [11], [12]. However, they are not designed to handle real-time data streams from the higher layers of the protocol stack, mostly because of limitations in the interface with the link-layer FPGAs. Other mmWave testbeds and studies use Commercial Off-the-shelf (COTS) devices, offering only limited control and insights into the PHY and MAC layers [13].

The challenges that mmWaves introduce at the link level, however, have practical impacts throughout the full protocol stack. For example, the typical adaptation strategies for transport layer or video streaming protocols suffer the high variability of the data rate available on mmWave links, leading to bufferbloat or inefficient resource utilization [14]. This has been shown through simulation studies, which also highlight the importance of cross-layer adaptation strategies to improve the reactiveness of transport and application layer adaptation loops to the rapidly changing mmWave channels [15].

This paper takes the first steps toward enabling full-stack, end-to-end experimental studies for the design and evaluation of cross-layer optimization techniques. It proposes a software-defined mmWave testbed that can be used for real-time, application layer data streaming at a carrier frequency of 60 GHz and over a bandwidth of 2 GHz. This lays the foundation for cross-layer optimization thanks to the combination of fully programmable, FPGA-based Digital Signal Processing (DSP) at the link level, and real transport layer implementations and applications, e.g., video streaming software. The key contributions are as follows:

- We extend the capabilities of a mmWave software-defined radio platform (i.e., the NI mmWave Transceiver System (MTS) with the SiBeam 60 GHz phased arrays) with

This work is partially funded by U.S. Office of Naval Research under Grant N00014-20-1- 2132 and by the U.S. National Science Foundation under Grant CNS-1923789.

(i) new MAC layer capabilities, i.e., an efficient real-time selective-repeat Automatic Repeat reQuest (ARQ) to enhance the reliability of the system; and (ii) new adaptation and configuration entry points.

- We introduce the possibility of transmitting application data on the mmWave over-the-air link. Specifically, we integrate streaming of real-time video data at data rates exceeding 100 Mbps. This represents one of the first experimental demonstrations of reliable, real-time video streaming on a fully programmable mmWave platform.
- We then profile the performance of the system with different Modulation and Coding Schemes (MCSs), video resolutions and data rates, retransmissions configurations, and blockage, showing the potential that this experimental platform offers for the evaluation of cross-layer solutions. We also discuss the challenges associated with developing such an experimental platform.

The rest of the paper is organized as follows. Sec. II reviews the state of the art in video streaming over mmWave links. Sec. III describes the platform and the extensions we implemented for retransmissions and video streaming. Sec. IV discusses the experimental evaluation. Sec. V concludes the paper and discusses future work.

II. State of the Art

As discussed above, there is limited research on experimental evaluations of video streaming solutions over mmWave links. Notably, papers [16], [17] evaluate different streaming policies using COTS devices, which usually prevent cross-layer optimization solutions. The authors of [16] analyze the impact of blockage on the PHY layer throughput, considering uncompressed video transmission for 360° immersive streaming. In this paper, we analyze the performance of different encoded video streams, taken from a widely used video benchmark dataset, and analyze the impact of the mmWave channel and blockage throughout the whole protocol stack, including the PHY layer, MAC with retransmissions, and the video quality. [17] considers the transmission of 4K uncompressed videos from surveillance drones with commercial 60 GHz dongles. This study focused on surveillance applications, including face recognition, and did not explore the system from a full-stack perspective. The authors of [18] developed MoVR, a mmWave-based system to support untethered VR, but only provide results in terms of Signal to Interference plus Noise Ratio (SINR) and evaluate the throughput based on a spectral efficiency table from the IEEE 802.11ad standard. [10] presents a testbed with programmable baseband units and software-defined radios at 28 GHz and 60 GHz, but reports results only for link level metrics.

Simulation-based studies on video streaming over mmWaves can be found in [19]–[22]. The authors of [19] propose a reliable video streaming architecture implemented with a heuristics-based dual-connectivity setup and network coding. The performance is evaluated through simulations based on actual video traces. Paper [20] focuses on QoE evaluation for 360° video streaming on mmWaves using simulations. [21] investigates buffering-based optimization of video streaming for mmWave-connected vehicles. [22] studies dual connectivity for panoramic video simulations.

