Is Wi-Fi 6 Ready for Virtual Reality Mayhem? A Case Study Using One AP and Three HMDs

Costas Michaelides 1, Miguel Casasnovas 1, Daniele Marchitelli 1, and Boris Bellalta 1

1Affiliation not available

October 31, 2023

Abstract

Is Wi-Fi 6 ready for virtual reality (VR) mayhem? The common use case for untethered VR is running a video game on a powerful desktop and streaming the generated content to a head-mounted display (HMD) over Wi-Fi. In this work, we investigated a challenging use case using one Wi-Fi 6 access point (AP) and three HMDs. Specifically, we evaluated experimentally the performance of VR streaming in terms of latency using: i) the distributed coordination function (DCF), ii) orthogonal frequency-division multiple access (OFDMA) in the downlink (DL) and the uplink (UL) and iii) multi-user multiple input multiple output (MU-MIMO). We showed that VR streaming to three users, up to 100 Mbps each, works flawlessly using either DCF or DL OFDMA, with latency in the order of 50 ms. In addition, our study unveiled the disruption of tracking data transmissions in the uplink using UL OFDMA, which had a negative impact on latency by delaying game rendering.
Is Wi-Fi 6 Ready for Virtual Reality Mayhem? A Case Study Using One AP and Three HMDs

Costas Michaelides, Miguel Casasnovas, Daniele Marchitelli and Boris Bellalta

Department of Information and Communication Technologies
Universitat Pompeu Fabra
Barcelona, Spain
Email: firstname.lastname@upf.edu

Abstract—Is Wi-Fi 6 ready for virtual reality (VR) mayhem? The common use case for untethered VR is running a video game on a powerful desktop and streaming the generated content to a head-mounted display (HMD) over Wi-Fi. In this work, we investigated a challenging use case using one Wi-Fi 6 access point (AP) and three HMDs. Specifically, we evaluated experimentally the performance of VR streaming in terms of latency using: i) the distributed coordination function (DCF), ii) orthogonal frequency-division multiple access (OFDMA) in the downlink (DL) and the uplink (UL) and iii) multi-user multiple input multiple output (MU-MIMO). We showed that VR streaming to three users, up to 100 Mbps each, works flawlessly using either DCF or DL OFDMA, with latency in the order of 50 ms. In addition, our study unveiled the disruption of tracking data transmissions in the uplink using UL OFDMA, which had a negative impact on latency by delaying game rendering.

Index Terms—video games, virtual reality, wireless networks, Wi-Fi.

I. INTRODUCTION

Virtual reality (VR) streaming to multiple users is an emerging use case. For example, immersive art installations such as Delirious Departures (see [1]) may include several VR users, each sending tracking data in the uplink and receiving video frames in the downlink. Such real-time use cases are pushing current technology to its limits. Having in mind that VR is particularly sensitive to latency, we investigate a challenging use case over Wi-Fi that includes one access point (AP) and three head-mounted displays (HMDs).

There are two fundamental aspects of latency in video games: responsiveness and consistency (see [2], [3] and references therein). Both of them are relevant to VR streaming systems. On the one hand, a video game is responsive when the time interval between a physical movement and a virtual action in the game is low. On the other hand, consistency is an attribute of multiplayer games, where the interaction between users implies that their game states should be synchronized. Therefore, low latency is desired for both responsiveness and consistency. In the case of VR streaming, latency is the sum of several delays, including the encoding, the decoding and the network delay. Network delay cannot be eliminated but can be reduced if the content is generated at the edge, close to the user. Moreover, a common way to experience VR is to use a Wi-Fi setup with a desktop computer as a server. Here, we focus on the last hop of such systems: Wi-Fi.

Wi-Fi is the most appealing solution for VR in terms of comfort. However, the feasibility of serving multiple users running real-time VR applications is still an open issue and a long-term goal for future Wi-Fi generations. Apart from the baseline distributed coordination function (DCF), Wi-Fi 6 includes multi-user (MU) enhancements in the downlink (DL) and the uplink (UL) for high-throughput applications, including orthogonal frequency-division multiple access (OFDMA) and multi-user multiple input multiple output (MU-MIMO). Please see also an overview of these features in [4] and an in-depth tutorial in [5].

There are several studies on VR streaming over Wi-Fi that focus on latency. The authors in [6] proposed a framework that combines the rendering on the device with remote rendering. Moreover, the authors in [7] proposed using multiple interfaces to increase the probability of timely reception. The study in [8] focused on the uplink transmissions. The authors proposed several ways to reduce latency, such as prioritizing the tracking data. In [9], the authors presented an experimental VR cloud streaming platform using a COSMOS node (see [10]) as an AP. There is also a study on 60 GHz Wi-Fi using ns-3 (see [11]). Lastly, we would like to mention a study on traffic modeling. The authors in [12] considered both local and cloud rendering and proposed a model for the video frame size and one for the interarrival time.

