Are We Ready for Metaverse? A Measurement Study of Social Virtual Reality Platforms

Ruizhi Cheng
George Mason University
rcheng4@gmu.edu

Nan Wu
George Mason University
nwu5@gmu.edu

Matteo Varvello
Nokia Bell Labs
matteo.varvello@nokia.com

Songqing Chen
George Mason University
sqchen@gmu.edu

Bo Han
George Mason University
bohan@gmu.edu

Abstract

Social virtual reality (VR) has the potential to gradually replace traditional online social media, thanks to recent advances in consumer-grade VR devices and VR technology itself. As the vital foundation for building the Metaverse, social VR has been extensively examined by the computer graphics and HCI communities. However, there has been little systematic study dissecting the network performance of social VR, other than hype in the industry. To fill this critical gap, we conduct an in-depth measurement study of five popular social VR platforms: AltspaceVR, Horizon Worlds, Mozilla Hubs, Rec Room, and VRChat. Our experimental results reveal that all these platforms are still in their early stage and face fundamental technical challenges to realize the grand vision of Metaverse. For example, their throughput, end-to-end latency, and on-device computation resource utilization increase almost linearly with the number of users, leading to potential scalability issues. We identify the platform servers' direct forwarding of avatar data for embodying users without further processing as the main reason for the poor scalability and discuss potential solutions to address this problem. Moreover, while the visual quality of the current avatar embodiment is low and fails to provide a truly immersive experience, improving the avatar embodiment will consume more network bandwidth and further increase computation overhead and latency, making the scalability issues even more pressing.

CCS Concepts

• **Networks → Network measurement**; • **Computing methodologies → Virtual reality**; • **Human-centered computing → Mobile computing**.

ACM Reference Format:
Ruizhi Cheng, Nan Wu, Matteo Varvello, Songqing Chen, and Bo Han. 2022. Are We Ready for Metaverse? A Measurement Study of Social Virtual Reality Platforms. In *Proceedings of the 22nd ACM Internet Measurement Conference (IMC '22), October 25–27, 2022, Nice, France.* ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3517745.3561417

IMC '22, October 25–27, 2022, Nice, France

ACM ISBN 978-1-4503-9259-4/22/10.
https://doi.org/10.1145/3517745.3561417

1 Introduction

Social virtual reality (VR) enables users around the world to interact and socialize with each other in a shared, interoperable, and virtual environment. In a virtual social world, users are embodied by digital avatars (*i.e.,* a first-person rendition via 3D models). The Metaverse, deemed as a hypothetical next-generation Internet [15], aims to be a utopian convergence of various virtual environments and eventually blend the physical and digital world [46]. While there is still no consensus on the definition of Metaverse, among its rudimentary prototypes, social VR is the closest to the original vision described in the science-fiction novel *Snow Crash*, where the term Metaverse was first coined in 1992.

Although the concept of the Metaverse has been around for about half a century (§2), the outbreak of the COVID-19 pandemic accelerated the digitization of our daily lives [6], leading to the renaissance of the Metaverse and the prosperity of social VR platforms [83]. Social VR has been deeply studied by the computer graphics and human-computer interaction (HCI) communities in recent years [34, 42, 44, 45, 53, 54, 76, 89, 90, 103]. Nonetheless, there has been limited thorough and systematic investigation on characterizing and quantifying its (network) performance.

Motivated by this crucial gap, we conduct, to the best of our knowledge, the first detailed measurement study of five popular social VR platforms: AltspaceVR [64], Horizon Worlds (hereinafter referred to as Worlds) [63], Mozilla Hubs (referred to as Hubs) [67], Rec Room [79], and VRChat [99]. Our study aims to answer the following research questions.

- What are the network protocols and infrastructure (*e.g.,* server locations) employed by social VR platforms?
- What content is being delivered for social VR, and can the current network sustain its bandwidth demand?
- How will the network throughput and on-device computation resource utilization change with the number of users?
- What is the end-to-end latency that existing social VR platforms can offer?
- How will these platforms respond to dynamic network conditions such as fluctuating available bandwidth?

Our measurement study reveals that, as early prototypes of the Metaverse, all five social VR platforms confront intrinsic technical challenges, although some of them have been available for more than 7 years[1]. For example, their digital avatars are still underdeveloped, and they can support only a small number of concurrent

[1] AltspaceVR was first released in 2015 and acquired by Microsoft in 2017.

users (*e.g.,* <20 in a social event of Worlds [73]). We summarize our key findings, as the main contribution of this paper, as follows.

(1) Social VR platforms employ different network protocols for control and data channels. Not all platforms use servers from the same provider for the two types of channels. Some platforms allocate servers farther away from end-users with >70ms round-trip time (RTT) (§4).

(2) With two users socializing in a private event, the throughput of continuous data exchange on all platforms is lower than 100 Kbps, except Worlds whose throughput is ~750 Kbps for uplink and ~410 Kbps for downlink. The throughput is independent of content resolution and is mainly contributed by avatar embodiment and motion. The platform servers directly forward avatar data among users without further processing (§5).

(3) All platforms have latent scalability issues with the throughput increasing almost linearly when more users join a social event. Moreover, as the number of users grows, the on-device resource utilization climbs and the FPS (frames per second) of each platform degrades. This is due to the simple forwarding of all avatar data from one user to others without optimization. Only AltspaceVR benefits from the viewport-adaptive optimization that delivers data for only avatars visible to a user. We identify remote rendering that offloads the processing of visual content to cloud/edge servers as a promising solution to address the scalability issues (§6).

(4) Hubs has the highest end-to-end latency among these platforms, because it is Web-based and cannot always allocate servers close to users. AltspaceVR has the highest server-processing latency, which is likely caused by the viewport-adaptive optimization. The receiver-side processing latency of all platforms, except AltspaceVR, is higher than the latency on the servers. Finally, the end-to-end latency also exhibits questionable scalability issues (§7).

(5) There is an interplay between the downlink bandwidth of Worlds and its uplink data transfer and CPU/GPU utilization on VR headsets. The TCP uplink traffic of Worlds has a higher priority than its UDP uplink, which is blocked until TCP packets have been successfully delivered (§8).

Our findings have broad implications for the future development of the Metaverse. To serve billions of users all over the world, the network infrastructure and system architecture of the Metaverse should be designed with scalability in mind, differently from the current practice. Furthermore, the avatar embodiment should be drastically enhanced to offer a satisfactory and truly immersive user experience. Nevertheless, this improvement will essentially demand more network bandwidth, prolong the end-to-end latency, and stretch the computation resources on VR headsets. We have released the source code used in this paper at https://github.com/felixshing/Metaverse_IMC2022. This work does not raise any ethical issues.

2 Background

2.1 Social VR and Metaverse

The design patterns of existing social VR platforms are similar. After users launch the application, they will first stay on the welcome page for system initialization. They can then choose the social interaction to experience next, which could be, for example, either a public event such as a concert or a private event such as an online meeting. Social VR platforms offer numerous features that can be divided into two categories. As basic features, these platforms all enable users to walk and chat in a virtual space (*e.g.,* a conference room). In terms of advanced features, users can interact more with each other and the platforms, such as playing games, creating user-generated content (UGC), and shopping/trading with non-fungible tokens (NFTs) [102].

The Metaverse strives to create a shared virtual world by bridging all virtual environments through the Internet. The development of the Metaverse started with text-based interactive games (*e.g.,* MUD, multi-user dungeon) in the late 1970s, followed by another wave around 2000 represented by Second Life, an online virtual world. Besides social VR, other recent developments of the Metaverse include massively multiplayer online games, such as Roblox [82], Fortnite [24], and Minecraft [65], and the emerging blockchain or NFT-based online games, such as Axie Infinity [56], Decentraland [21], and Upland [95]. However, as they are designed mainly for PC users with 2D content, these games currently cannot afford an immersive experience for their users, one of the most important goals of the Metaverse. Thus, we focus on the investigation of social VR in this paper.

2.2 Mobile VR

Mobile VR aims to present computer-generated virtual content, in real-time, to users of *untethered* headsets. A key challenge of mobile VR is to render high-quality content at a fast pace (*i.e.,* a high frame rate), which is computation-intensive. This can be achieved by either local or remote rendering, which we detail in the following.

Local Rendering. VR headsets can solely rely on their own computation resources (*e.g.,* CPU and GPU) for supporting the entire rendering pipeline. This requires powerful CPUs and GPUs that untethered headsets such as Oculus Quest 2 are usually not equipped with, mainly caused by their small form factor. A heavier VR headset may lead to motion sickness due to pulling the user's head forward and down [41, 94], especially when wearing it for a long time [16, 109]. As a result, local rendering on untethered VR headsets can offer only medium content resolution (*e.g.,* ~2K) and refresh rate (*e.g.,* 60–70). In contrast, tethered VR headsets such as HTC VIVE can achieve a higher refresh rate (*e.g.,* 90–120) by attaching to a PC with high-end GPUs for rendering.

Remote Rendering. With recent advances in cloud/edge computing, modern VR systems switch from rendering all virtual content locally on headsets to offloading the processing of at least part of the content to a server [43, 50, 61, 107]. After rendering the content, the server transfers back encoded frames to VR headsets as a video stream for display. This approach, known as remote rendering, is promising for VR because it can potentially make the headset a thin client by reducing its weight, which may alleviate motion sickness. Remote rendering has also been widely embraced in cloud gaming platforms [12, 22, 37] and immersive video streaming [86, 111]. The key requirements for remote rendering are that the available bandwidth between server and VR device should be high (*e.g.,* >25 Mbps for cloud gaming [37]), and the network latency should be low, as a prolonged end-to-end latency may deteriorate the user experience.

Platforms	Company	Locomotion	Facial Expression	Personal Space	Game	Share Screen	Shopping	NFT
AltspaceVR ('15)	Microsoft	Walk, Teleport	✗	✓	✓	✓	✗	✗
Rec Room ('16)	Rec Room	Walk, Jump, Teleport	✓	✓	✓	✗	✓	✓
VRChat ('17)	VRChat	Walk, Jump, Teleport	✓	✓	✓	✗	✗	✗
Hubs ('18)	Mozilla	Walk, Fly, Teleport	✗	✗	✗	✓	✗	✗
Worlds ('21)	Meta	Walk, Teleport	✓	✓	✓	✗	✗	✗

Table 1: Comparison of several important features offered by five social VR platforms (NFT – non-fungible token). Teleport means instantaneous transport from one location to another without moving step by step.

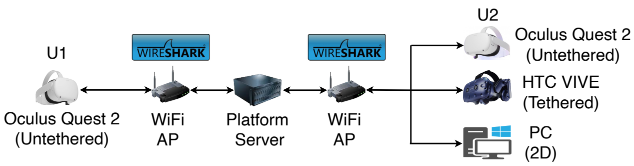


Figure 1: Measurement setup with two users U1 and U2.

3 Measurement Setup

In this section, we describe the social VR platforms under investigation, the setup of our testbed, and the performance metrics used in our measurement study.

3.1 Social VR Platforms

We study five prevalent social VR platforms: AltspaceVR [64], Horizon Worlds [63], Mozilla Hubs [67], Rec Room [79], and VRChat [99]. We choose these platforms because of their popularity [83], and because they have been extensively studied by the computer graphics and HCI research communities, for example, from the visual content and social interaction perspectives [29, 39, 42, 53, 58, 59]. In Table 1, we summarize and compare some of the unique features offered by these platforms. Hubs is a Web-based application, while the others are standalone applications for VR headsets and PCs. Given that Hubs open-sources its code [69], besides measuring the public Hubs service from Mozilla, we deploy a private Hubs server on an Amazon AWS [4] EC2 instance (`t3.medium`) for controlled experiments (§7).