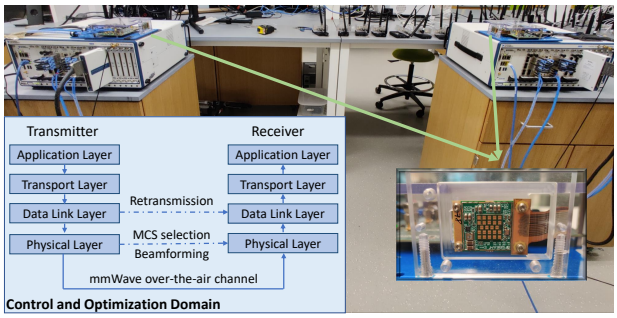


Figure 1: 60 GHz software-defined testbed based on the NI MTS and SiBeam antennas.

III. Software-defined, End-to-End mmWave Video Streaming Platform

This section describes the experimental software-defined platform, the integration of the MAC layer retransmission capabilities, and the video streaming components.

A. Platform Functionalities

The 60 GHz software-defined testbed consists of a set of NI MTSs and SiBeam 60 GHz phased array antennas [23]. Fig. 1 shows a pair of MTSs and the phased arrays, which act as Transmitter (TX) and Receiver (RX), deployed in an indoor lab environment with realistic scattering from benches and metal structures. The green arrows point toward the zoomed-in SiBeam antenna array on top of the MTSs.

The MTS is a modular, fully configurable software-defined radio with a PXIe chassis, a host device, the FlexRIO FPGA modules that implement DSP for the PHY layer, and high-speed Digital to Analog Converters (DACs) and Analog to Digital Converters (ADCs), capable of sampling the signal at 3.072 Gsamples/s. Overall, the system supports a maximum bandwidth of 2 GHz, and the SiBeam arrays have a carrier frequency at 60 GHz. The latter is a mmWave radio head (i.e., it also does the upconversion from baseband) with an electronically steerable array with 12 transmit and 12 receive antenna elements. The default codebook features 25 beam directions spanning from -60 degrees to 60 degrees with 5-degree increments in the azimuth plane.

The framework of the software-defined platform has been implemented by NI in LabVIEW, and provides a single-carrier PHY layer that aligns with the Verizon 5G TF specifications, which represent a 5G pre-standardization effort. In this prototype, transmissions occur in 10 ms frames subdivided into 100 slots of 100 μs each [24]. A slot comprises 92 Codewords (CWs), each with an associated Cyclic Redundancy Check (CRC) block. As the system supports 9 MCSs, which can be manually changed at runtime, the number of bits from the higher layers that a CW can accommodate depends on the MCS for the slot. The baseline implementation fills the CWs with randomly generated data by the PHY layer FPGAs

themselves for every frame. The slots can be uplink or downlink in a Time Division Duplexing (TDD) fashion.

By default, the platform has limited reconfigurability and did not allow the streaming of application data at high data rates. Moreover, it features unreliable, best effort PHY and MAC layers. Therefore, we significantly extended the LabVIEW implementation to expose additional parameters that can be programmatically tuned (e.g., the MCSs, beam selection) as well as metrics that can be logged for data collection. This makes it possible to perform cross-layer adaptation studies in the domain shown in Fig. 1. In addition, the following paragraphs provide details on how we implement retransmissions (Sec. III-B) to make the link reliable, and integrate video streaming into the platform (Sec. III-C).

B. Implementation of a Real-time Selective-repeat Retransmission Strategy

Retransmissions are a key component of ARQ protocols that handle the presence of packet errors by repeating the transmission of corrupted packets. We incorporated a real-time MAC layer ARQ process in the MTSs to guarantee reliable transmission and compensate for the high packet loss that affects the 60 GHz PHY layer. Our measurements indicate a CW error rate of 3% with MCS Quadrature Phase Shift Keying (QPSK) and 1/4 coding rate, which translates for example into an error probability of 42% for a 1316-byte higher-layer packet spanning 18 CWs.