These works refer to previous generations of Wi-Fi and focus on the case where one AP serves one HMD. In this work we investigated experimentally the feasibility of serving multiple users over Wi-Fi 6. Specifically, we evaluated the performance of VR streaming using one AP and three HMDs, multiple encoding bitrates and multiple Wi-Fi configurations. Our main findings are summarized as follows:

1) It works! One AP can serve three HMDs using either DCF or DL OFDMA,

2) The scheduling of UL OFDMA disrupts the transmission of the tracking data and increases latency by delaying the rendering of new frames by the video game.

To the best of our knowledge, this is the first work in the literature that evaluates VR streaming to multiple users in the same Wi-Fi 6 network. We hope that our findings will motivate further research on Wi-Fi in order to identify and solve issues that may have a negative impact on VR systems.
The rest of this work is structured as follows. In Section II we introduce the preliminaries and in Section III our setup. In Section IV we present our results and in Section V we discuss our findings and highlight some indicative research directions. Section VI concludes this work.

II. PRELIMINARIES

In this section we introduce the VR streaming system and present the main Wi-Fi features in order to facilitate the discussion of our results.

A. The VR streaming system

The contemporary VR streaming system includes a streamer running on a powerful server and a client application running on an HMD. The server is expected to be at the edge, close to its clients. In our case, the server is connected to the AP over Ethernet and to the HMDs over Wi-Fi.

The goal of each client is to display video frames at a constant rate taking into account the user’s actions. In practice, there are several correlated processes running sequentially and in parallel. As depicted in Fig. 1, the client queries the pose and transmits tracking data in the uplink, while the server runs a video game that renders frames based on the received input. Each frame is encoded and transmitted in the downlink. Upon reception, each frame is decoded and sent to vsync to be displayed. In this abstract system there are two crucial components: 1) the video game and 2) the display vsync. On the one hand, the video game receives information about the pose and the controllers and renders frames at a constant rate, e.g. 120 fps. On the other hand, vsync receives video frames and displays them also at a constant rate, preferably the same one as the video game. These processes are decoupled but closely related. We need timely updates in the uplink to maintain consistency and timely updates in the downlink to achieve responsiveness.

A crucial part of this system is prediction. Based on Fig. 1, the expected (albeit optimistic) value of latency is \( \sim 45 \) ms. In order to compensate for this latency, the client always reports to the game its future state. According to Oculus VR (OVR) metrics\(^1\), if we do not have any significant delays, the prediction time should be between 45 ms and 50 ms.

\(^1\)https://developer.oculus.com/documentation/unity/ts-ovrstats/

B. Wi-Fi 6 essentials

The basic medium access control (MAC) scheme in Wi-Fi is DCF, based on the carrier sense multiple access with collision avoidance (CSMA/CA) protocol and optional request to send (RTS) and clear to send (CTS) messages. The exchange of RTS/CTS is a simple yet effective way to make a reservation for a future transmission and inform as many contending stations (STAs) as possible. The data are transmitted in MAC protocol data units (MPDUs) or aggregated ones (A-MPDUs), and their reception is acknowledged by the receiver with an ACK for each MPDU or a block ACK (BACK) for multiple ones. This procedure is depicted in Fig. 2a. Please see [13] for an in-depth discussion and evaluation of DCF.

OFDMA was introduced by the IEEE 802.11ax amendment, also known as Wi-Fi 6. OFDMA can be enabled on top of DCF and relies mostly on scheduling. The scheduling is coordinated by the AP, which allocates resource units (RUs) to the STAs either for the downlink or the uplink transmissions. In the case of DL OFDMA, Fig. 2b shows the exchange of multi-user RTS (MU-RTS) and CTS and the transmission of multiple A-MPDUs from the AP to multiple STAs in the downlink, followed by the BACKs in the uplink. Moreover, in UL OFDMA (see Fig. 2c), the AP triggers the transmissions from multiple STAs using a Trigger frame, followed by a multi-STA BACK (MS-BACK) in the downlink. On top of DL/UL OFDMA we may also enable MU-MIMO for spatial multiplexing per RU. For more information on OFDMA in Wi-Fi please see [14] and [15].
III. EXPERIMENTAL SETUP

In this section we introduce our experimental setup. First we present our equipment and then the selected parameters and settings.