3.2 Testbed & Data Collection

Figure 1 shows our measurement setup. All our experiments, except for measuring the network infrastructure of these platforms (§4.2), were conducted in the eastern U.S. from 02/2022 to 05/2022. Most experiments involve two users, U1 and U2. U1 is always equipped with a Quest 2 VR headset, whereas U2 uses either Quest 2, HTC VIVE Cosmos, or a PC with i7-7700K CPU (4.2 GHz) and GTX 1070 GPU. Note that VIVE is a tethered headset and needs to be connected to the PC for rendering VR content. We conduct most of our experiments with both U1 and U2 using Oculus Quest 2 because it is the most popular VR headset [91] and is representative of how users typically access social VR platforms nowadays. Moreover, Worlds is currently available on only Oculus VR headsets. U1 and U2 are connected to two different WiFi access points (APs) that are attached to a university campus network. We use Wireshark [105] on each AP to capture and analyze network traffic. Meanwhile, we run the OVR Metrics Tool [72], an official performance monitoring tool from Oculus, to measure the performance and resource utilization of client-side social VR applications on Quest 2. There is no other background process on Quest 2 during our experiments. *Unless otherwise mentioned, we report the averaged measurement results from more than 20 experiments.*

In the following, we describe the performance metrics we collect and analyze in this paper:

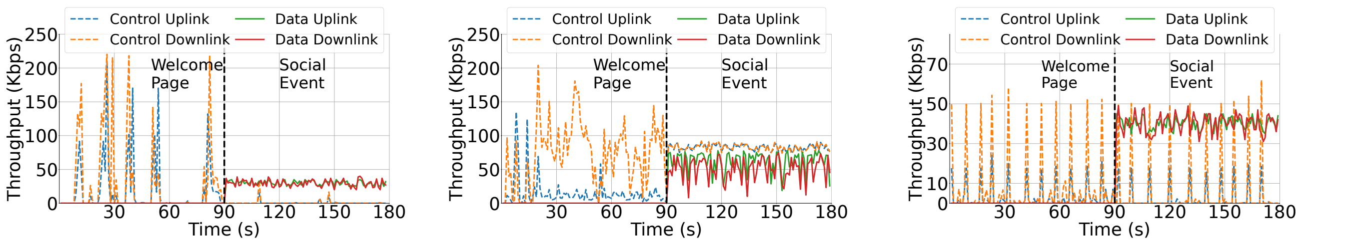
• *Throughput:* We measure the throughput of both the initialization (*i.e.,* welcome page) and social interaction stages.
• *End-to-end Latency*: The end-to-end delay of a social VR platform is the time difference between when one user performs an action and when that action is perceived by others.
• *Average FPS*: Ideally, FPS should be equal to the refresh rate, which, by default, is 72 on Quest 2. VR headsets also keep track of the number of stale frames per second (*i.e.,* how many times the frames are not delivered on time and are substituted with the previous one).
• *Resolution*: The resolution of content rendered by applications. The higher it is, the higher the rendering overhead is, and the better the user experience would be. This resolution is determined by applications. The default display resolution for Quest 2 is 1832×1920 per eye (W×H).
• *Resource Utilization*: These metrics indicate how heavy the computation of an application is, with respect to CPU and GPU usage, memory footprint, and energy consumption.

4 Platform Analysis

In this section, we explore the network protocols and infrastructure (*e.g.,* server locations) of social VR platforms.

4.1 Network Protocols

Our measurement reveals that different network protocols have been employed by these platforms, as summarized in Table 2. Further analysis shows that they can be separated as control-channel (*e.g.,* menu operations and clock synchronization in games) and data-channel (*e.g.,* avatar embodiment and voice data) protocols. We determine that these two channels are distinct based on two findings: 1) Except for Hubs, these two channels are used under different scenarios. As shown in Figure 2, the control channel transmits data when users interact with the platform (mainly on the welcome page), while the data channel transmits data when users interact with each other (*e.g.,* during social events). Both control channel and data channel transmit data in social events of Hubs. 2) As shown in Table 2, the servers that manage these two channels have different owners (*e.g.,* Rec Room and VRChat) or different geolocations and thus RTTs (*e.g.,* AltspaceVR). Moreover, the servers of Worlds have different hostnames (*e.g.,* "edge-star-shv–01-iad3.facebook.com" for the control channel and "oculus-verts-shv-01-iad3.facebook.com"



(a) VRChat (b) Mozilla Hubs (c) AltspaceVR

Figure 2: Throughput of control and data channels for U1 on VRChat, Hubs, and AltspaceVR during experiments with two users (U1 and U2). Users enter a social event at 90s. Rec Room has a similar pattern as VRChat. Worlds has a similar pattern as AltspaceVR. We omit the throughput of initial data downloading of Hubs (§5.2), which is higher than 100 Mbps.

Platform	Control Channel				Data Channel			
	Protocol	Server Loc. / Owner	Anycast?	RTT (ms)	Protocol	Server Loc. / Owner	Anycast?	RTT (ms)
AltspaceVR	HTTPS	- / Microsoft	✓	3.08/0.1	UDP	Western U.S. / Microsoft	✗	72.1/0.2
Hubs	HTTPS	Western U.S. / AWS	✗	74.1/0.3	RTP/RTCP HTTPS	Western U.S. / AWS	✗	73.5/0.2 74.1/0.3
Rec Room	HTTPS	- / ANS	✓	2.21/0.1	UDP	- / Cloudflare	✓	2.97/0.1
VRChat	HTTPS	Eastern U.S. / AWS	✗	2.32/0.3	UDP	- / Cloudflare	✓	3.24/0.3
Worlds	HTTPS	Eastern U.S. / Meta	✗	2.23/0.2	UDP	Eastern U.S. / Meta	✗	2.71/0.1

Table 2: Network protocols and infrastructure of five social VR platforms (experiments are conducted on the U.S. east coast). The server location is marked as – when anycast is used. The two numbers for RTT are the average and standard deviation.

for the data channel). The official documentation of Hubs shows that its HTTPS server (control channel) is a set of load-balanced nodes, and its WebRTC server (data channel) acts as a central routing machine [68].

Control Channel. All five social VR platforms use HTTPS to transfer data that is related to the control plane, for example, when users select an option from the menu. The HTTPS connections of AltspaceVR and Worlds have periodic spikes every ~10s. The throughput of these spikes is low, 50/17 Kbps (downlink/uplink) for AltspaceVR and 300 Kbps (uplink) for Worlds. There is no downlink spike for Worlds. We infer that these periodic data transfers are reports of client-side information. Our subsequent experiments discover that one role of the periodic HTTPS spike of Worlds is to synchronize the clock between users when they play games (§8.1).

Data Channel. All platforms, except Hubs, use UDP to deliver information that belongs to the data plane, for example, audio content and avatar motion. Based on the official documentation and source code of Hubs [68, 69], we find that it uses WebRTC [35] to deliver voice data, while other information, such as the avatar's location, is still transmitted via HTTPS. Note that when remote rendering is used in commercial cloud gaming platforms, WebRTC is widely adopted to deliver the resulting video streams [12, 22, 37]. However, we do not observe any WebRTC-based video data on all five social VR platforms when users interact with each other (*e.g.,* wandering and chatting), which indicates that remote rendering may not be exploited yet by these platforms.

4.2 Network Infrastructure

Next, we analyze the five platforms' server locations and network latency. We use ICMP and TCP (when ICMP is blocked) `pings` to estimate the RTT between our WiFi APs and platform servers. However, they both fail for the data-channel server of Hubs. Since Hubs uses WebRTC, we can get RTT statistics from Chrome's debugging console [30] via the `RTCIceCandidatePairStats` method.

We use MaxMind [57] and ipinfo.io [36] to geolocate the IP address of each server identified in our experiments. With a combination of `ping` and `traceroute`, we infer whether these servers rely on *anycast* [49, 60], a network addressing approach where the same IP address is used by multiple servers at different physical locations. The goal of anycast is to bring the service closer to end-users, without abusing DNS [98]. We use `traceroute` to the identified platform servers from three locations (the northern U.S., eastern U.S., and the Middle East) and analyze the IP address of each hop and the RTT between our test machines at the three locations and the platform servers. Since our machines are located in different places, if the RTT between them and the platform server is comparable and/or there is a significant difference in the IP addresses of the hops right before reaching the platform server, it implies that this server relies on anycast.

Control Channel. As shown in Table 2, VRChat and Worlds consistently assign us HTTPS servers (<3ms RTT) close to our WiFi APs. On the other hand, Hubs always assigns servers with >70ms RTT from our WiFi APs; geolocation via `traceroute` and MaxMind suggests that those servers are likely located on the U.S. west coast. With the method presented above, we find that AltspaceVR and Rec Room rely on anycast for addressing their servers.

Data Channel. To deliver data related to avatars (*e.g.,* motion and facial expressions), all platforms switch to a new server (*i.e.,* different from the one used for control channels). While the HTTPS server for Hubs does not change, the RTP/RTCP server for WebRTC is different. AltspaceVR and Hubs always assign servers in the western U.S. to serve users, resulting in >70ms RTT. Conversely, Worlds consistently assign nearby servers with <3ms RTT. VRChat and Rec Room's servers rely on anycast, providing <4ms RTT.

We next leverage WHOIS data to comment on the usage of third-party cloud services for deploying social VR platforms. AltspaceVR (acquired by Microsoft) and Worlds (developed by Meta) deploy the platforms on their own servers, whereas Hubs relies on Amazon AWS [4]. Rec Room and VRChat rent servers for data channels from Cloudflare [17], and control-channel servers from AWS (VRChat) and Advanced Network and Services [2] (Rec Room).

Most platforms allocate our two test users, even when they are physically co-located and use the same access network, to two different servers, possibly for load balancing purposes. Only AltspaceVR and Hubs (for RTP/RTCP) consistently assign the same server to both users.

To further explore the network infrastructure of the five social VR platforms, we conduct more experiments in the western U.S. (Los Angeles) and Europe (United Kingdom). Note that we cannot experiment with Worlds in Europe as it is currently available in only the U.S. and Canada [71]. Our findings are as follows.

- AltspaceVR's servers for control channels still rely on anycast with <5ms RTT from the testing locations. However, the servers for data channels are always located in the western U.S. with ~150ms RTT from the testing site in Europe.
- Hubs has HTTPS servers in the western U.S. and Europe with <5ms RTT from the testing locations. However, its WebRTC server always resides in the western U.S., with ~140ms RTT from the testing site in Europe.
- Rec Room, VRChat, and Worlds still assign us nearby servers or anycast servers with <5ms RTT from the testing sites.

Implications to the Metaverse 1 : The future Metaverse is envisioned to connect users all around the world, like the current online social media. However, our measurement study reveals that some of the social VR platforms are not well-provisioned yet and could not allocate servers close to end-users. While we believe the separation of control and data channels and their corresponding servers is a proper design principle, it may give rise to the synchronization problem among different types of servers, especially when serving a large number of geo-distributed users.

5 Throughput Analysis

In this section, we measure the network throughput of social VR platforms with two test users. We will investigate the impact of a larger user base in §6.

5.1 Measurement Results

We first measure the throughput of control channels over HTTPS connections. It is 30–200 Kbps for the uplink and 100–4,500 Kbps for the downlink of these platforms. Note that the data transfers are bursty, and the amount of exchanged data is small, 5–20 KB for uplink and 15–600 KB for downlink.

Table 3 summarizes the throughput of data channels when two users walk around and chat with each other (basic features) on these social VR platforms via Quest 2[2]. Our measurement results lead to the following key observations.

