An Empirical Evaluation of Video Conferencing Systems Used in Industry, Academia, and Entertainment [Work-in-Progress]

Jim Cuijpers[*]
Vrije Universiteit Amsterdam
Amsterdam, the Netherlands
j.j.c.j.cuijpers@vu.nl

Kelvin Elsendoorn[*]
Vrije Universiteit Amsterdam
Amsterdam, the Netherlands
k.elsendoorn@student.vu.nl

Ean-Dan Tjon-Joek-Tjien[*]
Vrije Universiteit Amsterdam
Amsterdam, the Netherlands
e.tjonjoektjien@student.vu.nl

Riccardo Iesari[*]
Vrije Universiteit Amsterdam
Amsterdam, the Netherlands
r.iesari@student.vu.nl

Federico Casenove[*]
Vrije Universiteit Amsterdam
Amsterdam, the Netherlands
f.casenove@student.vu.nl

Jesse Donkervliet, and
Alexandru Iosup
Vrije Universiteit Amsterdam
{j.j.r.donkervliet,a.iosup}@vu.nl

ABSTRACT

Video Conferencing Systems (VCS) are used daily—at work, in online education, and for get-togethers with friends and family. Many new VCSs have emerged in the past decade and a new market-leader has risen during the coronavirus period of 2020. Understanding how these systems work could help us improve them rapidly. However, no experimental comparison of such systems currently exists. In this work we propose a method to compare VCSs in real-world operation and implement it as a tool. Our method considers four main kinds of real-world experiments. Each captures different aspects, such as communication channels (audio, video, audio-video) and types of network environments (e.g., Ethernet, WiFi, 4G), and reports system and network utilization. We further implement an automated tool to conduct these real-world experiments, and experiment with three popular VCSs, Zoom, Microsoft Team, and Discord. We find that there are significant performance differences between these systems, and their behavior in different environments.

ACM Reference Format:
Jim Cuijpers, Kelvin Elsendoorn, Ean-Dan Tjon-Joek-Tjien, Riccardo Iesari, Federico Casenove, and Jesse Donkervliet, and, Alexandru Iosup. 2021. An Empirical Evaluation of Video Conferencing Systems Used in Industry, Academia, and Entertainment [Work-in-Progress]. In *Companion of the 2021 ACM/SPEC International Conference on Performance Engineering (ICPE '21 Companion), April 19–23, 2021, Virtual Event, France.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3447545.3451174

1 INTRODUCTION

Video Conferencing Systems (VCSs) are essential to communication between people, in business, academia, and for entertainment purposes. For example, the ICPE 2020 conference took place online, in April 2020, and chose to leverage Zoom as its live (synchronous) communication platform [11]. However, relatively to other online and interactive systems [12, 20], we understand little about how these VCSs perform in the real world. The organizing team of ICPE considered many alternatives to Zoom and tested several, but these are anecdotal experiences and currently no peer-reviewed performance study of VCSs exists. To address this problem, we develop a method to compare VCS behavior in real-world settings. We further design and implement an automated tool around this method, and use it to conduct real-world experiments using three popular VCSs; Figure 1 depicts a typical result, for Zoom.

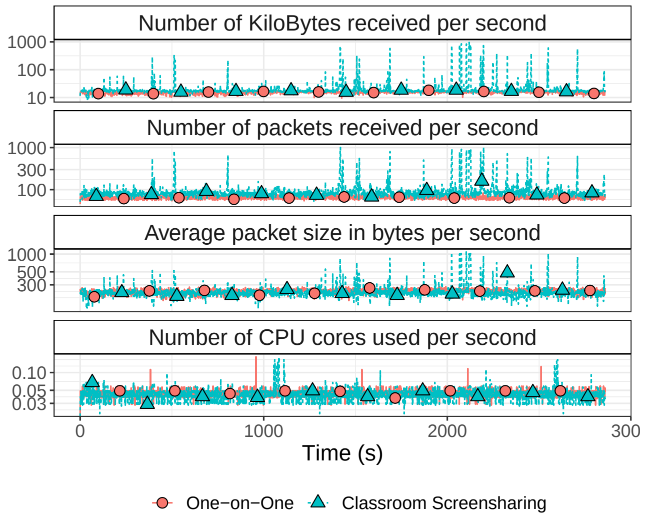

Figure 1: Performance results for Zoom during Experiment 1 from Section 3 (in red) and during a real-world lecture (in blue); performance variability is much higher for the latter.