Considering the high CW error rate, we opted for a selective-repeat retransmission mechanism. Indeed, selective-repeat only retransmits the CWs that were actually received with errors, and exploits pipelining, i.e., the TX is permitted to send new packets without waiting for the acknowledgments of previously transmitted frames. Therefore, it presents a higher efficiency when compared to simple stop-and-wait, which introduces unnecessary latencies and lower throughput, and go-back-N, which requires the retransmission of the entire group of CWs from the first wrong CW. This leads to a high number of unnecessary retransmissions in the presence of a high CW error rate. In addition to the buffer of unacknowledged packets at the TX, selective-repeat also requires a buffer at the RX for reliable and in-order packet delivery to the upper layers, hence making it more challenging to implement on a real-time platform if compared to the other ARQ policies.

For retransmissions of individual CWs, the receiver has to explicitly send acknowledgments or feedback to the transmitter for it to learn the received packet's error status and estimate the erroneous CWs. In this setup, the TX procures the CRC feedback and the associated frame number from the RX via uplink signaling at the end of each frame, i.e., once every 10 ms. However, the feedback requires (i) the decoding of the CWs and the processing of the CRC at the RX; and (ii) the decoding of the uplink CRCs at the TX, thus introducing a round-trip delay of 4 frames (i.e., 40 ms).

Prior to every transmission, the TX verifies the frame number attached to the CRC feedback received in the previous frame and checks if it corresponds to the top frame in the unacknowledged buffer. If this is the case, the feedback is considered free of errors, i.e., it can be used to select the CWs to retransmit. Alternatively, the feedback is ignored, and subsequently, the TX resends the entire frame.

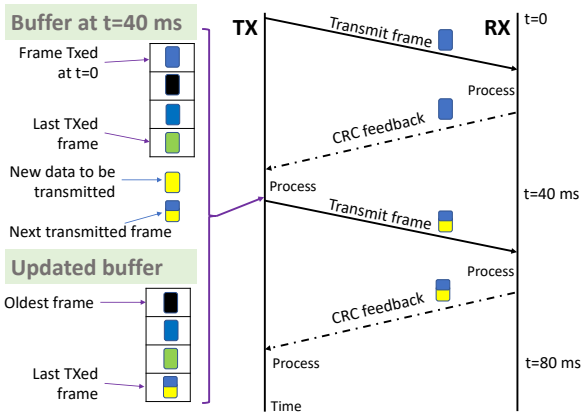


Figure 2: Example of TX buffer management during the selective-repeat retransmission process.

We modified each CW to include a header with the CW and frame numbers, which are allocated sequentially, and the frame length. Depending on the CRC feedback, new higher-layer data (e.g., video stream packets) can also be embedded in a CW. Every new transmitted frame includes the retransmitted CWs belonging to an earlier transmission at the beginning of the frame, as shown in the blue block in Fig. 2, and new CWs corresponding to that frame in the trailing portion, as shown in the yellow block in Fig. 2. The frame length corresponds to the number of new CWs.

The TX maintains a buffer to retain a copy of each unacknowledged frame, as illustrated in Fig. 2, whose size (set to 4 frames) depends on the time taken for the TX to receive the CRC feedback from the RX, i.e., 40 ms. For every transmission, we dequeue the oldest element of the buffer to create a new frame, shift the remaining ones, and enqueue the newly transmitted frame, as shown in Fig. 2.

Similarly, the RX buffer is also configured as a circular buffer, and its size can be adapted based on the number of retransmissions required and the MCS scheme. This improves the efficiency of the memory utilization at the receiver. The buffer contains information on the frame number, the frame length, and the CW data and numbers. The buffer gets updated with correctly received CWs based on their frame number until any missing CW is received. Next, the entire sequence of correctly received CWs in that frame can be delivered sequentially to the upper layers of the stack, therefore guaranteeing in-order delivery. We designed the system to release a maximum of two frames at any given time, to limit rate of the information that needs to be processed by the higher layers of the stack.