Platform	Tput (Kbps)		Resolution	Avatar (Kbps)
	Up	Down		
VRChat	31.4/2.6	31.3/3.3	1440×1584	24.7/1.5
AltspaceVR	41.3/2.1	40.4/3.2	2016×2224	11.1/1.2
Rec Room	41.7/3.8	41.5/3.0	1224×1346	35.2/4.1
Hubs	83.3/5.6	83.1/6.4	1216×1344	77.4/7.7
Worlds	752/12	413/8.3	1440×1584	332/7.5

Table 3: Overall throughput, content resolution, and throughput of data related to avatar embodiment of five platforms (two users with Quest 2). The two numbers for throughput are the average result and standard deviation.

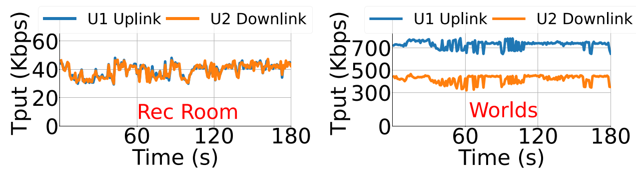


Figure 3: U1's uplink and U2's downlink throughput in Kbps for Rec Room (left) and Worlds (right).

- The throughput of data channels for most platforms is low, less than 100 Kbps in both uplink and downlink. Worlds has the highest throughput, ~750 Kbps for uplink and ~410 Kbps for downlink, more than 10× higher than AltspaceVR, Rec Room, and VRChat, which require merely ~30–40 Kbps network bandwidth. In contrast, even after various optimizations, the throughput of gaming-based networked VR systems such as Furion [43], which leverage remote rendering, could still be higher than 120 Mbps.
- The uplink throughput is almost identical to that of the downlink for all platforms, except for Worlds. As shown in Figure 3, we further find the instantaneous uplink throughput or its trend (only for Worlds) of U1 largely matches the downlink throughput or its trend of U2, and vice versa. This suggests that the servers of these platforms may just simply forward (part of) the data uploaded by one user to others without much processing, implying that these platforms do not employ remote rendering, which will be further verified by our subsequent scalability experiments (§6). Furthermore, U2's downlink bandwidth is noticeably lower than U1's uplink bandwidth on Worlds, which may indicate its servers perform some processing (*e.g.*, data compression) on uploaded data before forwarding it. However, there might be other reasons, for example, only part of the data uploaded by the user needs to be forwarded to others, while the servers of Worlds keep the remainder (*e.g.*, status reports from clients). Since all the data is encrypted, we do not know exactly what operations the servers perform.
- We observe that a social VR platform's throughput is independent of its content resolution (listed in Table 3). For example, the throughput of AltspaceVR, which owns the highest resolution (2016×2224), is comparable to that of Rec Room, whose resolution is much lower

[2] We do not observe significant throughput differences when using other devices such as HTC VIVE headsets and PCs to access these platforms, except for Worlds, which currently supports only Oculus VR headsets.

(1224×1346). This also indicates that all platforms adopt local rendering. The reason is that with remote rendering, the rendered frames will be encoded and delivered as video streams, whose throughput should depend on the resolution of the frames (*i.e.,* displayed content) and should be much higher than what we have observed (*e.g.,* >10 Mbps for a 1080p video at 60 FPS [37]).

• The throughput of these platforms does not rely on the location of the displayed avatars in the virtual environment and their distance to the user. A common optimization in computer graphics is to reduce the level of details of a 3D model when it is farther away from viewers or when the content is not in the focal area of their eyes [62]. While the current low throughput does not justify such optimizations, the full-fledged Metaverse in the future will likely require these optimizations to reduce the rendering overhead.

5.2 What is Being Delivered and Why is the Throughput Low?

Next, we explore why the throughput of these social VR platforms is low by dissecting the content that they deliver. In our analysis, we distinguish between virtual background and data related to avatar embodiment and motion.

Virtual Background. All five platforms offer only *static* virtual backgrounds. As such, in theory, the background needs to be downloaded only once, either during the installation of the application or in the initialization stage. On the other hand, when users interact with each other, the locations and behaviors of their embodied avatars are dynamic and thus should be updated in real-time. While the static background reduces the rendering burden and communication overhead, it limits users' interaction with the virtual environment, which would be required in the Metaverse [47, 75].

These platforms have different ways of downloading the virtual background. AltspaceVR and VRChat download 10–30 MB of data in the initialization stage. For Rec Room, we do not observe any large data transfer when users launch the application, even for the first time. The reason may be that Rec Room pre-downloads the virtual background during the installation of the application, which is indicated by the large size of its app in the Oculus store (1.41 GB), much larger than that of AltspaceVR (541 MB) and VRChat (793 MB). Worlds downloads only ~5 MB of data during the "Preparing for Visitors" phase, which is shown on the screen, every time users launch the application. Since the rendering tasks of other previous phases are performed without significant data traffic, we infer that Worlds also downloads at least some of the virtual background in advance, which is validated by the large size of its app in the Oculus store (1.13 GB).

Users have to download ~20 MB of data *each time* they join the Hubs platform. Since Hubs is browser-based and does not have an application installed on the device, this means that it does not cache the virtual background. We verify this by checking its cached files on Quest 2. It is likely a bug in the implementation of Hubs, and we have communicated this issue with Mozilla.

Avatar Embodiment and Motion. We next examine the throughput of data related to the avatar's appearance. Given that all platforms render the static virtual background which does not lead to a high bandwidth requirement, most of the continuously exchanged data should be contributed by avatars.

Figure 4: Avatars on five platforms. From left to right: Worlds, VRChat, Rec Room, AltspaceVR, and Hubs.

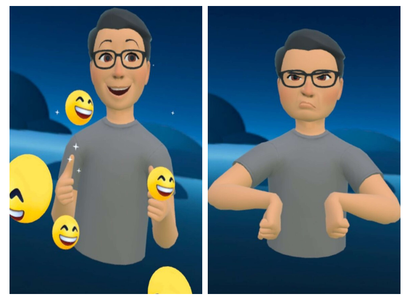
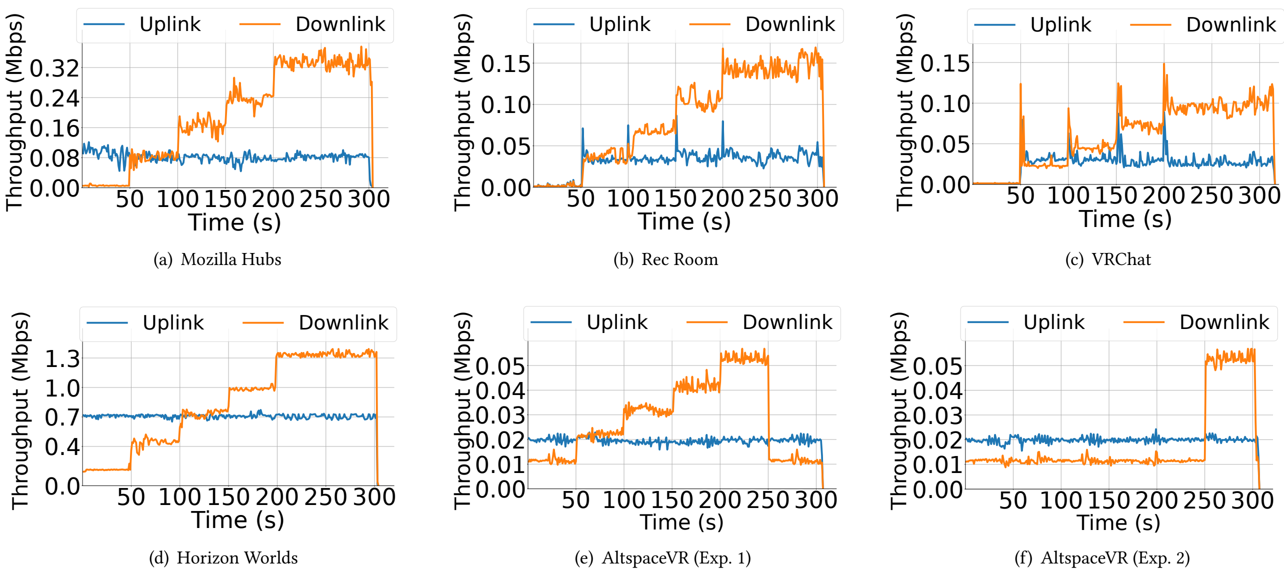


Figure 5: Facial expressions of avatars on Worlds when the user performs thumbs-up and thumbs-down.

We separate the throughput for delivering the dynamics of embodied avatars from the total throughput. Our approach is to first let U1 join a platform mutely (to exclude voice data) and measure the downlink throughput T. We then let U2 join the same platform mutely and measure the downlink throughput T' of U1 again. The difference between T and T' should be roughly the throughput for delivering U2's avatar embodiment and motion to U1. As shown in the last column of Table 3, the throughput of avatar data does account for a large portion of the total throughput for all platforms. The throughput for exchanging avatar data on Worlds is still much higher than other platforms (>300 Kbps vs. <100 Kbps).

To further demonstrate that the throughput difference of these social VR platforms is indeed largely caused by avatar data, we compare the embodied avatars of these platforms. As shown in Figure 4, the embodiment of avatars does vary across these platforms. For example, the avatar of AltspaceVR has no arms and no facial expressions, resulting in the lowest throughput (~10 Kbps). Although Rec Room's avatar also does not have arms, it has some simple facial expressions, such as laughing and sadness, which explains why its throughput (30–40 Kbps) is higher than AltspaceVR. The avatar of Hubs is similar to AltspaceVR in that it has no arms and lacks facial expressions. One possible reason that Hubs has a higher throughput for its avatars is that it uses HTTPS, instead of UDP, to transmit avatar data (§4.1), which introduces both protocol and encryption overhead.

The avatar of Worlds looks quite different from those of other platforms and presents some unique features. First, only the avatar of Worlds is human-like, while all other platforms' avatars are cartoon-shaped 3D models. Second, only Worlds updates avatars' facial expressions via hand gesture recognition by tracking users' hand motions through the headset's controllers. For example, Figure 5 shows the reactions of avatars on Worlds when the user performs thumbs-up and thumbs-down gestures. This explains why the avatar of Worlds leads to higher throughput than others



(a) Mozilla Hubs (b) Rec Room (c) VRChat

(d) Horizon Worlds (e) AltspaceVR (Exp. 1) (f) AltspaceVR (Exp. 2)

Figure 6: Throughput for U1 on five platforms. U2, U3, U4, and U5 join the same platform at 50, 100, 150, and 200s, respectively.

because when the avatar moves, the 3D coordinates of these tracked body parts should be delivered to others for updating their display.

Note that the avatars of most platforms, except VRChat, have only the upper torso, due to the lack of capture devices for modeling the lower limbs [3]. Moreover, the avatar's motion is not driven by the user's actual movement. It is solely determined by how users operate the hand-held controllers of the VR headset. Thus, high-quality full-body avatars may lead to much higher throughput than what we have observed on these social VR platforms. For example, existing work such as Holoportation [74] demonstrates that the bandwidth consumption for creating a photo-realistic 3D model of the human body and capturing its movement in real-time would be higher than 1 Gbps for even a single person.

Implications to the Metaverse 2 : Previous work has pointed out that avatar appearance on social VR platforms can affect the immersive experience of users, including their sense of identity, presence, co-presence, and interaction with others [28, 34, 42, 45]. In short, the more the avatar resembles a real human, the better the user experience would be. Overall, the visual quality of embodied avatars on present social VR platforms is far from satisfactory. We envision that the full-fledged Metaverse should offer a much better avatar embodiment, for example, by recreating the full-body motion via kinematics [20, 101], to provide a truly immersive experience. However, doing this will significantly increase the bandwidth requirement (*e.g.,* 10× throughput of Worlds with better avatar embodiment *vs.* others in the current practice).