VCSs are becoming prominent in many areas of business, academia, and entertainment. With in-person meetings limited in the entire world due to COVID-19, VCSs are becoming the de facto tool for meetings; Zoom and Microsoft Teams already service over 100 million users each [21]. Video conferencing is rapidly becoming the norm for internal teams at Microsoft [8] and a large percentage of recruitment interviews occur over video call [3]. In a recent survey by Zoom, respondents suggest using VCSs could help more people get advanced degrees and lower dropout rate [19]. Game streamers combine both in-game and real-world channels [13].

Existing work has studied VCSs from different perspectives, focusing on a particular video-encoding protocol [12], on using

[*]These authors have an equal and leading contribution. The others have proposed the research problem, helped manage the project, and contributed to writing the article.

ACM ISBN 978-1-4503-8331-8/21/04.
https://doi.org/10.1145/3447545.3451174

virtualization platforms to deploy VCSs [23], and from a societal perspective [22, 26]. The ICPE 2020 organizers give a sample performance result—the bandwidth consumption of Zoom during a single conference-session [11, Fig.2]. However, no study analyzes or compares the performance of commonly used VCSs, such as Zoom, (Microsoft) Teams, and Discord.

As suggested by the high-level model for VCSs introduced in Section 2, comparing VCSs experimentally is challenging. *First*, VCSs allow the user to choose between different communication channels, i.e., audio and video, and combined. This choice, in turn, drastically influences the resource consumption and the performance of VCSs. *Second*, the resource consumption of VCS clients, their CPU, memory, and bandwidth, may show idiosyncratic characteristics relatively to other online applications. For example, Figure 1 depicts resource consumption for Zoom in two different settings, an experiment we control and a real-world online lecture. The figure summarizes four main metrics, each in a sub-plot. Among the possible observations supported by this figure, the one-on-one controlled experiment exhibits lower performance variability than the real-world lecture. *Third*, the interplay between network and VCS performance is essential but non-trivial. *Fourth*, and as a last focus of this work, also the codecs used by the VCS affect performance. The codec itself can trade-off local performance (i.e., CPU-limited) with remote operation (i.e., adding bandwidth consumption). Addressing these challenges, we focus in this work on the research question *How to measure and compare the system- and application-level performance of VCSs?*

We also identify as a problem the *lack of open-access datasets about VCS operation*, ready for FAIR [25] consumption.

Addressing the research question and providing a public and Findable, Accessible, Interoperable, and Reusable (FAIR) dataset, the main contribution of this work is three-fold:

(1) We design a method for comparing VCSs and implement it as a tool for real-world experimentation (Section 3). Our method considers various communication channels, e.g., audio and/or video, different types of networks (Ethernet vs. Wi-Fi 4 vs. Wi-Fi 5 vs. 4G LTE), different experiment sizes (one on one vs. entire classroom), etc. The tool automates real-world experiments following the method.
(2) A real-world evaluation of three VCSs (Section 4.2). We focus on the popular VCSs Zoom, Microsoft Teams, and Discord [9] [7] . We evaluate them using 4 different experiments, and analyze system- and application-level performance metrics.
(3) We share FAIRly our results, on Zenodo [5].

2 SYSTEM MODEL FOR VCS

VCSs such as Zoom, Teams, and Discord differ in many aspects, e.g., their architecture [4], how their traffic is routed [2] or redirected [14]. To facilitate the design of a method to experiment across multiple VCSs, in this section we propose a high-level model.