C. Integration of End-to-End Video Streaming Capabilities

As discussed in Sec. III-A, by default the MTS cannot stream real application data with high data rates, and generates random data directly in the PHY layer FPGA. We implemented

MCS	1/5 BPSK	1/4 QPSK	1/2 16-QAM	7/8 16-QAM
Bytes/slot	2632	6580	30268	39480
Max data rate	103 Mbps	257 Mbps	1186 Mbps	1547 Mbps

Table I: Application-layer data that can be embedded in a PHY layer frame of 10 ms, assuming no CWs are used for retransmitted data. For the MCSs, we indicate the coding ratio and the modulation.

a high-speed interface between the host PC in the PXIe chassis and the PHY layer FPGA, enabling real-time streaming at tens of Mbps. The first iteration only supports one-way application traffic (i.e., from TX to RX), which will be extended in future implementations. For these reasons, we consider only UDP as the supported transport protocol. In this sense, introducing a reliable MAC layer through ARQ, as described in Sec. III-B, is key to guaranteeing the application layer data's correct and in-order reception.

For this study, we focus on a video streaming use case, and use the popular video player VLC [25] to generate application layer data over UDP. Notably, video frames are split into packets with a fixed size of 1316 bytes, independent of the video source encoding. The MTS host PC exposes a socket that receives the UDP packets and puts them in a queue for further processing from the host and then the DSP FPGAs. This host-to-FPGA interface is implemented through asynchronous queues to enable data transfer across two different clock domains, i.e., the host, which relies on CPU-based processing with indeterministic timing, and the FPGAs, which implement timed-loops with stringent deadlines to complete the operation, and process the real-time data for the transmission in frames of 10 ms. Before embedding the data into CWs, the TX performs the XOR-ing of the content of the UDP packets with a scrambling sequence. The actual number of bytes from UDP packets embedded in a frame depends on the MCS and the number of CWs from previous frames that need to be retransmitted. Notably, the maximum number of bytes that a slot can carry and the maximum theoretical data rate for each MCS are reported in Table I.

The RX extracts data from the frames released by ARQ buffer and transfers them to a designated socket. Multiple parallel queues have been implemented to handle the transfer of data from the PHY layer FPGAs to the host without overflowing the RX buffers in different time domains and to guarantee that the application layer can consume the received data in a timely fashion.

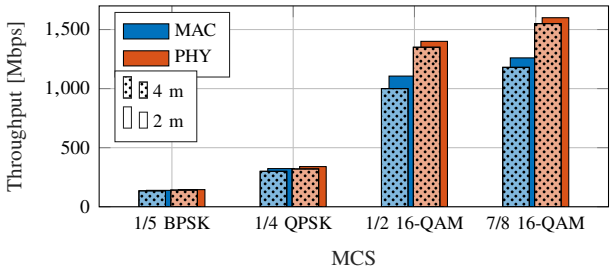


Figure 4: Throughput for different MCSs and distances.

IV. Experimental Evaluation

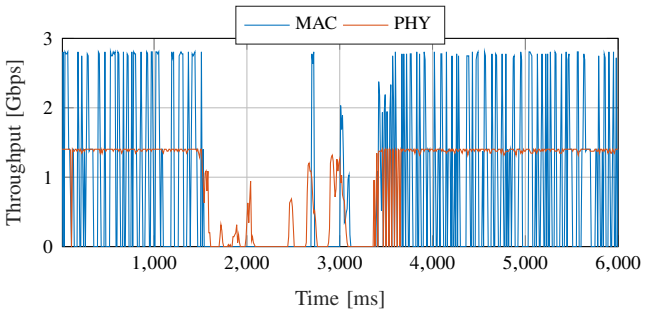
This section presents the results of the mmWave video streaming use case, with the experimental setup (Sec. IV-A) and the performance evaluation in terms of PHY and MAC layer throughput, the impact of blockage, and other video streaming performance metrics (Sec. IV-B).