Takeaways 1 : We summarize our key findings of the measurement of network throughput as follows.

• The throughput of all investigated social VR platforms (with two users) is lower than 1 Mbps, and it does not depend on the resolution of displayed content, device type, and the location of avatars (which determines their distance to viewers).

• All platforms support only static virtual background that has limited contribution to their throughput, and they download the virtual background in different ways.

• Avatar embodiment and motion account for a major portion of the throughput of these platforms, with the complexity of embodied avatars as the dominating factor. The avatar of Worlds is the most refined, requiring much higher bandwidth than other platforms.

6 Scalability Analysis

After investigating the throughput for two users on social VR platforms, we study the scalability by measuring the throughput and resource utilization when serving more users.

6.1 Controlled Experiments

We first conduct experiments in a controlled laboratory environment with five headset users (U1–U5). After U1 joins a social VR platform, we let four other users (U2–U5) join at 50, 100, 150, and 200s, respectively. U1 stands at the center of the virtual space and can see the avatars of all other users once they join for the first 250s. After that, U1 turns around for 180°, which makes the avatars of other users disappear in the viewport. We measure how the throughput of downlink and uplink changes as more users join and eventually leave U1's viewport. To avoid the interference of voice data on the experimental results, we ensure all users join mutely.

Throughput Scalability Issue. Figure 6 shows the instantaneous throughput of the five social VR platforms during the 300-second experiments. With the addition of new users, all platforms' downlink throughput increases almost *linearly.* This suggests a potential scalability issue: as more users join these social VR platforms, the servers simply forward the avatar data of each user to others, without further processing or optimizations such as aggregation. As expected, the uplink throughput of each user is unaffected by the presence of more avatars.

Viewport-adaptive Optimization. In the context of social VR, this optimization consists of sending updates for only the avatars contained within a user's viewport. This technique has been extensively explored in immersive video streaming [32, 33, 78] for reducing communication and rendering overhead. Figure 6 shows that when U2–U5 fall out of U1's viewport at 250s, the throughput of only AltspaceVR (Figure 6(e)) is reduced as a consequence of some form of viewport-adaptive optimization.

We conduct more experiments to better understand the adoption of viewport-adaptive optimization. In these experiments, U1 faces the corner of the virtual space during the first 250s and turns towards the center after that. When other avatars join one by one (again with a 50-second interval), they gather at the center of the virtual space and are thus not visible to U1 for the first 250s. We find that the downlink throughput of only AltspaceVR is dynamic, confirming the usage of viewport-adaptive optimization. Figure 6(f) demonstrates that, for the first 250s, the AltspaceVR server does not forward avatar data of U2–U5 to U1, as they are not visible to U1, leading to a significant throughput reduction.

To investigate the viewport-adaptive optimization of AltspaceVR, we conduct the following experiment with two users (U1 and U2). We first let U1's avatar turn its back to U2's avatar, and then gradually change the viewing direction of U1's avatar. In AltspaceVR, users can turn around their avatars with the controller of VR headsets. Avatars will turn 360° after users operate the controller 16 times in the same direction (*i.e.,* each operation will change the viewing direction by $360/16 = 22.5°$). Thus, it is feasible to detect the range of U1's "viewport" that is used by the server to determine whether U2's avatar is visible or not based on the change in U1's downlink throughput. Note that this viewport is usually larger than the actual field of view (FoV) of the VR headset to compensate for the prediction error of users' future viewport. Our experimental results show that the width of AltspaceVR's viewport to determine which avatar is visible and thus what content should be delivered is ~150°. As a result, it can, in theory, save up to ~58% (*i.e.,* $1 - 150/360$) of data consumption.

A key requirement of viewport-adaptive optimization is that the server should predict the future viewport of users when determining the data of which avatars should be forwarded, as the data delivery may take time. Suppose the data transmission takes t seconds. At time T, the server needs to predict users' viewport at $T + t$. When the prediction is not accurate, this optimization may lead to missing content, negatively affecting the user experience [33, 78].

6.2 Measurement in the Wild

Besides the controlled experiments, we conduct measurement studies in the wild through various public events that are available to users all over the world. Social VR platforms currently set an upper limit on the number of concurrent users per event, possibly due to the scalability issue. Among them, Worlds supports the smallest number of users, recommending 8–12 users in an event [73]. When attending public events, we find its actual cap is 16 users. Therefore, we use Quest 2 to attend public events with 7 to 15 users on these platforms. Since we have no control over other users' access, we do not know what devices they use to join the events. However, this does not affect our measurement results because the throughput is almost the same when using different types of devices to access these platforms (§5.1).

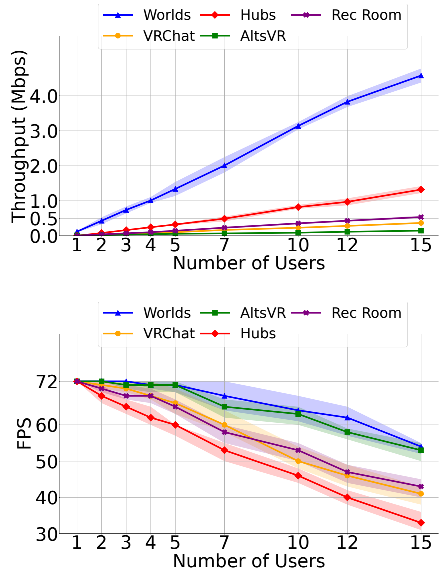


Figure 7: Average downlink throughput (top) and FPS (bottom) of five platforms with different numbers of users. The bands represent 95% confidence intervals.

Throughput and FPS. Figure 7 illustrates the average downlink throughput and FPS on the five platforms as the number of users increases. We add the results for 1–5 users from our controlled experiments in §6.1 (*i.e.,* private events) for comparison. As in private events, these platforms' throughput in public events grows linearly with the number of users. Since the avatar of Worlds is the most complicated (§5.2), its downlink throughput exceeds 4.5 Mbps when there are 15 users in the same event. Thus, to host a public event with 100 users, the resulting downlink throughput would be ~30 Mbps, already higher than the 25 Mbps downlink throughput defined by the U.S. Federal Communications Commission (FCC) for the standard broadband service [52].

When the number of users climbs, FPS starts to decrease for all platforms. Among them, Worlds performs the best with the smallest FPS drop (*e.g.,* 25% from 1 to 15 users), although its avatar is far more complex than others. In contrast, the FPS of Hubs drops from 72 to 60 (~17% decrease) when there are 5 users, and to only 33 (~54% drop) when the number of users increases to 15. This significant decrease in FPS will drastically affect the quality of user experience during social interactions, as the full-fledged Metaverse will be expected to host much more users.

Resource Utilization. Figure 8 shows how the average CPU and GPU utilization on Quest 2 for the five platforms changes with an increasing number of users. Among them, since Hubs runs on the browser, its CPU utilization is the highest and is close to 100% when there are 15 users. We find that when the number of users increases, AltspaceVR prefers to use the GPU to handle the additional processing overhead, while other platforms tend to use more of the CPU. For example, when the number of users grows from 1 to 15, the CPU utilization of AltspaceVR grows by only 15%,

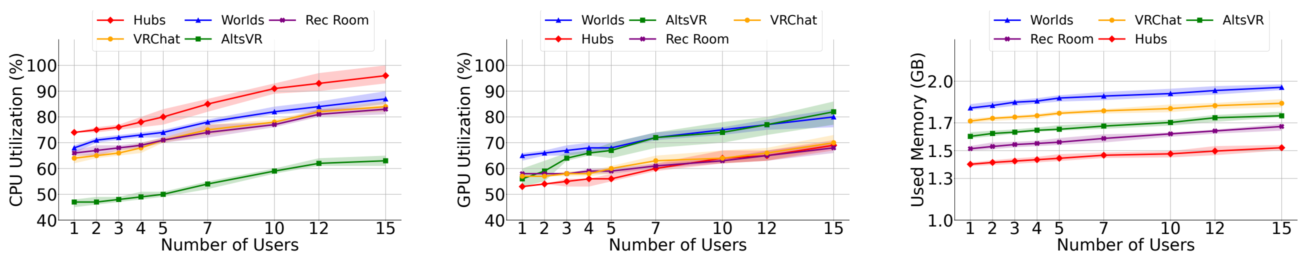


Figure 8: Average CPU (left) and GPU (middle) utilization and memory footprint (right) on Quest 2 for five platforms, with different numbers of users. The bands represent 95% confidence intervals.

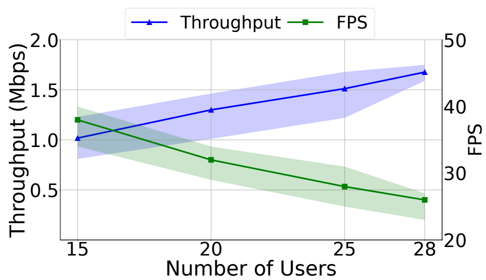


Figure 9: Average downlink throughput and FPS with different numbers of users in the large-scale experiments on Hubs. The bands represent 95% confidence intervals.

while the GPU utilization increases by 25%. In contrast, the CPU utilization of all other platforms grows by about 20%, and the GPU utilization increases by only 10-15%.

Memory and Energy Consumption. We next measure the memory footprint and energy consumption on Quest 2 for these platforms. The increase in the number of users has a limited effect on both memory and energy consumption. Regardless of the number of users (from 1 to 15), all platforms consume <10% of a fully charged Quest 2's battery after running the experiments for 10 minutes. Although the memory footprint does grow as the number of users increases, only <150 MB of extra memory is used when adding 14 more users, as shown in Figure 8. As a result, each avatar consumes only a small portion of memory space (~10 MB). Among them, Worlds consumes the most memory, for example, ~2 GB memory when there are 15 users, which is about 33% of the total memory of Quest 2 (~6 GB).

A Large-scale Experiment on Hubs. Given that the number of users is limited to 15 for the above public events, we also hosted a large-scale event on our private Hubs server, with up to 28 users. During this event, users were free to walk and talk with others. We used a fully-charged Quest 2 to join the event in our controlled laboratory environment. Figure 9 shows the average downlink throughput and FPS measured on this device, with different numbers of users. We observe that as the number of users increases to up to 28, the throughput of Hubs keeps increasing linearly, and its FPS decreases ~32% (from 15 to 28 users).

Implications to the Metaverse 3 : We have indicated the potential scalability issues of social VR platforms when there are up to 28 concurrent users. The future Metaverse may host thousands of millions of users simultaneously in the shared, interoperable virtual world. Viewport-adaptive optimizations can alleviate the scalability issue only to some extent. When there is a large number of avatars visible in the viewport, the required network bandwidth to deliver their data and the on-device computation resources for rendering may still be extremely high. Utilizing peer-to-peer (P2P) communication may be a potential direction. Our measurements indicate that currently none of these platforms' clients directly communicate with each other. Although Hubs employs WebRTC to deliver voice data (§4.2), which can utilize P2P [51], based on our measurements and the official documentation of Hubs [68], a central server is still used to forward data between users. If social VR platforms adopt P2P, user devices will aggregate the content received from multiple peers and render the virtual scene accordingly, alleviating the server workload. However, even with P2P, the scalability issues of throughput and on-device computation will remain. One further optimization is to reduce the frequency of updating data for avatars that the user is not interacting with [8]. Next, we discuss another potential solution, remote rendering.