Our model captures the high-level structure and operation of VCSs. Figure 2 is a visual overview of the model. We focus on VCSs based on a client/server communication model. Due to the lack of publicly available information, this model focuses on the client-side and does not detail how the server operates (typically, inside the data center). The model considers two main roles for each client, as *sender* (component ❷ in Figure 2), which captures

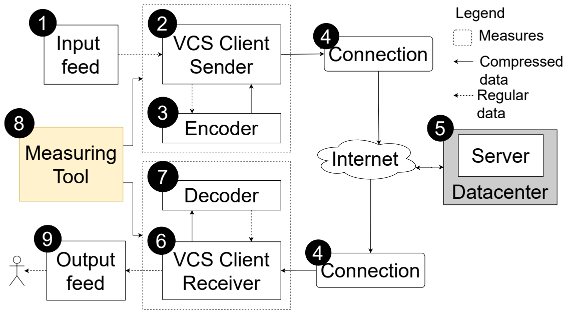

Figure 2: Model of the common VCS under test.

the input feed (❶) and uses an encoder on it (❸); and as *receiver* (❻, mirroring ❷), which uses a decoder on the data it receives (❼) and provides for another participant in the meeting an output feed (❾). The encoding and decoding processes use VCS-specific software.

To reach the network, each client uses a connection (❹). The datacenter redirects the incoming data to a VCS server (❺), which processes and possibly re-encodes it, and sends it to all receivers. Consequently, the (possibly re-encoded) data traverses again the Internet and another network connection, per client.

We do not make any assumptions about the architecture of the server, e.g., centralized or distributed, hierarchical or flat. However, we make the assumption that clients do not form their own interconnections, bypassing the data center round-trip.

Our model considers two main types of data-streams, regular and compressed. The former are unencrypted data, the audio and/or video streams the user experiences. Compressed data are encrypted, often smaller-sized, possibly re-encoded, streams.

3 DESIGN OF EXPERIMENTS AND AUTOMATED EXPERIMENTATION TOOL

In this section we propose a method to compare VCSs experimentally, and implement the method as an automated tool.

3.1 Method Overview

In our method, clients run on different machines in the network, and the input feed represents the benchmark *workload*. To evaluate the performance of a VCS, the measuring tool (component ❽ in Figure 2) runs on one client, capturing traffic-metrics and other resource-consumption information, and later calculates performance metrics. It measures computer and network metrics, focusing on: CPU utilization; and network bandwidth, number of packets, and packet sizes, each of these network metrics are measured in and out.

Our method considers four major operational aspects: regular operation of VCS systems, and operation under constrained bandwidth, with various connection types, and under a large workload. We design a specific experiment for each; Table 1 summarizes them. *Experiment 1* captures regular operation with various input channels enabled, i.e., audio, video, and audio+video; for all other experiments, we use only the latter, audio+video. *Experiment 2* throttles (limits) the download speed (bandwidth) available to clients, lowering it by 50 kBps every minute from the initial limit (of 500 kBps, for a 10-minute experiment). *Experiment 3* compares the performance with a wired connection (Ethernet) against various wireless

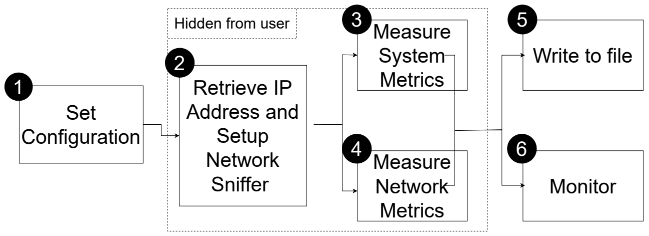

Figure 3: Our automated workflow for evaluating VCSs.

connections, i.e., Wi-Fi 4 (802.11n) and 5 (802.11ac), and 4G LTE.[1] *Experiment 4* consider a larger workload. Section 3.2 details the workload for each experiment.

Limitations: Our method represents work in progress toward understanding the performance characteristics of VCSs used prominently from 2020 on. We specifically do not consider within the scope of this work: throttling local resources other than bandwidth, setting encoding and re-encoding parameters for audio and/or video, uncovering the operation and architecture of VCS servers, analyzing delay and the impact of network conditions (e.g., jitter, delay) on it, assessing quality of experience from the user's perspective, considering a larger set of VCSs and deeper VCS models.

3.2 Workload for Each Experiment

All experiments maintain the default video and audio settings, for all systems under test. The video *resolution* defaults to 1920 × 1080, a commonly used resolution for 2020.