A. Experiment Setup and Parameters

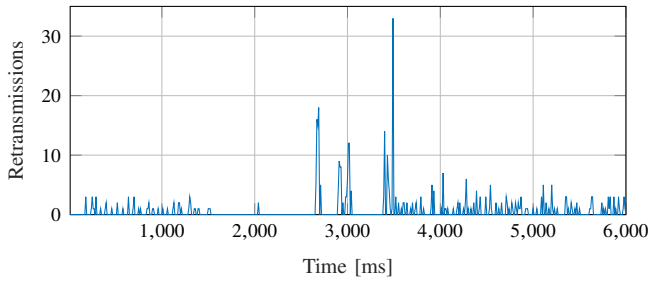
We run multiple experiments to measure the system performance with different videos, distances, MCSs, and with or without retransmissions. The MTS TX and RX are configured with a proper RX gain, antenna alignment, and carrier frequency offset for synchronization. We consider distances of 2 m and 4 m, and the MCSs in Table I.

We consider test videos typically used for video streaming evaluation from the Jellyfish MKV dataset [26], using VLC, with bitrates ranging from 10 Mbps to 140 Mbps. Videos have a different level of compression (based on H.264 encoding), and either 4K resolution with 3840×2160 pixels per frame (for bitrate higher than 100 Mbps) or Full HD resolution with 1920×1080 pixels. We use `ffmpeg` [27] to evaluate the Structural Similarity Index (SSIM) score of each received video as the average SSIM for all the frames. SSIM measures the perceived change in the structural information of the video using its luminance, contrast, and structure, and its value ranges from 0 to 1. Generally, frames with SSIM above 0.9 have visually unrecognizable distortion [28], [29].

We define MAC throughput as the rate of successfully received bits by the MAC layer (released from the RX buffer



(a) PHY and MAC throughput.



(b) Retransmissions.

Figure 5: Impact of blockage on different link metrics. The link uses 1/4 QPSK and the TX and RX are at a distance of 2 m.

to the upper layer), thus reflecting on throughput loss due to retransmissions, whereas PHY throughput is the rate of correctly received bits by the PHY layer. The average of MAC throughput is always lower than PHY throughput's mean.

B. Experimental Results

PHY and MAC layer results. Fig. 4 illustrates the PHY and MAC layer throughput for 2 m and 4 m with different MCSs. The redundancy introduced by the MAC layer (i.e., retransmissions, headers) leads to a lower MAC layer throughput, with a difference that increases with higher MCSs (5% overhead for 1/5 BPSK at 2 m, 22% for 7/8 16-QAM at 2 m), which are less robust to signal distortions and have higher CW error rate, but not with the distance. As expected, the throughput is higher at 2 m than 4 m, with a loss of ~ 50 Mbps (3.12%) for 7/8 16-QAM and ~ 4 Mbps (2.7%) for 1/5 BPSK at the PHY layer.

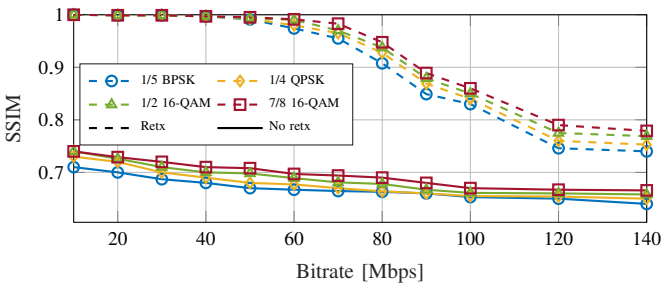
Blockage results. We investigate the effects of blockage on the PHY and MAC layer throughput and the number of retransmissions in Fig. 5. A human hand is the blockage source, moving across the line of sight between the TX and the RX. This highlights one of the main benefits of using an end-to-end, full-stack experimental approach to study mmWave systems, i.e., it is possible to accurately characterize channel phenomena in realistic settings while evaluating higher-layer metrics. The blockage happens in the time interval between 1.53 s and 3.48 s in Fig. 5, leading to a partial link outage.