6.3 Remote Rendering – A Potential Solution

A promising scheme to address the scalability issues of social VR platforms is to leverage remote rendering, for which the downlink throughput and on-device computation overhead are determined by the visual quality of encoded video streams, instead of the number of concurrent users.

We first summarize the evidence we identify in this paper that indicates these social VR platforms currently still use local rendering, other than remote rendering.

- All five platforms do not deliver video streams (with WebRTC technology), which is widely employed by remote rendering in commercial cloud gaming platforms (§4.1).
- The downlink throughput (two users) is below 100 Kbps for most platforms, which is much lower than the throughput of video streaming. Moreover, the throughput is independent of the resolution of displayed content (§5.1).
- As the number of users grows, the downlink throughput increases linearly for all platforms (§6).
- When delivering the action from one user to another, the receiver-side processing latency is at least 10ms higher than that on the sender-side, and it is higher than the server processing time for most platforms (§7).

Based on the above evidence, we can conclude, with high confidence, that all these social VR platforms have not benefited from remote rendering yet and utilized local rendering instead. This is caused by the design decision that the platform servers directly forward avatar data among all users for real-time updates (the main reason for scalability issues). On the other hand, when the server is responsible for rendering the content, even if there are a large number of concurrent users (especially when their avatars are clustered together), the servers will render the entire scene in the viewport of a user, including only visible avatars, into a 2D video frame. Hence, the amount of resulting data is independent of the number of users, mitigating the scalability issues.

We emphasize that remote rendering is not a silver bullet and has its own technical challenges. For example, similar to viewport-adaptive optimizations, the server should predict users' future viewport as the rendering of the content and the delivery over the network to VR headsets take time. Moreover, the server needs to render the same number of scenes as the number of users, since different users may have different viewports. Fortunately, these issues have been extensively investigated in cloud-based VR gaming and immersive video streaming [11, 32, 43, 61, 78], which makes remote rendering a promising solution for scaling the Metaverse.

Note that our prior work has identified the throughput scalability issue of Horizon Workrooms, a social VR platform (from Meta) for hosting online meetings [14]. This suggests that scalability is indeed a common problem faced by today's social VR platforms. Besides finding more evidence on other platforms, in this paper we identify its root cause, discover the viewport-adaptive optimization that is adopted by AltspaceVR, measure the scalability of end-to-end latency (§7) and on-device computation resource utilization, and point out possible solutions to address this potential barrier of building the Metaverse.

Takeaways (2) : We have the following key findings from our scalability experiments in the controlled laboratory environment and in the wild.

- All five social VR platforms face potential scalability issues, and their throughput increases almost linearly with the number of users.
- Only AltspaceVR adopts the viewport-adaptive optimization, while others blindly exchange avatar data, regardless of whether or not the avatars are visible to other users.
- For all platforms, the on-device resource utilization increases and the FPS degrades as the number of users grows. In this case, AltspaceVR tends to prioritize the GPU to handle the increased processing load, while other platforms tend to use more of the CPU.
- We suggest remote rendering as a promising solution to alleviate the identified scalability issues.

7 End-to-end Latency

In this section, we measure the end-to-end latency when users interact with each other on social VR platforms. It is defined as the time that is taken from when one user performs an action to when that action is displayed on others' screens.

We design the following experiment to measure the end-to-end latency between two users U1 and U2. We first let the index fingers of U1 and U2 touch together. Then U1 quickly moves them away from those of U2. During the experiment, we record the screens of U1 and U2 at the running FPS with the following command, "`adb shell setprop debug.oculus.fullRateCapture 1`". We use ffmpeg [25] to extract the frames from the recorded videos and get their timestamps to calculate the end-to-end latency. We show the last frame before displaying this action on U1's screen and the first frame to reflect it on U2's screen in Figure 10.

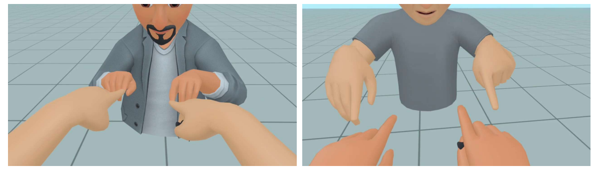


Figure 10: The extracted video frames that reflect an action on its sender U1's screen (left) and receiver U2's screen (right) for calculating the end-to-end latency.

Platform	E2E	Sender	Receiver	Server
Rec Room	101.7/8.7	25.9/8.6	39.9/7.8	29.9/6.4
VRChat	104.3/9.3	27.3/6.2	37.4/6.4	33.5/9.5
Worlds	128.5/11	26.2/4.5	49.1/9.1	40.2/11
AltspaceVR	209.2/13	24.5/5.2	36.1/9.9	68.6/12
Hubs	239.1/7.3	42.4/6.3	60.1/6.5	52.2/7.7
Hubs∗	130.7/6.3	40.3/5.2	61.5/5.7	16.2/2.4

Table 4: The end-to-end (E2E) latency in ms and the latency for the sender, receiver, and server processing. The two numbers in each cell are the average result and standard deviation. Hubs∗ refers to our private Hubs server.

Although similar methods have been utilized to measure the end-to-end latency of cloud gaming platforms [37], a key challenge here is to synchronize Quest 2 VR headsets. Widely used time synchronization protocols such as NTP [55] are not supported on Quest 2 without root access. While Quest 2 runs the Android OS, the method to root it has not been well developed yet. Therefore, we resort to an alternative and more generic approach. We first connect to Quest 2 from the WiFi AP via the Android Debug Bridge (ADB) tool. Then we use the "`adb shell echo $EPOCHREALTIME`" command on the WiFi AP to get the system time of Quest 2, and the system call to get the clock of WiFi AP at the same time. Furthermore, we measure the RTT between the WiFi AP and Quest 2. With these results, we can know the time difference between Quest 2 and the WiFi AP and synchronize them at the millisecond level. By synchronizing both Quest 2 headsets with the WiFi AP, we can calculate the end-to-end latency based on the timestamps of the two extracted video frames in Figure 10.

We report the measurement results in Table 4. Among the five platforms, Hubs has the highest end-to-end latency (~240ms), followed by AltspaceVR (~210ms), both higher than the 150ms target threshold of an immersive collaborative environment [81]. The latency of Rec Room and VRChat is around 100ms. Note that both users are located on the U.S. east coast. The end-to-end latency will be higher when they are at different farther-away places, as the network delay will grow in this case.

To better understand the variance of the end-to-end latency of these platforms, we break it down by measuring the latency between the Quest 2 headsets, our WiFi AP, and platform servers. We also retrieve the timestamps of data packets that deliver the finger movement from the traces collected on our WiFi AP and Quest 2 to facilitate the breakdown into the sender, receiver, and

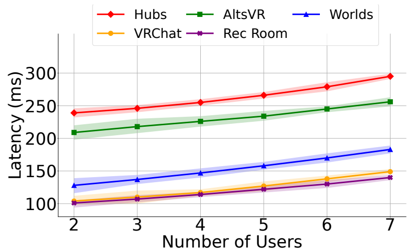


Figure 11: Average end-to-end latency for five platforms, with different numbers of users. The bands represent 95% confidence intervals.

server processing latency, which is feasible due to the low data rate and thus sparse data transfers of these platforms.

The high end-to-end latency of AltspaceVR and Hubs is partially contributed by the high RTT (>70ms) between our WiFi AP and the platform servers. Recall that while we conduct the experiments on the U.S. east coast, both platforms assign servers on the U.S. west coast to forward avatar data (§4.2). Another reason for the high latency of Hubs is that it is a Web-based application, which introduces extra computation overhead, as shown in Figure 8 (*i.e.,* the highest CPU utilization). Thus, it has the highest sender and receiver processing latency of these platforms.

Table 4 reveals that AltspaceVR has the highest server processing latency among all platforms. We infer that it may be induced by the viewport-adaptive optimization adopted by AltspaceVR (§6), as viewport prediction may involve complicated machine learning algorithms [108]. Hubs has the highest server processing latency among the platforms that do not have the viewport-adaptive optimization (>50ms vs. ≤40ms). Since it open-sources its code, we have the opportunity to set up our own Hubs server on an AWS `t3.medium` instance to measure the end-to-end latency again. With this server, the latency drops to only ~16ms (~70% reduction), which indicates that the Hubs server may not be well provisioned to handle the workload.

In general, the processing latency on the receiver is much higher than that on the sender and is even higher than the server-processing latency (except for AltspaceVR that leverages viewport-adaptive optimizations), which is another indication of local rendering. In contrast, the server-processing latency of commercial cloud gaming platforms that utilize remote rendering ranges from 43 to 140ms [37]. We also find that Worlds has higher receiver-processing latency than other non-Web-based platforms (close to 50ms vs. <40ms), which is likely due to its more realistic avatar embodiment and thus higher rendering overhead.

After examining the end-to-end latency of two users, we study whether it will change with an increasing number of users (*i.e.,* the scalability). With the above method, we measure the latency between U1 and U2 when up to five more users join the same social VR platform. Figure 11 demonstrates that the end-to-end latency also faces potential scalability issues. For example, with seven users in an event, the latency grows from 239.1 (two users) to 295.4ms for Hubs, from 128.5 to 181.4ms for Worlds, and from 101.7 to 140.3ms for Rec Room, respectively. Figure 11 also indicates that the difference in latency when adding one more user grows. For example, when increasing the number of users from 2 to 3, the delta of latency for Hubs is 7ms. It grows to 9, 11, 13, and 16ms for 4–7 users. One possible reason is that the server-side queuing delay increases when serving more users.

It becomes difficult to break down the end-to-end latency when there are more users. The reason is that the increase of data packets reduces the packet interval, making it challenging to determine which packet carries a specific action performed by users. However, given the increase in computation resource utilization on VR headsets shown in Figure 8, we can infer that the growth of end-to-end latency is mainly caused by the receiver-side processing. Remote rendering is still effective to handle the latency scalability issue, as the decoding and display time of video frames mostly depends on their visual quality, not the number of avatars in the scene.

8 Network Disruptions

In this section, we explore how social VR platforms respond to dynamic network conditions for the two-user scenario (U1 and U2), as in practice, network fluctuations are common. We use `tc-netem` [40] to constrain the uplink and downlink of U1, in terms of throughput, latency, and packet loss rate. Our selected parameters are as follows:

- Uplink Bandwidth: 1.5, 1.2, 1, 0.7, 0.5, and 0.3 Mbps.
- Downlink Bandwidth: 1, 0.7, 0.5, 0.3, 0.2, and 0.1 Mbps.
- Uplink/Downlink Latency: 50, 100, 200, 300, 400, and 500ms.
- Uplink/Downlink Packet Loss: 1, 3, 5, 7, 10, and 20%.

Each restricted condition lasts for 40s. The network then goes back to normal for another 60s, leading to a duration of 300s for the entire experiment.

8.1 Throughput Disruption

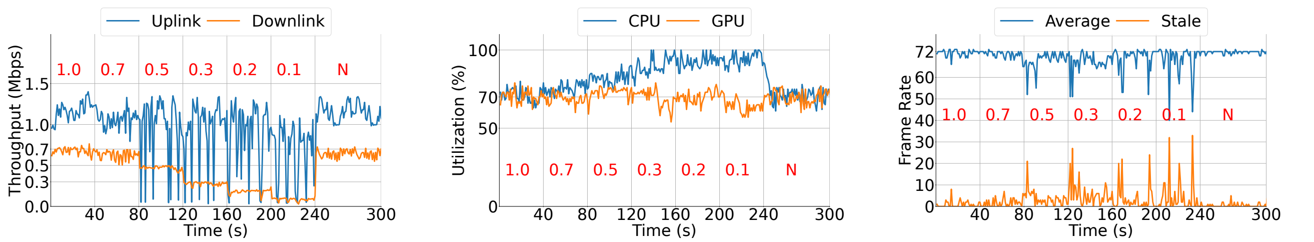
Since the throughput of all platforms except Worlds is <100 Kbps (§5.1), we conduct throughput disruption experiments on only Worlds. We focus our study on the impact of throughput disruption on social events, for example, when users play together a game, which is more interactive and bandwidth-intensive. Also, it is considered one of the key use cases for social VR [31]. We select the Arena Clash [100] shooting game on Worlds for our experiments.