Our setup is depicted in Fig. 3. We used one AP and three HMDs connected over Wi-Fi. The traffic was generated by one desktop and two laptops. Each one of these devices was connected to the AP over Ethernet and was associated with one HMD. Therefore, the experiments were performed by three users: Costas on the desktop, Miguel on the first laptop and Daniele on the second one. The experiments took place in the same room and each user was seated in front of his device in order to save the logs of each test. Each one of us collected his own Air Light VR (ALVR)\(^2\) statistics (.json) and Wireshark traces (.csv), using different configurations on the AP.

The main component of our setup was the AP. At the time of writing, the Asus ROG Rapture GT-AXE11000 was one of the most powerful Wi-Fi APs and supported the latest features, including DL/UL OFDMA and MU-MIMO. In addition, the desktop and the laptops have been evaluated using the SteamVR\(^3\) Performance Test as VR Ready. On the desktop Costas played Half-Life: Alyx and on the laptops Miguel and Daniele played SteamVR Home. Both games were developed by Valve using the Source 2 engine. Moreover, we used the same version of the ALVR streamer on each device. The generated traffic by the three devices was streamed to the associated Meta Quest 2 HMDs over Wi-Fi. Please refer to Table I for a detailed description of our equipment.

Our goal was to put some real stress on our AP in order to identify the implications of using the following multi-user options: DL OFDMA, DL/UL OFDMA and MU-MIMO on top of DL/UL OFDMA. We used the following encoding bitrates: 50 Mbps, 100 Mbps and 200 Mbps, which indicate an increase in image quality. We relied mostly on the default parameter values of ALVR. Our most crucial parameter is the frame rate, since it drives the performance of the system in terms of latency. We used 120 fps which is the maximum frame rate that is supported by our HMDs. In the network settings we selected the user datagram protocol (UDP). At the AP we enforced the use of Wi-Fi 6 (IEEE 802.11ax) at 5 GHz with an 80 MHz bandwidth. See also Table II for an overview of our parameters and settings.

---

\(^2\)https://github.com/alvr-org/ALVR

\(^3\)https://www.steamvr.com
IV. PERFORMANCE EVALUATION

In this section we present our experimental procedure and our results, including a subjective evaluation. The full dataset of our results is available on Zenodo (see [16]).

A. Experimental procedure

Our experiments were performed in two parts. In the first part, i.e. the single-user tests, each user took turns playing a game using different encoding bitrates: 50 Mbps, 100 Mbps and 200 Mbps. This was an opportunity to set a baseline for our expectations and also a safe check that our devices were capable of running the selected video games. In the second part, i.e. the multi-user tests, the three of us were playing at the same time using the same set of bitrates. We repeated these tests using four different configurations. First we used the DCF, next we enabled DL OFDMA and then DL/UL OFDMA. Finally, we enabled MU-MIMO on top of DL/UL OFDMA. The duration of each test was ~2 minutes.

The variable in our experiments was the encoding bitrate. We also refer to this as the target bitrate. Our main metrics are the network delay and the latency, which were calculated per video frame. Network delay includes both downlink and uplink delays. Ideally, we expect this to be in the order of 5 ms. Latency is an objective metric that includes every possible delay, and shows clearly whether the system works or not. We expect this to be between 45 ms and 50 ms, in agreement with the recommendation in OVR metrics.

B. Experimental results

Fig. 4 shows the single-user results of Costas per video frame. Please consider this as a baseline of our expectations for the multi-user results. First, note that the sum of the server-side delays (game, encoder) is much lower than the client-side sum (decoder, vsync). On the server side, notice the game delay. This is the time that has passed since the last tracking data was received. A low value of game delay indicates the timely reception of tracking data. It appears here to be somewhat higher than our expectations. Nevertheless, it follows the delay imposed by the frame rate (~8 ms for 120 fps). On the client side, notice the extended range of delays at the decoder queue. This is our dejittering buffer! Lastly, note that network delay is well below 10 ms and latency is in the order of 50 ms. Overall, we observe a stable performance without any significant delays during gameplay.

Fig. 5 shows the multi-user results of Costas, Miguel and Daniele using a target bitrate of 50 Mbps and several configurations on the AP. Our observation here is a remarkable increase in latency in the cases of DL/UL OFDMA and MU-MIMO (on top of DL/UL OFDMA). Indeed, the game delay jumps from 10 ms to 25 ms. The packet traces of Costas in Fig. 6 are enlightening. In the downlink we have the audio and video packets that follow the frame rate. In the uplink we have two streams of updates: 1) our statistics at the bottom (one update per frame) and 2) the tracking data. In the case of DL/UL OFDMA, we observe a periodic scheduling in the uplink that disrupts the delivery of tracking data. Despite the fact that we do not observe any extra network delay, this disruption adds a significant burden to our system by delaying game rendering.
Fig. 7. Multi-user results of Costas, Miguel and Daniele at 100 Mbps: (a) Network delay and (b) Latency.