Figure 5a shows that both PHY and MAC layer throughput decrease and assume zero values during the blockage but recover quickly after removing the blocker. The PHY layer drop and recovery happen in a matter of a few ms, while the MAC layer is slower (tens of ms) because of the release of frames from previous retransmissions.

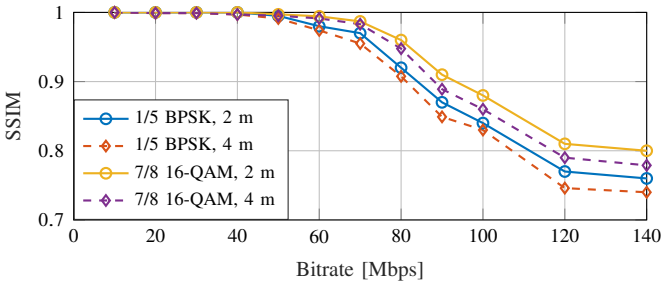
In particular, at the end of the blockage period, there is a spike in retransmissions of unacknowledged frames, as shown in Fig. 5b. This makes a case for dynamic sizing of the TX/RX ARQ buffers based on channel conditions, which we will explore as part of our future work. At t = 349 ms, the maximum number of retransmissions is 33, compared to an average of 1, to recover the lost frames.

Video streaming results. We report the SSIM for video streaming in Fig. 6 for different MCSs, with and without retransmissions, and for different distances.

Figure 6a clearly shows that the support of reliable lower layers of the protocol stack through retransmission improves the SSIM, with high-quality streaming possible only when retransmissions are used. As a reference, we compare in Fig. 7 an example of a video frame received without retransmissions (Fig. 7a), which clearly shows distortions throughout the entire frame, and with retransmissions (Fig. 7b), which presents no visible distortion, thus leading to a high SSIM score. For the trend of the SSIM with retransmissions, it is possible to identify four regions. For a video bitrate lower than 40 Mbps, the SSIM is 1, i.e., there are no errors in the received frames. Between 40 and 80 Mbps, the SSIM score starts decreasing, but the SSIM is still above 0.9, associated with unrecognizable

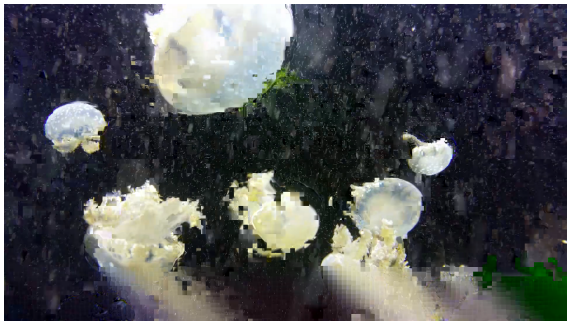


(a) SSIM with (dashed) and without (solid) retransmissions, distance 4 m.



(b) SSIM variation with distance, solid line for 2 m and dashed line for 4 m.

Figure 6: SSIM analysis for different MCSs, retransmission, and distances.



(a) Without retransmissions.

(b) With retransmissions.

Figure 7: Example of a video frame during a transmission at distance 2 m with MCS 1/5 BPSK, with and without retransmissions.

distortion in the literature [28], [29]. Notably, 1/5 BPSK and 1/4 QPSK have an SSIM above 0.9 for up to 80 Mbps, and the other modulations for 90 Mbps. Above these thresholds, the SSIM is below 0.9, and it plateaus between 0.75 and 0.8 above 120 Mbps. This is due to limitations that persist in the asynchronous queue between the DSP FPGA and the host at the receiver: the link layer guarantees perfect reliability, but the host cannot process the received packets fast enough. We

will be working on relaxing this constraint in future iterations of the platform. Finally, as expected, Fig. 6b shows that the SSIM is slightly higher for distance 2 m than 4 m, for both MCS 7/8 16-QAM and 1/5 BPSK.