We first conduct downlink throughput disruption experiments for U1 on Worlds. Figure 12(a) shows the resulting uplink and downlink throughput. For this shooting game, the throughput of Worlds increases to ~0.7/1.2 Mbps (downlink/uplink)[3]. When we limit the downlink bandwidth to 0.5 Mbps, Worlds exhibits an "aggressive" state, using all available bandwidth as much as possible. Moreover, when the downlink capacity is insufficient, the unrestricted uplink starts to fluctuate violently and consequently affects U2's downlink throughput (not shown for clarity)[4]. The reason may be that Worlds prioritizes the process of missing critical information that it does not receive when the downlink bandwidth is insufficient. At this point, it does not have enough computation resources to handle the to-be-uploaded data, resulting in a fluctuating uplink throughput.

This conjecture is supported by the increased CPU utilization shown in Figure 12(b) and the decreased FPS shown in Figure 12(c). For example, Figure 12(b) demonstrates that the CPU utilization

[3]The throughput of shooting games on Rec Room and VRChat is still lower than 100 Kbps (*e.g.,* 75 Kbps for Laser Tag [80] on Rec Room and 40 Kbps for Voxel Shooting [106] on VRChat).

[4]The majority of U2's downlink is contributed by U1's uplink (Figure 3).



(a) The uplink and downlink throughput for U1.

(b) The CPU/GPU utilization for U1.

(c) The average and stale frame count for U1.

Figure 12: Downlink throughput disruption of U1 in a shooting game of Worlds. The numbers in red indicate the stages of disruption (throughput in Mbps). "N" means no disruption.

does start to increase after 40s when the downlink is limited to 0.7 Mbps and sometimes reaches 100% when the downlink bandwidth is even lower. The GPU utilization, on the other hand, slightly drops when the downlink bandwidth is limited. One possible reason is that some stale frames are re-used at this time (as shown in Figure 12(c)), reducing the rendering overhead.

We next conduct uplink throughput disruption experiments for U1 on Worlds. We plot the uplink throughput for both UDP and TCP (for HTTPS connections) and the downlink UDP throughput in Figure 13 (top). As with the experiments of downlink throughput, U1 uses all available uplink bandwidth as much as possible when it is limited. Moreover, we observe that the UDP downlink throughput of U1 starts to decrease after the uplink is constrained. The reason is that when U1 cannot upload enough data to U2, U2 needs to prioritize the processing of missing information. As we discussed above, this affects the uplink throughput of U2, and in turn, the downlink throughput of U1.

We find in Figure 13 (top) that there are drastic fluctuations in the uplink UDP throughput of U1 after 80s when the available bandwidth is lower than 1.2 Mbps. Moreover, the drop in uplink UDP throughput happens whenever there is a spike in uplink TCP throughput. This is, nevertheless, counter-intuitive, as usually UDP will win when competing with TCP due to TCP's congestion control. Therefore, we infer that the TCP data may carry some critical information for control channels (§4) that Worlds prioritizes.

To further explore the interaction between TCP uplink and UDP uplink, we conduct the following experiment. We control only the TCP uplink by applying an increasing delay of 5, 10, and 15s and a 100% packet loss rate, respectively. The experimental results in Figure 13 (bottom) show that when we increase the delay of TCP uplink, significant gaps appear in the data transfers of UDP uplink, and the duration of the gaps is about the same as the introduced delay on the TCP uplink. This implies that Worlds will not send data over UDP until the TCP packets have been successfully delivered. Moreover, when the packet loss of TCP uplink is 100% (starting from 180s), there are only tiny data exchanges over UDP for about 30s. During this period, U1's avatar can still freely walk in the game space, although U1 and U2 cannot see each other's avatar anymore. After that (at ~210s), the UDP connection is broken, and U1's screen is frozen. Even when we remove the applied packet loss on TCP uplink at 240s, the UDP connection is not restored (*i.e.,* the screen is still frozen), while the TCP connection can recover.

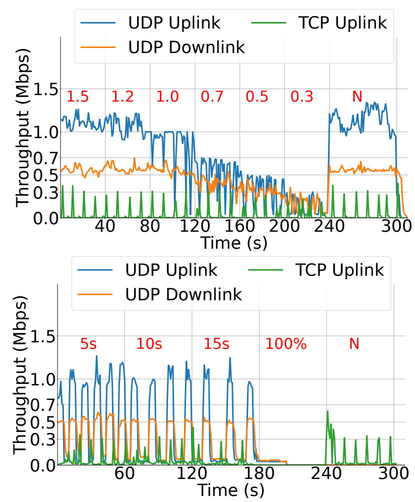


Figure 13: Uplink throughput disruption of U1 in a shooting game of Worlds. Top: Throttling the bandwidth for all uplink traffic. Bottom: Controlling only uplink TCP traffic (0–180s: increased latency; 180-240s: 100% packet loss rate). The numbers in red indicate the stages of disruption (throughput in Mbps, latency in seconds, or packet loss rate). "N" means no disruption.

The experimental results in Figure 13 indicate that there is indeed an interplay between TCP uplink and UDP uplink, and TCP traffic has a higher priority than UDP traffic (Implications 1 in §4). Moreover, we observe that when we add delay to TCP uplink, the countdown board in the game fails to update the remaining time of the game in real-time. This suggests that one role of the TCP connection is to synchronize the clock between users via the server.

8.2 Latency and Packet Loss Disruptions

We next investigate the impact of latency and packet loss disruptions on social VR platforms. When users just walk around and chat with each other, their experience will be affected only when the end-to-end latency is higher than 300ms (*e.g.,* adding ~200ms extra latency for Rec Room and VRChat, or ~100ms for AltspaceVR (Table 4)). However, higher latency significantly impacts the user experience for the gaming scenario. Table 1 shows that Hubs is currently the only platform which does not support gaming. AltspaceVR has only Q&A games without much interactivity (*i.e.,* questions are displayed on the screen for users to choose the answer). Thus, besides the Arena Clash [100] game on Worlds, we select two other shooting games, Laser Tag [80] on Rec Room and Voxel Shooting [106] on VRChat, to conduct latency disruption experiments. Recall that the end-to-end latency of these three platforms is lower than 130ms (Table 4). We find that 50ms of additional latency is sufficient to deteriorate users' gaming experience. All platforms are not sensitive to packet losses. Even when the packet loss rate reaches 20%, users do not perceive any disturbance. One of the reasons may be that the embodied avatars of these platforms are relatively rough (Figure 4 in §5.1). Even if some parts of the avatar are not updated in time, users may not be able to perceive the difference. Moreover, these platforms may compensate for the missing movement data of avatars through methods such as motion prediction.

Takeaways (3) : Through the network disruption experiments, we have two key findings. First, the drop of downlink bandwidth affects uplink data transfer of Worlds and the CPU/GPU utilization, indicating an interplay between networking and computation. Second, Worlds gives TCP uplink traffic higher priority than UDP uplink traffic, by blocking UDP until TCP packets have been properly delivered.

9 Discussion

While this is the first comprehensive study of social VR platforms, there are a few limitations of our work that we plan to address in the future.

Large-scale Crowd-sourced Experiments. We perform the measurement study with a combination of manual and automated experiments. Oculus officially provides a tool called AutoDriver [70] that enables the test of VR applications by automatically playing back pre-defined inputs. Nevertheless, we have conducted all experiments by ourselves at limited locations. We are currently building open-source tools on Oculus Quest 2 by further extending AutoDriver and OVR Metrics Tool in order to facilitate large-scale crowd-sourced experiments and better examine the performance of social VR platforms in the wild.

Security, Privacy, and Harassment. We have not yet explored the security [38, 97] and privacy [54, 88] issues of social VR platforms and how they handle online harassment [9, 27], which are all crucial problems that need to be addressed when building the Metaverse. Actually, there are an increasing number of reports regarding harassment on most of the platforms that we study in this paper [18, 19, 92, 93, 96]. As we list in Table 1, all five platforms, except for Hubs, have set up the personal boundary/bubble/space [10] that aims to protect users from harassment. In future work, we will investigate these mechanisms of social VR platforms and measure their effectiveness.

Other Types of Metaverse Platforms. Our measurement study focuses on social VR platforms, which can be viewed as a rudimentary prototype of the Metaverse. There are other kinds of platforms such as massively multiplayer online games (*e.g.,* Roblox [82], Fortnite [24], and Minecraft [65]) and blockchain or NFT-based online games (*e.g.,* Axie Infinity [56], Decentraland [21], and Upland [95]) that are considered as part of the Metaverse ecosystem [23]. We intend to conduct measurement studies on different types of platforms to gain a more inclusive understanding of the Metaverse.

10 Related Work

Social VR, by combining social media and VR technologies [59, 87], has attracted increasing attention from HCI and computer graphics communities. HCI researchers have been investigating a wide range of topics in social VR, including interaction with avatars [28, 34, 42], interpersonal relationships [26, 53, 89, 90], privacy and security concerns [54], *etc.* Moreover, the HCI community studied various aspects of organizing different events on social VR platforms, such as remote learning [29, 84], dancing [76], watching movies [48], and hosting conferences [103]. The computer graphics community focuses mainly on the appearance of avatars for social VR [44, 45]. In the network community, Zhang *et al.* [110] identified potential bottlenecks of mobile social VR and proposed a tentative system architecture. In our previous work, we conducted a preliminary measurement study of Horizon Workrooms, a social VR platform for online meetings [14].

Online Social Networks. There has been a wealth of measurement studies of online social networks, such as user behavior [7], information propagation [13], group structure [66], and user interactions [85, 104]. In addition to measurement studies, previous work investigated user privacy [5] and scalability [77] of online social networks. Different from the above work, we conduct an extensive measurement study of social VR, the next generation of online social media towards the Metaverse.

Networked VR. There is plenty of work on improving networked VR systems [1, 11, 43, 50, 61]. For example, Coterie splits the rendering of the background environment between mobile devices and servers to support high-quality multi-user VR [61]. Q-VR proposes a collaborative rendering solution based on software-hardware co-design for low-latency VR systems [107]. In contrast to the above work, we measure various aspects of existing social VR platforms.

11 Concluding Remarks

In this paper, we presented a first-of-its-kind in-depth, and systematic measurement study of popular social VR platforms, one of the key enablers of the Metaverse. Our work started with a detailed analysis of network protocols and infrastructure of these platforms, followed by thorough evaluations of their throughput. We then discovered the potential scalability issues, in terms of throughput, end-to-end latency, and on-device computation resource consumption, and identified the servers' direct forwarding of avatar data as the root cause. Moreover, we investigated how these platforms respond to dynamic network conditions such as throughput drops, increased latency, and packet losses. We hope our findings can shed light on the design practices to eventually realize the Metaverse.

Acknowledgment

We thank the anonymous reviewers and our shepherd Mark Crovella for their insightful comments. This work was supported in part by NSF CNS-2007153 and the Commonwealth Cyber Initiative, an investment in the advancement of cyber R&D, innovation, and workforce development. For more information about CCI, visit www.cyberinitiative.org.

