Towards Enabling Residential Virtual-Desktop Computing

Hongying Dong, Aaron T. Kinfe, Jiakai Yu, Qi Liu, Dan Kilper, *Senior Member, IEEE*, Ronald D. Williams, *Senior Member, IEEE*, and Malathi Veeraraghavan§, *Fellow, IEEE*

Abstract—Commercial virtual-desktop computing is well established using computers optimized to serve as thin clients connected to centralized computing systems. Some common residential applications impose more stringent requirements on both communication bandwidth and latency than those of typical commercial applications. This paper describes an objective study of residential applications accessed through thin-client virtual-desktops for the purpose of investigating the feasibility of applying virtual-desktop computing to residential users. New metrics are introduced to quantify user-received application performance. The results suggest that certain commercial solutions with a commodity datacenter server show a strong potential for being adapted to residential virtual-desktop computing.

Index Terms—edge cloud, measurements, remote desktops, virtual desktops, VDI, thin client, residential environment, video playback.

1 Introduction

CLOUD computing can support a wide range of applications as it provides flexible infrastructure for on-demand scaling. Virtual desktop infrastructure (VDI) is one such application supported by cloud computing. Unlike most cloud computing, VDI is designed to perform computational tasks in the cloud and transport the resulting display and I/O updates over the network with little or no computation taking place in the client, allowing professional information technology (IT) staff to administer user desktops on virtual infrastructure hosted in the datacenter while users access their desktops using a remote desktop protocol [1]. Commercial entities have developed specialized tools to support this paradigm for enterprise users. Examples include Amazon WorkSpace (AWS), Microsoft Remote Desktop Services (RDS), Citrix' offering of Desktop-as-a-Service (DaaS) and Software-as-a-Service (SaaS), Google Cloud Platform, and VMware VDI solutions. These virtual desktops run over thin or fat client machines in terms of the varied computing capabilities. Zero clients are also available in which a specialized communication processor, developed by Teradici, enables VDI clients to run without a central processing unit (CPU) [2].

Enterprise organizations license commercial cloud-based virtual-desktop services because these services enable computing resource consolidation. Such consolidation can dramatically reduce expenses as the hardware and management costs for client-side equipment are minimized. More importantly, this consolidation allows resource-sharing among enterprise employees. For example, Teradici claims support for creative teams of Sony Pictures Imageworks by delivering high resolution, full frame-rate 3D graphics and high definition multimedia to remote workstation users. The users have worked remotely across Southern California and Vancouver with access to the same data and information in real time [3].

Despite this progress in commercial use, few residential customers use virtual-desktop resources in the cloud. Network availability and bandwidth are critical factors as VDI clients support little or no computing without a reliable network connection. From this perspective, fifth generation (5G) wireless communication [4] and edge cloud computing offer expanded opportunities for virtual-desktop techniques applied to personal or home computing. The 5G service category of ultra-reliable low latency communication (URLLC) together with the increased bandwidth and machine to machine communication has the potential to enable general consumer use of virtual desktops. Since edge computing decentralizes the processing power to the edges, using an edge cloud to provide performance levels at the top range of the VDI enterprise categories, residential customers would no longer suffer from limited network bandwidth and may enjoy a full-PC experience using lower cost thin-client interfaces. Further, high-performance residential virtual desktops would provide residential customers with the resource sharing and cloud management enjoyed by commercial users.

Practical residential virtual desktops would also enhance cybersecurity for all. A poorly-maintained and patched consumer device can easily be turned into a weapon that potentially harms others. The failure to patch operating systems and applications and residential user susceptibility to phishing and other similar attacks contribute to the success of cyberattacks that may be used to spread malware. Infected residential devices can be aggregated into large-scale botnets, which can be used to launch attacks such as

• *H. Dong, A.T. Kinfe, Q. Liu, R.D. Williams, and M. Veeraraghavan are with the University of Virginia, Charlottesville, VA 22904. E-mail: {hd7gr, atk4a, ql8va, rdw, mv5g}@virginia.edu.*

• *J. Yu and D. Kilper are with the University of Arizona, Tucson, AZ 85721. E-mail: jiakaiyu@email.arizona.edu, dkilper@optics.arizona.edu.*

• *D. Kilper is also with Palo Verde Networks, Inc.*

§. Deceased May 11, 2020

Distributed Denial of Service (DDoS) on third parties. In September 2016, the Mirai botnet took the Internet by storm with massive DDoS attacks [5], revealing vulnerabilities in the software installed on IoT devices, and showing the importance of limiting access to processors to reduce the cyberattack surface. Broader deployment of virtual-desktop technology centralizes computing resources so that they can be administrated by professionals with responsibility to reduce exposure to malware and other cyberattacks.

This work seeks to determine the conditions under which typical applications used by residential users can achieve acceptable performance when provided through existing commercial virtual desktop tools via edge cloud using a thin-client model. We address the following questions:

1) Can technologies developed to provide virtual desktops to enterprise users be applied to residential use? If so, what network conditions for edge cloud are needed for good performance for residential user applications?
2) What is the performance of the edge cloud virtual-desktop solution using a notebook thin-client device with a high-resolution monitor?
3) How can you maximize the number of users that can share edge cloud resources while each user experiences good performance using edge cloud virtual desktops?
4) How does a commodity datacenter server differ from a server with a graphics processing unit (GPU) in personal computing performance? We ask this question because commodity data-center servers do not include GPU capabilities and commercial virtual-desktop offerings with GPUs are more expensive.