The basis of each workload consists of (1) for video-only, a video shared FAIRly [18], with a duration of 2:36 minutes played in a continuous loop; (2) for audio-only, an audio stream generated by turning on the public radio and recording using the microphone (channel NPO2 [1]); (3) for audio+video, both, combined and looped. These are real-world fragments, and workloads combining them are realistic and reproducible.

Experiment 1 (see Table 1) considers each of the three types of workload, in turn. *Experiments 2* and *4* consider only the combined workload (type 3). Last, *Experiment 3* contrasts the combined workload with the workload produced by a real-world classroom (an interactive session in our Distributed Systems course); although the classroom workload is not fully reproducible, we conjecture that many classrooms or generally interactive sessions with a central moderator will generate very similar workloads. We repeat the experiments for every system 50 times. This is done by taking 1-minute samples from a benchmark that we run for 50 minutes.

3.3 Tool to Automate Experiments

In this section, we design and implement a tool to automate the experiments proposed in our method for comparing VCSs. We have tested the tool and found it able to work with popular VCSs, such as Zoom, Teams, and Discord.

Figure 3 depicts the workflow our tool automates. First, the tool sets up the configuration (component ❶); the user only gives the network interface as input. Then, the tool retrieves the IP address used by the VCS, and sets up a network sniffer for it (❷). To achieve this, the tool leverages PyShark [15], which is a Python wrapper for the popular network analyzer Wireshark/Tshark. Over the course of the experiment, the tool collects all the metrics (❸ and ❹), using Psutils [10]. Wondershaper [16] throttles the bandwidth (used in *Experiment 2*, see Table 1). The tool writes the results to a file (❺) and/or displays them for the experiment admin to see (❻).

[1]Wi-Fi 6 routers are relatively expensive and thus not widely deployed.

Table 1: Overview of experiments. *Scale* indicates the number of participants. *Cha.* indicates the use of *A*udio, *V*ideo, or both channels (*A+V*). *Dur.* is the duration of the experiment, in minutes. *Reps.* is the number of repetitions.

ID /	Focus	Workload			Test	
Sec.	(keyword)	Scale	Cha.	Dur.	SUT	Reps.
1/§4.2.1	Channels	2	A±V	1	All	50
2/[5]	Throttling	2	A+V	10	All	1
3/§4.2.3	Scalability	≥20	A+V	45	Zoom	1
4/§4.2.4	Connections	2	A+V	1	Zoom	10

4 REAL WORLD EXPERIMENTS

We use our method discussed in Section 3 to evaluate the performance of three popular VCSs through real-world experiments. Table 1 gives an overview of our experiments, from which we derive the following **M**ain **F**indings:

MF1 There are large performance differences between different VCSs under the same workload.
MF2 Bandwidth limitations lead to local CPU-usage limitations.
MF3 Increasing the number of participants and video streams introduces high performance variability.
MF4 Wireless networks significantly increase resource performance variability compared to wired Ethernet. Wi-Fi 5 is more stable than both Wi-Fi 4 and 4G LTE.

4.1 Experimental Setup

The workloads used in our experiments are shown in Table 1. Each experiment is conducted using a PC equipped with a Ryzen 7 2700X CPU (8 logical cores) and 16GB DDR4 memory.

The wired connection is Ethernet-based, with the machine connected to the Internet via the Dutch ISP *KPN NetwerkNL*, an ADSL with 100 Mbps download bandwidth and 10 Mbps upload bandwidth. The wireless connections use a high-quality AC1750 Wi-Fi router, via and the same Internet connection. The 4G LTE connection carries through KPN, a high-quality provider.

4.2 Results of Real-World Experiments

In the remainder of this section we describe the results of real-world experiments leading to **MF1** through **MF4**. For detailed analysis of our results please refer to our technical report [6].