References

[1] O. Abari, D. Bharadia, A. Duffield, and D. Katabi. Enabling High-Quality Untethered Virtual Reality. In *Proceedings of USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 2017.
[2] Advanced Network and Services (ANS). https://www.anscorporate.com/, 2022. [accessed on 19-September-2022].
[3] K. Ahuja, V. Shen, C. M. Fang, N. Riopelle, A. Kong, and C. Harrison. ControllerPose: Inside-Out Body Capture with VR Controller Cameras. In *Proceedings of ACM Conference on Human Factors in Computing Systems (CHI)*, 2022.
[4] Amazon. Amazon Web Services (AWS). https://aws.amazon.com/, 2022. [accessed on 19-September-2022].
[5] R. Baden, A. Bender, N. Spring, B. Bhattacharjee, and D. Starin. Persona: An Online Social Network with User-Defined Privacy. In *Proceedings of ACM SIGCOMM Conference on Data Communication (SIGCOMM)*, 2009.
[6] M. Barreda-Ángeles and T. Hartmann. Psychological Benefits of Using Social Virtual Reality Platforms During the Covid-19 Pandemic: The Role of Social and Spatial Presence. *Computers in Human Behavior*, 127:107047, 2022.
[7] F. Benevenuto, T. Rodrigues, M. Cha, and V. Almeida. Characterizing User Behavior in Online Social Networks. In *Proceedings of ACM SIGCOMM Conference on Internet Measurement (IMC)*, 2009.
[8] A. Bharambe, J. R. Douceur, J. R. Lorch, T. Moscibroda, J. Pang, S. Seshan, and X. Zhuang. Donnybrook: Enabling Large-Scale, High-Speed, Peer-to-Peer Games. In *Proceedings of ACM SIGCOMM Conference on Data Communication (SIGCOMM)*, 2008.
[9] L. Blackwell, N. Ellison, N. Elliott-Deflo, and R. Schwartz. Harassment in Social Virtual Reality: Challenges for Platform Governance. *Proceedings of the ACM on Human-Computer Interaction*, 3:1–25, 2019.
[10] A. Bönsch, S. Radke, H. Overath, L. M. Asché, J. Wendt, T. Vierjahn, U. Habel, and T. W. Kuhlen. Social VR: How Personal Space is Affected by Virtual Agents' Emotions. In *Proceedings of IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, 2018.
[11] K. Boos, D. Chu, and E. Cuervo. FlashBack: Immersive Virtual Reality on Mobile Devices via Rendering Memoization. In *Proceedings of ACM International Conference on Mobile Systems, Applications, and Services (MobiSys)*, 2016.
[12] M. Carrascosa and B. Bellalta. Cloud-gaming: Analysis of Google Stadia Traffic. *Computer Communications*, 188:99–116, 2022.
[13] M. Cha, A. Mislove, and K. P. Gummadi. A Measurement-driven Analysis of Information Propagation in the Flickr Social Network. In *Proceedings of International Conference on World Wide Web (WWW)*, 2009.
[14] R. Cheng, N. Wu, S. Chen, and B. Han. Reality Check of Metaverse: A First Look at Commercial Social Virtual Reality Platforms. In *Proceedings of IEEE Workshop for Building the Foundations of the Metaverse (Metabuild), co-located with IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, 2022.
[15] R. Cheng, N. Wu, S. Chen, and B. Han. Will Metaverse be NextG Internet? Vision, Hype, and Reality. *IEEE Network*, 2022 (to appear).
[16] B. A. Ciccone, S. K. Bailey, and J. E. Lewis. The Next Generation of Virtual Reality: Recommendations for Accessible and Ergonomic Design. *Ergonomics in Design*, 2021.
[17] Cloudflare. https://www.cloudflare.com/, 2022. [accessed on 19-September-2022].
[18] CNET. As Facebook Plans the Metaverse, It Struggles to Combat Harassment in VR. https://www.cnet.com/tech/gaming/features/as-facebook-plans-the-metaverse-it-struggles-to-combat-harassment-in-vr/, 2022. [accessed on 19-September-2022].
[19] CNN. Harassment is a problem in VR, and it's likely to get worse. https://www.cnn.com/2022/05/05/tech/virtual-reality-harassment/index.html, 2022. [accessed on 19-September-2022].
[20] H. Dai, A. Valenzuela, and R. Tedrake. Whole-body Motion Planning with Centroidal Dynamics and Full Kinematics. In *Proceedings of IEEE-RAS International Conference on Humanoid Robots*, 2014.
[21] Decentraland. Decentraland. https://decentraland.org/, 2022. [accessed on 19-September-2022].
[22] A. Di Domenico, G. Perna, M. Trevisan, L. Vassio, and D. Giordano. A Network Analysis on Cloud Gaming: Stadia, GeForce Now and PSNow. *MDPI Network*, 1(3):247–260, 2021.
[23] H. Duan, J. Li, S. Fan, Z. Lin, X. Wu, and W. Cai. Metaverse for Social Good: A University Campus Prototype. In *Proceedings of ACM International Conference on Multimedia (MM)*, 2021.
[24] Epic Games. Fortnite. https://www.epicgames.com/fortnite/en-US/home, 2022. [accessed on 19-September-2022].
[25] FFmpeg. https://ffmpeg.org/, 2022. [accessed on 19-September-2022].
[26] G. Freeman and D. Acena. Hugging from A Distance: Building Interpersonal Relationships in Social Virtual Reality. In *Proceedings of ACM International Conference on Interactive Media Experiences*, 2021.
[27] G. Freeman, S. Zamanifard, D. Maloney, and D. Acena. Disturbing the Peace: Experiencing and Mitigating Emerging Harassment in Social Virtual Reality. *Proceedings of the ACM on Human-Computer Interaction*, 6(CSCW1):1–30, 2022.
[28] G. Freeman, S. Zamanifard, D. Maloney, and A. Adkins. My Body, My Avatar: How People Perceive Their Avatars in Social Virtual Reality. In *Proceedings of ACM Conference on Human Factors in Computing Systems (CHI), Extended Abstracts*, 2020.
[29] F. Genz, N. Fuchs, D. Kolb, S. Müller, and D. Kranzlmüller. Evaluation of Proprietary Social VR Platforms for Use in Distance Learning. In *Proceedings of International Conference on Augmented Reality, Virtual Reality and Computer Graphics*, 2021.
[30] Google. Chrome WebRTC Debugging Console. chrome://webrtc-internals/, 2022. [accessed on 19-September-2022].
[31] S. Gunkel, H. Stokking, M. Prins, O. Niamut, E. Siahaan, and P. Cesar. Experiencing Virtual Reality Together: Social VR Use Case Study. In *Proceedings of ACM International Conference on Interactive Experiences for TV and Online Video*, 2018.
[32] B. Han, Y. Liu, and F. Qian. ViVo: Visibility-Aware Mobile Volumetric Video Streaming. In *Proceedings of ACM International Conference on Mobile Computing and Networking (MobiCom)*, 2020.
[33] J. He, M. A. Qureshi, L. Qiu, J. Li, F. Li, and L. Han. Rubiks: Practical 360° Streaming for Smartphones. In *Proceedings of ACM International Conference on Mobile Systems, Applications, and Services (MobiSys)*, 2018.
[34] P. Heidicker, E. Langbehn, and F. Steinicke. Influence of Avatar Appearance on Presence in Social VR. In *Proceedings of IEEE Symposium on 3D User Interfaces (3DUI)*, 2017.
[35] C. Holmberg, S. Hakansson, and G. Eriksson. Web Real-Time Communication Use Cases and Requirements. RFC 7478, 2015. [accessed on 19-September-2022].
[36] ipinfo.io. https://ipinfo.io/, 2022. [accessed on 19-September-2022].
[37] H. Iqbal, A. Khalid, and M. Shahzad. Dissecting Cloud Gaming Performance with DECAF. In *Proceedings of ACM on Measurement and Analysis of Computing Systems (SIGMETRICS)*, 2021.
[38] B. John, S. Jörg, S. Koppal, and E. Jain. The Security-Utility Trade-off for Iris Authentication and Eye Animation for Social Virtual Avatars. *IEEE Transactions on Visualization and Computer Graphics*, 26(5):1880–1890, 2020.
[39] M. Jonas, S. Said, D. Yu, C. Aiello, N. Furlo, and D. Zytko. Towards a Taxonomy of Social VR Application Design. In *Proceedings of the Annual Symposium on Computer-Human Interaction in Play Companion (CHI PLAY), Extended Abstracts*, 2019.
[40] M. Kerrisk. tc-netem(8) — Linux manual page. https://man7.org/linux/man-pages/man8/tc-netem.8.html, 2022. [accessed on 19-September-2022].
[41] B. Keshavarz and J. F. Golding. Motion Sickness: Current concepts and Management. *Current Opinion in Neurology*, 35(1):107–112, 2022.
[42] A. Kolesnichenko, J. McVeigh-Schultz, and K. Isbister. Understanding Emerging Design Practices for Avatar Systems in the Commercial Social VR Ecology. In *Proceedings of ACM Conference Companion Publication on Designing Interactive Systems (DIS)*, 2019.
[43] Z. Lai, Y. C. Hu, Y. Cui, L. Sun, and N. Dai. Furion: Engineering High-Quality Immersive Virtual Reality on Today's Mobile Devices. In *Proceedings of ACM International Conference on Mobile Computing and Networking (MobiCom)*, 2017.
[44] M. E. Latoschik, F. Kern, J.-P. Stauffert, A. Bartl, M. Botsch, and J.-L. Lugrin. Not Alone Here?! Scalability and User Experience of Embodied Ambient Crowds in Distributed Social Virtual Reality. *IEEE Transactions on Visualization and Computer Graphics*, 25(5):2134–2144, 2019.
[45] M. E. Latoschik, D. Roth, D. Gall, J. Achenbach, T. Waltemate, and M. Botsch. The Effect of Avatar Realism in Immersive Social Virtual Realities. In *Proceedings of ACM Symposium on Virtual Reality Software and Technology (VRST)*, 2017.
[46] L.-H. Lee, T. Braud, P. Zhou, L. Wang, D. Xu, Z. Lin, A. Kumar, C. Bermejo, and P. Hui. All One Needs to Know about Metaverse: A Complete Survey on Technological Singularity, Virtual Ecosystem, and Research Agenda. https://arxiv.org/abs/2110.05352, 2021.
[47] L.-H. Lee, Z. Lin, R. Hu, Z. Gong, A. Kumar, T. Li, S. Li, and P. Hui. When Creators Meet the Metaverse: A Survey on Computational Arts. https://arxiv.org/abs/2111.13486, 2021.
[48] J. Li, S. Subramanyam, J. Jansen, Y. Mei, I. Reimat, K. Ławicka, and P. Cesar. Evaluating the User Experience of a Photorealistic Social VR Movie. In *Proceedings of IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, 2021.
[49] K. E. Lindqvist and J. Abley. Operation of Anycast Services. RFC 4786. [accessed on 19-September-2022].
[50] L. Liu, R. Zhong, W. Zhang, Y. Liu, J. Zhang, L. Zhang, and M. Gruteser. 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)*, 2018.
[51] S. Loreto and S. P. Romano. *Real-Time Communication with WebRTC: Peer-to-Peer in the Browser.* " O'Reilly Media, Inc.", 2014.
[52] K. MacMillan, T. Mangla, J. Saxon, and N. Feamster. Measuring the Performance and Network Utilization of Popular Video Conferencing Applications. In *Proceedings of ACM SIGCOMM Conference on Internet Measurement (IMC)*, 2021.
[53] D. Maloney, G. Freeman, and D. Y. Wohn. "Talking without a Voice" Understanding Non-verbal Communication in Social Virtual Reality. *Proceedings of*