Fig. 8. Multi-user results of Costas, Miguel and Daniele at 200 Mbps: (a) Network delay and (b) Latency.

Fig. 7 shows the multi-user results using a target bitrate of 100 Mbps. These results are in agreement with our previous discussion regarding the disruption due to the scheduling in the uplink. In addition, we have one interesting remark that highlights the peculiarities of this system. Here, the network delay has increased considerably compared to the case of 50 Mbps; it is in the order of ~ 10 ms. However, the total latency is in the range of 50 ms – 75 ms as in Fig. 5. Indeed, this is a case where the extra delay is tolerated by the system due to its discrete time intervals of execution. In general, high tolerance is expected at low frame rates. It is quite remarkable that we can still enjoy some tolerance at 120 fps.

Fig. 8 shows the point where our system collapse. Using a target bitrate of 200 Mbps, the network delay is clearly the major part of the total latency. Even at this point, DCF and DL OFDMA outperform DL/UL OFDMA and MU-MIMO. However, in all cases, the system is severely affected to the point that the games are not responsive. Latency goes well beyond 50 ms, and the prediction is simply not accurate any more. Consider the case of a user turning his head towards one direction. As mentioned earlier, we assumed that predicting this pose in the near future is feasible: It should be towards the same direction! Unfortunately, this is most probably not the case beyond 100 ms of latency.

Table III provides an overview of our experience using a good-bad-ugly scale. This is a subjective evaluation that summarizes our observations. We rated DCF and DL OFDMA up to 100 Mbps as good. Indeed, the multi-user experience was as good as the single-user one. We also noticed that DL/UL OFDMA and MU-MIMO (on top of DL/UL OFDMA) introduced extra latency. Up to 100 Mbps, the games were still playable, but we rated them as bad since the gameplay was less responsive compared to our previous experience with DCF and DL OFDMA. Finally, at 200 Mbps, we rated everything as ugly: Using any AP configuration, each turn of the head caused severe visual issues to the point where our games were not playable and definitely not responsive.

V. LESSONS LEARNED

In this section we provide a discussion of our results and some research opportunities based on our findings.

A. Discussion

We investigated a challenging use case where each user connected his device to the AP over Ethernet and streamed the generated content to his HMD over Wi-Fi. We showed that VR streaming to three users over Wi-Fi, up to 100 Mbps each, works remarkably well using DCF or DL OFDMA.

Surprisingly, our main finding was related to the uplink instead of the downlink. We observed that the scheduling of UL OFDMA disrupts the delivery of the tracking data. Specifically, the grouping of uplink transmissions does not allow the users to transmit their pose updates as soon as possible. This has a negative impact on the VR system because it delays the video game’s frame rendering process. In contrast, we also observed that a small increase in network delay can be tolerated by the system, depending on the frame rate, without any significant impact on latency.

B. Research opportunities

Having in mind the peculiarities of VR streaming, i.e. the periodic transmissions in the downlink and the opportunistic ones in the uplink, we would like to highlight some indicative research directions. A promising task is revisiting random access schemes. For example, using OFDMA, we may reserve some RUs for random access in the uplink (see [17]). Another interesting approach is the use of repeated contentions in the frequency domain (see [18]). Moreover, the upcoming features of Wi-Fi should be investigated in the context of real-time applications, specifically multi-link operation (MLO) (see [19]) and multi-AP coordination (see [20]).
VI. Conclusion

In this work we investigated experimentally the feasibility of VR streaming to multiple users over Wi-Fi 6. Specifically, we showed that the case of one AP and three HMDs, using up to 100 Mbps encoding bitrate, is perfectly feasible using either DCF or DL OFDMA. In contrast, using DL/UL OFDMA, we identified significant disruptions of the tracking due to the scheduling in the uplink. We hope that our work will be useful to users and the research community for further research on Wi-Fi as the last hop of VR streaming systems.

Acknowledgments

Costas Michaelides would like to thank Xavier Trilla, CEO at Clouding.io, for his enlightening comments regarding the uplink issues of VR streaming.

The authors would like to thank the Interactive Technologies Group of Universitat Pompeu Fabra for providing one of the HMDs and Marc Carrascosa-Zamacois for taking photos during the tests.

This work has been partly funded through projects MAX-R (101070072, EU HORIZON-CL4-2021), Wi-XR PID2021-123995NB-I00 (MCIU/AEI/FEDER,UE), and by MCIN/AEI /10.13039/501100011033 under the Maria de Maeztu Units of Excellence Programme (CEX2021-001195-M).

References