*1. Varying Input Channels (Leads to **MF1**).* Figure 4 shows the results of Experiment 1, in which we observe the behavior of our VCSs while using an audio+video workload, a video-only workload, and an audio-only workload. The results show large performance differences between the observed VCSs. The CPU utilization is expressed as the number of utilized CPU cores, where a value of 1 is equivalent to 100% CPU utilization on a single core. The maximum value is equal to the number of logical CPU cores (8, in

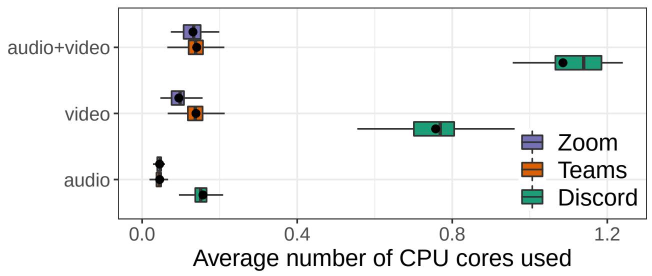

Figure 4: Average number of CPU cores used for Zoom, Teams, and Discord. Setting: one-on-one, audio±video call.

our experiments). When using audio+video and video, Discord has a significantly higher CPU utilization than both Microsoft Teams and Zoom. When using audio, Discord's CPU usage remains higher than the CPU usage of Teams and Zoom, but by a lower margin. This indicates that Discord is less CPU-efficient, especially when handling video feeds. All three systems show a significant reduction in CPU utilization when using only audio. The results for *bandwidth* utilization for this experiment appear in the technical report [6]: importantly, for audio+video, the median bandwidth use for Discord is 500 kBps, whereas for Teams and Zoom the median and the entire IQR remain under 200 kBps [6, §V.A].

*2. Throttling Network Bandwidth (**MF2**).* We cover Experiment 2 in our technical report [6]. To experiment the bandwidth throttling we limited the receiving client. In a stream-oriented communication, codecs play a key role to achieve high-quality audio and video, yet with reduced resource usage. Zoom uses the codec H.264 Annex G(SVC) [24], which is a Scalable Video Codec (SVC). For it, we expect a decrease in CPU utilization when bandwidth throttling is applied. Teams uses the H.264 Advanced Video Codec (AVC) [17], which use a single bit-rate for a stream. We expect its CPU utilization to decrease when the bandwidth is throttled, as the codec will need to run the same algorithm on less data. The Audio Codec used by Teams and Zoom are SILK [17] and Opus. Since they both use the same technology for streaming audio frames we did expect the same results in terms of audio-only workload for the CPU utilization. Overall, our results are consistent with these expectations, but also we have encountered problems in properly applying bandwidth throttling. We leave solving this issue for future work.

*3. Scalability Experiment (**MF3**).* Experiment 3 compares two settings, (1) a small one-on-one discussion and (2) a large an interactive classroom of over 20 persons, both using Zoom. Figure 1 depicts the results; for more details, see our technical report [6]. Overall, the classroom-related results exhibit much higher variability—the curves have higher and more peaks. This matches the expectation of more dynamic nature of the classroom activity.

*4. Different Connection Types (**MF4**).* We cover Experiment 4 in our technical report [6]. The main findings are: (1) When using a wireless connection type, both the variability and the maximum values increase for all three metrics, and (2) Wi-Fi 5 shows less variability than both Wi-Fi 4 and 4G LTE.

5 CONCLUSION AND FUTURE WORK

We designed an experimental method to analyze and compare VCSs such as Zoom, Microsoft Teams, and Discord. Our experiments consider this kind of software as black-boxes, and subjects each system under test to a set of controlled and uncontrolled experiments. As our results indicate, the method allows us to compare the resource consumption of VCSs in settings with different properties: in one-on-one and classroom environments, with one or both of the audio and video channels enabled, using various kinds of network connections, and under limited (throttled) bandwidth. We have presented four main findings. Among the detailed results, we see strong evidence that Discord uses significantly more CPU utilization and bandwidth compared to Teams and Zoom during video streaming. For future work, we would like to measure server-side metrics such as latency and jitter. A community effort should continue this task—join our effort!