the ACM on Human-Computer Interaction, 4(CSCW2):1–25, 2020.
[54] D. Maloney, S. Zamanifard, and G. Freeman. Anonymity vs. Familiarity: Self-Disclosure and Privacy in Social Virtual Reality. In *Proceedings of ACM Symposium on Virtual Reality Software and Technology (VRST)*, 2020.
[55] J. Martin, J. Burbank, W. Kasch, and P. D. L. Mills. Network Time Protocol Version 4: Protocol and Algorithms Specification. RFC 5905, 2010. [accessed on 19-September-2022].
[56] S. Mavis. Axie Infinity. https://axieinfinity.com/, 2022. [accessed on 19-September-2022].
[57] MaxMind. https://www.maxmind.com/en/home, 2022. [accessed on 19-September-2022].
[58] J. McVeigh-Schultz, A. Kolesnichenko, and K. Isbister. Shaping Pro-Social Interaction in VR: An Emerging Design Framework. In *Proceedings of ACM Conference on Human Factors in Computing Systems (CHI)*, 2019.
[59] J. McVeigh-Schultz, E. M. Segura, N. Merrill, and K. Isbister. What's It Mean to "Be Social" in VR?: Mapping the Social VR Design Ecology. In *Proceedings of ACM Conference Companion Publication on Designing Interactive Systems*, 2018.
[60] T. Mendez, W. Milliken, and D. C. Partridge. Host Anycasting Service. RFC 1546. [accessed on 19-September-2022].
[61] J. Meng, S. Paul, and Y. C. Hu. Coterie: Exploiting Frame Similarity to Enable High-Quality Multiplayer VR on Commodity Mobile Devices. In *Proceedings of the ACM International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, 2020.
[62] X. Meng, R. Du, M. Zwicker, and A. Varshney. Kernel Foveated Rendering. *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, 1(1):1–20, 2018.
[63] Meta. Horizon Worlds. https://www.oculus.com/horizon-worlds/, 2022. [accessed on 19-September-2022].
[64] Microsoft. AltspaceVR. https://altvr.com/, 2022. [accessed on 19-September-2022].
[65] Minecraft. https://www.minecraft.net, 2022. [accessed on 19-September-2022].
[66] A. Mislove, M. Marcon, K. P. Gummadi, P. Druschel, and B. Bhattacharjee. Measurement and Analysis of Online Social Networks. In *Proceedings of ACM SIGCOMM Conference on Internet Measurement (IMC)*, 2007.
[67] Mozilla. Mozilla Hubs. https://hubs.mozilla.com/, 2022. [accessed on 19-September-2022].
[68] Mozilla. Mozilla Hubs System Overview. https://hubs.mozilla.com/docs/system-overview.html, 2022. [accessed on 19-September-2022].
[69] Mozilla. The client-side code for Mozilla Hubs. https://github.com/mozilla/hubs, 2022. [accessed on 19-September-2022].
[70] Oculus. AutoDriver. https://developer.oculus.com/documentation/unity/ts-autodriver/, 2022. [accessed on 19-September-2022].
[71] Oculus. Horizon Worlds Opens To Those 18+ in the US and Canada. https://www.oculus.com/blog/horizon-worlds-opens-to-those-18-in-the-us-and-canada/, 2022. [accessed on 19-September-2022].
[72] Oculus. OVR Metrics Tool. https://developer.oculus.com/documentation/unity/ts-ovrmetricstool/, 2022. [accessed on 19-September-2022].
[73] Oculus. Player Count of Horizon Worlds. https://www.oculus.com/horizon-worlds/learn/tutorial/player-count/, 2022. [accessed on 19-September-2022].
[74] S. Orts-Escolano, C. Rhemann, S. Fanello, W. Chang, A. Kowdle, Y. Degtyarev, D. Kim, P. L. Davidson, S. Khamis, M. Dou, et al. Holoportation: Virtual 3D Teleportation in Real-time. In *Proceedings of Annual ACM Symposium on User Interface Software and Technology (UIST)*, 2016.
[75] S.-M. Park and Y.-G. Kim. A Metaverse: Taxonomy, Components, Applications, and Open Challenges. *IEEE Access*, 10:4209–4251, 2022.
[76] R. Piitulainen, P. Hämäläinen, and E. D. Mekler. Vibing Together: Dance Experiences in Social Virtual Reality. In *Proceedings of ACM Conference on Human Factors in Computing Systems (CHI)*, 2022.
[77] J. M. Pujol, V. Erramilli, G. Siganos, X. Yang, N. Laoutaris, P. Chhabra, and P. Rodriguez. The Little Engine(s) That Could: Scaling Online Social Networks. In *Proceedings of ACM SIGCOMM Conference on Data Communication (SIGCOMM)*, 2010.
[78] F. Qian, B. Han, Q. Xiao, and V. Gopalakrishnan. Flare: Practical Viewport-Adaptive 360-Degree Video Streaming for Mobile Devices. In *Proceedings of ACM International Conference on Mobile Computing and Networking (MobiCom)*, 2018.
[79] Rec Room. https://recroom.com/, 2022. [accessed on 19-September-2022].
[80] Rec Room. Laser Tag. https://rec-room.fandom.com/wiki/Laser_Tag, 2022. [accessed on 19-September-2022].
[81] D. Roberts, T. Duckworth, C. Moore, R. Wolff, and J. O'Hare. Comparing the End to End Latency of an Immersive Collaborative Environment and a Video Conference. In *Proceedings of IEEE/ACM International Symposium on Distributed Simulation and Real Time Applications*, 2009.
[82] Roblox. https://www.roblox.com/, 2022. [accessed on 19-September-2022].
[83] Ryan Schultz. Welcome to the Metaverse: A Comprehensive List of Social VR/AR Platforms and Virtual Worlds. https://ryanschultz.com/list-of-social-vr-virtual-worlds/, 2022. [accessed on 19-September-2022].
[84] D. Saffo, S. Di Bartolomeo, C. Yildirim, and C. Dunne. Remote and Collaborative Virtual Reality Experiments via Social VR Platforms. In *Proceedings of ACM Conference on Human Factors in Computing Systems (CHI)*, 2021.
[85] F. Schneider, A. Feldmann, B. Krishnamurthy, and W. Willinger. Understanding Online Social Network Usage from a Network Perspective. In *Proceedings of ACM SIGCOMM Conference on Internet Measurement (IMC)*, 2009.
[86] S. Shi, V. Gupta, and R. Jana. Freedom: Fast Recovery Enhanced VR Delivery Over Mobile Networks. In *Proceedings of ACM International Conference on Mobile Systems, Applications, and Services (MobiSys)*, 2019.
[87] P. Sykownik, L. Graf, C. Zils, and M. Masuch. The Most Social Platform Ever? A Survey about Activities & Motives of Social VR Users. In *Proceedings of IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, 2021.
[88] P. Sykownik, D. Maloney, G. Freeman, and M. Masuch. Something Personal from the Metaverse: Goals, Topics, and Contextual Factors of Self-Disclosure in Commercial Social VR. In *Proceedings of ACM Conference on Human Factors in Computing Systems (CHI)*, 2022.
[89] P. Sykownik and M. Masuch. The Experience of Social Touch in Multi-User Virtual Reality. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology (VRST)*, 2021.
[90] T. J. Tanenbaum, N. Hartoonian, and J. Bryan. "How do I make this thing smile?": An Inventory of Expressive Nonverbal Communication in Commercial Social Virtual Reality Platforms. In *Proceedings of ACM Conference on Human Factors in Computing Systems (CHI)*, 2020.
[91] Tech Times. Meta's Oculus Quest 2 is the Top Selling VR Headset of 2021. https://bit.ly/3Ltruuq, 2022. [accessed on 19-September-2022].
[92] TechCrunch. Microsoft shuts down AltspaceVR's social hubs to combat harassment. https://tcrn.ch/3DD6wHs, 2022. [accessed on 19-September-2022].
[93] The New York Times. The Metaverse's Dark Side: Here Come Harassment and Assaults. https://www.nytimes.com/2021/12/30/technology/metaverse-harassment-assaults.html, 2022. [accessed on 19-September-2022].
[94] N. Tian, P. Lopes, and R. Boulic. A Review of Cybersickness in Head-mounted Displays: Raising Attention to Individual Wusceptibility. *Virtual Reality*, pages 1–33, 2022.
[95] Uplandme, Inc. Upland. https://www.upland.me/, 2022. [accessed on 19-September-2022].
[96] USA Today. Sexual harassment in the metaverse? Woman alleges rape in virtual world. https://www.usatoday.com/story/tech/2022/01/31/woman-allegedly-groped-metaverse/9278578002/, 2022. [accessed on 19-September-2022].
[97] S. Valluripally, A. Gulhane, K. A. Hoque, and P. Calyam. Modeling and Defense of Social Virtual Reality Attacks Inducing Cybersickness. *IEEE Transactions on Dependable and Secure Computing (Early Access)*, 2021.
[98] P. Vixie. What DNS is not. *Communications of the ACM*, 52(12):43–47, 2009.
[99] VRChat. VRChat. https://hello.vrchat.com/, 2022. [accessed on 19-September-2022].
[100] VRScout. The Top Five Places To Visit In Horizon Worlds. https://vrscout.com/news/the-top-five-places-to-visit-in-horizon-worlds/, 2022. [accessed on 19-September-2022].
[101] K. J. Waldron and J. Schmiedeler. Kinematics. In *Springer Handbook of Robotics*, pages 11–36. Springer, 2016.
[102] Q. Wang, R. Li, Q. Wang, and S. Chen. Non-Fungible Token (NFT): Overview, Evaluation, Opportunities and Challenges. https://arxiv.org/abs/2105.07447, 2021.
[103] J. Williamson, J. Li, V. Vinayagamoorthy, D. A. Shamma, and P. Cesar. Proxemics and Social Interactions in an Instrumented Virtual Reality Workshop. In *Proceedings of ACM Conference on Human Factors in Computing Systems (CHI)*, 2021.
[104] C. Wilson, B. Boe, A. Sala, K. P. Puttaswamy, and B. Y. Zhao. User Interactions in Social Networks and their Implications. In *Proceedings of ACM European Conference on Computer Systems (EuroSys)*, 2009.
[105] Wireshark. https://www.wireshark.org/, 2022. [accessed on 19-September-2022].
[106] Worlds on VRChat. Voxel Shooting. https://bit.ly/3RNSrLz, 2022. [accessed on 19-September-2022].
[107] C. Xie, X. Li, Y. Hu, H. Peng, M. Taylor, and S. L. Song. Q-VR: System-Level Design for Future Mobile Collaborative Virtual Reality. In *Proceedings of the ACM International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, 2021.
[108] T. Xu, B. Han, and F. Qian. Analyzing Viewport Prediction Under Different VR Interactions. In *Proceedings of the International Conference on Emerging Networking Experiments And Technologies (CoNEXT)*, 2019.
[109] Y. Yan, K. Chen, Y. Xie, Y. Song, and Y. Liu. The Effects of Weight on Comfort of Virtual Reality Devices. In *Proceedings of the International Conference on Applied Human Factors and Ergonomics*, 2018.
[110] L. Zhang, L. Sun, W. Wang, and J. Liu. Unlocking the Door to Mobile Social VR: Architecture, Experiments and Challenges. *IEEE Network*, 32(1):160–165, 2017.
[111] W. Zhang, F. Qian, B. Han, and P. Hui. DeepVista: 16K Panoramic Cinema on Your Mobile Device. In *Proceedings of International Conference on World Wide Web (WWW)*, 2021.