V. Conclusions

This paper presented the implementation and performance characterization of a software-defined, fully programmable, end-to-end platform for 60 GHz experiments based on real traffic and applications. Notably, we focused on the validation of a video streaming use case. To do so, we enhanced the NI MTS platform with retransmissions at the MAC layer, as well as the possibility of streaming real data in and out of the FPGAs that process the lower layers of the protocol stack. We presented an extensive performance evaluation with different MCSs, distances, ARQ configurations, video bitrates, evaluating link and application parameters.

Challenges and future work. The implementation of ARQ and video streaming on a real-time system that processes more than 3 Gsamples/s has required a careful design in terms of timing and synchronization, as well as memory consumption. The latter has introduced a practical constraint that has limited the evaluation to the 16-QAM modulation, while, as previously discussed, timing issues at the receiver lead to supporting application data rates which are below the full throughput that the PHY layer can achieve.

As future work, we will explore strategies to overcome these limitations, e.g., by moving more higher-layer processing directly on the FPGAs thus reducing the requirements on the interface with the host. In addition, we will leverage this platform to implement and study adaptive, cross-layer algorithms for the joint optimization of application and link performance. We will integrate an adaptive modulation and coding scheme to properly switch the MCS based on the channel conditions and the video requirements, and dynamic ARQ buffer management algorithms that can work with a closed-loop with the video streaming application.

References

[1] M. S. Elbamby, C. Perfecto, M. Bennis, and K. Doppler, "Toward Low-Latency and Ultra-Reliable Virtual Reality," *IEEE Network*, vol. 32, no. 2, pp. 78–84, March 2018.

[2] S. Rangan, T. S. Rappaport, and E. Erkip, "Millimeter-Wave Cellular Wireless Networks: Potentials and Challenges," *Proceedings of the IEEE*, vol. 102, no. 3, pp. 366–385, Mar. 2014.

[3] H. Shokri-Ghadikolaei *et al.*, "Millimeter Wave Cellular Networks: A MAC Layer Perspective," *IEEE Transactions on Communications*, vol. 63, no. 10, pp. 3437–3458, Oct 2015.

[4] M. I. Rochman *et al.*, "A comparison study of cellular deployments in chicago and miami using apps on smartphones," in *Proceedings of the 15th ACM Workshop on Wireless Network Testbeds, Experimental Evaluation & CHaracterization*, ser. WiNTECH'21. New Orleans, LA, USA: Association for Computing Machinery, 2022, p. 61–68.

[5] M. Giordani *et al.*, "A Tutorial on Beam Management for 3GPP NR at mmWave Frequencies," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, pp. 173–196, First quarter 2019.

[6] S. K. Saha *et al.*, "X60: A programmable testbed for wideband 60 ghz wlans with phased arrays," in *Proceedings of the 11th Workshop on Wireless Network Testbeds, Experimental Evaluation & CHaracterization*, Snowbird, Utah, USA, 2017, p. 75–82.

[7] I. K. Jain *et al.*, "mMobile: Building a MmWave Testbed to Evaluate and Address Mobility Effects," in *Proceedings of the 4th ACM Workshop on Millimeter-Wave Networks and Sensing Systems*, ser. mmNets'20. London, United Kingdom: Association for Computing Machinery, 2020.

[8] K. Buisman and T. Eriksson, "Designing and characterizing MATE, the Chalmers mm-wave MIMO testbed," in *12th European Conference on Antennas and Propagation (EuCAP 2018)*, April 2018, pp. 1–5.

[9] J. O. Lacruz, R. R. Ortiz, and J. Widmer, "A Real-Time Experimentation Platform for Sub-6 GHz and Millimeter-Wave MIMO Systems," in *Proceedings of the 19th Annual International Conference on Mobile Systems, Applications, and Services*, ser. MobiSys '21. Virtual Event, Wisconsin: Association for Computing Machinery, 2021, p. 427–439.

[10] T. Chen *et al.*, "Programmable and Open-Access Millimeter-Wave Radios in the PAWR COSMOS Testbed," in *Proceedings of the 15th ACM Workshop on Wireless Network Testbeds, Experimental Evaluation & CHaracterization*, ser. WiNTECH'21. New Orleans, LA, USA: Association for Computing Machinery, 2022, p. 1–8.