REFERENCES

[1] NPO Radio 2. 2020. De Staat van Stasse - donderdag 19 november 2020. https://www.nporadio2.nl/destaatvanstasse/gemist/uitzendingen/931742/2020-11-19
[2] Tom Arbuthnot. 2019. Where in the world will my Microsoft Teams meeting be hosted? http://bit.ly/2ZspDyA
[3] Mary Baker. 2020. Gartner HR Survey Shows 86 of Organizations Are Conducting Virtual Interviews to Hire Candidates During Coronavirus Pandemic. https://gtnr.it/33Zr9uM
[4] Salman Baset and Henning Schulzrinne. 2006. An Analysis of the Skype Peer-to-Peer Internet Telephony Protocol. In *INFOCOM*.
[5] Cuijpers et al. 2020. An Empirical Evaluation of Video Conferencing Systems Used in Industry, Academia, and Entertainment: Dataset. https://doi.org/10.5281/zenodo.4449679
[6] Cuijpers et al. 2020. *An Empirical Evaluation of Video Conferencing Systems Used in Industry, Academia and Entertainment in 2020.* Technical Report. Vrije Universiteit Amsterdam. https://doi.org/10.5281/zenodo.4452480
[7] Discord. 2021. About Discord | Our Mission and Values. http://bit.ly/3beUe8c
[8] Forbes Insights Team. 2017. Optimizing Team Performance: How and Why Video Conferencing Trumps Audio. https://bit.ly/2W6gx92
[9] Gartner. 2020. Gartner Magic Quadrant for Meeting Solutions 2020. http://bit.ly/3bkUVNr
[10] Giampaolo Rodola. 2020 (accessed). psutil. https://bit.ly/2Ltk9Qv
[11] Iosup et al. 2020. Flexibility Is Key in Organizing a Global Professional Conference Online: The ICPE 2020 Experience in the COVID-19 Era. *CoRR* (2020). https://arxiv.org/abs/2005.09085
[12] Jansen, Kuipers, et al. 2017. Performance Evaluation of WebRTC-based Video Conferencing. *SIGMETRICS Perform. Evaluation Rev.* 45, 3 (2017), 56–68.
[13] Jia et al. 2016. When Game Becomes Life: The Creators and Spectators of Online Game Replays and Live Streaming. *ACM ToMM* 12, 4 (2016), 47:1–24.
[14] Peter Judge. 2020. Most of Zoom runs on AWS, not Oracle - says AWS. http://bit.ly/3do6oyw
[15] KimiNewt. 2020 (accessed). pyshark. https://bit.ly/3nbD5kz
[16] Magnific0. 2020 (accessed). Wondershaper bw/ throttling. https://bit.ly/345EFgt
[17] Rodolfo Castro. 2020. Is your company's network ready for Microsoft teams? https://bit.ly/3m52kE5
[18] RoyaltyFree Videos for content creators. 2018. RoyaltyFree Channel Trailer Free To Use Stock Footage No Copyright. https://bit.ly/3a7NB97
[19] S. Ann Earon. 2020 (accessed). The Value of Video Communications in Education. https://bit.ly/2IDpN1i
[20] Mirko Suznjevic and Maja Matijasevic. 2013. Player behavior and traffic characterization for MMORPGs: a survey. *Multim. Syst.* 19, 3 (2013), 199–220.
[21] TechRepublic. 2020. Watch out Zoom! https://www.techrepublic.com/article/watch-out-zoom-microsoft-teams-now-has-more-than-115-million-daily-users/ 300 million users for Zoom and over 100 million for Teams in Oct 2020.
[22] Townsend et al. 2001. Desktop video conferencing in virtual workgroups: anticipation, system evaluation and performance. *Infor. Sys. J.* 11, 3 (2001), 213–227.
[23] Vasconcelos et al. 2016. Virtualization technologies in web conferencing systems: A performance overview. In *ICITST*. 376–383.
[24] Vikram Sachdeva. 2018. Zoom — Video conf app at scale. https://bit.ly/3gEgcDT
[25] Wilkinson et al. 2016. The FAIR Guiding Principles for scientific data management and stewardship. *Nature SciData* 3 (2016).
[26] Xiong et al. 2017. Design and evaluation of a real-time video conferencing environment for support teaching: an attempt to promote equality of K-12 education in China. *Interact. Learn. Environ.* 25, 5 (2017), 596–609.