[11] Y. Ghasempour *et al.*, "Multi-Stream Beam-Training for MmWave MIMO Networks," in *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*, ser. MobiCom '18, New York, NY, USA, 2018, p. 225–239.

[12] M. Polese, F. Restuccia, and T. Melodia, "Deepbeam: Deep waveform learning for coordination-free beam management in mmwave networks," in *Proceedings of ACM MobiHoc*, Shanghai, China, 2021, p. 61–70.

[13] S. K. Saha *et al.*, "Performance and Pitfalls of 60 GHz WLANs Based on Consumer-Grade Hardware," *IEEE Transactions on Mobile Computing*, vol. 20, no. 4, pp. 1543–1557, April 2021.

[14] M. Polese *et al.*, "milliProxy: A TCP proxy architecture for 5G mmWave cellular systems," in *51st Asilomar Conference on Signals, Systems, and Computers*, 2017, pp. 951–957.

[15] D. A. Hayes, D. Ros, and O. Alay, "On the importance of TCP splitting proxies for future 5G mmWave communications," in *IEEE 44th LCN Symposium on Emerging Topics in Networking (LCN Symposium)*, Oct 2019, pp. 108–116.

[16] T.-T. Le, D. N. Van, and E.-S. Ryu, "Real-time 360-degree video streaming over millimeter wave communication," in *International Conference on Information Networking (ICOIN)*, 2018, pp. 857–862.

[17] Y. Takaku, Y. Kaieda, T. Yu, and K. Sakaguchi, "Proof-of-Concept of Uncompressed 4K Video Transmission from Drone through mmWave," in *IEEE 17th Annual Consumer Communications Networking Conference (CCNC)*, 2020, pp. 1–6.

[18] O. Abari, D. Bharadia, A. Duffield, and D. Katabi, "Enabling high-quality untethered virtual reality," in *14th USENIX Symposium on Networked Systems Design and Implementation (NSDI 17)*, Boston, MA, 2017, pp. 531–544.

[19] M. Drago *et al.*, "Reliable video streaming over mmwave with multi connectivity and network coding," in *International Conference on Computing, Networking and Communications (ICNC)*, 2018, pp. 508–512.

[20] L. Sun *et al.*, "Multi-Path Multi-Tier 360-Degree Video Streaming in 5G Networks," in *Proceedings of the 9th ACM Multimedia Systems Conference*, ser. MMSys '18. Amsterdam, Netherlands: Association for Computing Machinery, 2018, p. 162–173.

[21] J. Qiao, Y. He, and X. S. Shen, "Improving Video Streaming Quality in 5G Enabled Vehicular Networks," *IEEE Wireless Communications*, vol. 25, no. 2, pp. 133–139, 2018.

[22] Y. Liu, J. Liu, A. Argyriou, and S. Ci, "MEC-Assisted Panoramic VR Video Streaming Over Millimeter Wave Mobile Networks," *IEEE Transactions on Multimedia*, vol. 21, no. 5, pp. 1302–1316, 2019.

[23] NI, "Introduction to the NI mmWave Transceiver System," White Paper, 2019.

[24] S. G. Larew, T. A. Thomas, M. Cudak, and A. Ghosh, "Air interface design and ray tracing study for 5G millimeter wave communications," in *IEEE Globecom Workshops (GC Wkshps)*, Dec. 2013, pp. 117–122.

[25] VideoLAN, "VLC media player," 2006. [Online]. Available: https://www.videolan.org/vlc/index.html

[26] "Jellyfish Video Bitrate Test Files." [Online]. Available: http://www.jell.yfish.us/

[27] FFmpeg Developers, "ffmpeg tool (Version be1d324)," 2016. [Online]. Available: http://ffmpeg.org/

[28] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

[29] "SSIM: Structural Similarity Index." [Online]. Available: https://www.imatest.com/docs/ssim/