We selected the video streaming application as our residential use-case both because it is a common residential application and it requires frequent display updates with many changes in pixel values over a range of performance levels. Thus, this typical application demands relatively high computation and communication bandwidth that can be varied systematically.

We answer our problem statement questions using available virtual-desktop technologies. In this work, we deploy the edge cloud VMware Horizon 7 Architecture using a commodity data center server. We experimentally evaluate the performance of video playback using a thin-client model by leveraging different remote display protocols with various computing resources and network conditions. To answer the first question, we measure user-received multimedia quality under an ideal network condition and with altered network conditions using a high-end desktop as the client device to remove any impact on performance introduced by the end-user interface. For the second question, we compare video quality using affordable thin-client devices to that of the high-end user device to illustrate practical virtual-desktop performance in a residential-use scenario. We study sharing through VMware's resource allocation shares and monitor video playback quality as a function of computational resource availability to maximize the number of users that can be sustained with the video streaming application using virtual desktops. We measure the impact of edge cloud server capabilities on residential virtual desktops to address our last concern regarding cost and performance. Though other situations may arise, a complete study considering all possible factors would be beyond the scope of this paper.

The main contributions of this paper are as follows:

1) New metrics, user-received PSNR and SSIM (recv-PSNR, recv-SSIM), were defined to quantify the quality of user-received frames during video playback.
2) We conducted an objective study applying commercial-offering virtual-desktop solutions to residential use.
3) We proposed an engineering solution to maximize the number of satisfied users sharing edge cloud computing resources based on empirical results.
4) We measured virtual-desktop capabilities with GPU virtualization enabled at the server side.

The remainder of this paper is organized in the following way: Section 2 provides background on VDI and challenges raised by the residential environment, and reviews related work. Section 3 illustrates our VMware Horizon 7 deployment. Section 4 describes the objective evaluation approach including input media, virtual-desktop specification, remote display protocol, media player, and metrics. Experimental results for user-received media quality evaluation, edge cloud resource sharing, and virtual-desktop capabilities with GPU-virtualization are presented in Section 5, 6, and 7, respectively. The paper is concluded in Section 8.

2 Background and Related Work

2.1 VDI and Commercial Offerings

Figure 1 illustrates the edge cloud VDI computing approach in which virtual desktops are hosted on the edge-cloud. The client drives user devices via remote-desktop protocols with encryption and video decoding to access virtual-desktop services. Remote-desktop protocols send user inputs, including keyboard strokes and mouse clicks, and display updates over the network. All application code is executed on a remote-desktop server. Well-known remote-display protocols include Microsoft Remote Desktop Protocol (RDP), Teradici PC-over-IP protocol (PCoIP), VMware Blast Extreme display protocol (Blast Extreme), and the open-source Remote Frame Buffer protocol (RFB).

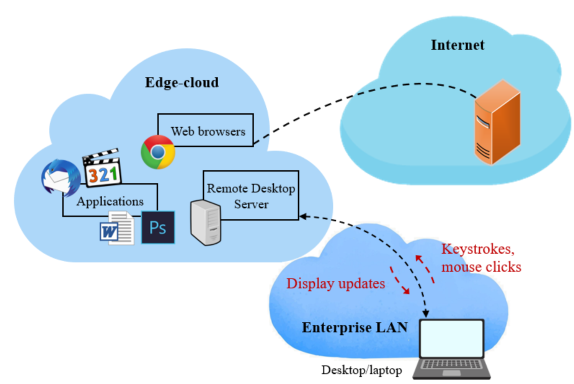


Fig. 1: Edge cloud VDI computing approach

Allied Market Research [6], reports that the global cloud-based VDI market was valued at $3.645 billion in 2016, and is expected to grow with a compound annual growth rate (CAGR) of 16.5% from 2017 to 2023 to reach at $10.154

billion. With the rapid spread of VDI enterprise usage, several commercial companies that offer cloud-based VDI service stand out. Amazon WorkSpaces, owned by Amazon, provides a managed secure Desktop as a Service (DaaS), which utilizes Amazon WorkSpaces Streaming Protocol (WSP) to deliver users Windows or Linux desktops. Microsoft Azure, as a cloud-computing platform designed for building, testing, deploying, and managing applications and services through data centers, provides Software as a Service (SaaS), Platform-as-a–Service (PaaS), and Infrastructure-as-a-Service (IaaS) offerings. Citrix Workspace, developed by Citrix Systems, allows multiple users to remotely access and operate Microsoft Windows desktops hosted on cloud. Teradici, as mentioned above, offers a proprietary remote display protocol, PCoIP, that supports zero clients and thin clients for VDI, specializing in lossless, high performance video coding. Whereas VMware provides the guest operating system with completely virtualized hardware. Commercial offerings considered above are limited to those specifically intended to provide remote virtual desktops. Other offerings intended for remote control or other applications that differ form the remote desktop application may not be included.

2.2 Residential Environment

While virtual desktops have proven their value in commercial environments, deployment in the targeted residential environment imposes at least four significant additional challenges:

- First, the abilities of users are much broader in the targeted residential environment than in the typical commercial environment. User technical abilities will vary from highly sophisticated to very basic. For example, users will include bright students who want to experiment with application development with a grandparent who struggles with web browsing and email.
- Second, the target environment is highly cost sensitive so that the end terminal devices must be relatively inexpensive. The number of users will likely be very large with a smaller percentage of simultaneous users than might be expected in a commercial environment. Therefore, spending for resources should be weighted toward the servers while the cost of terminal devices is minimized.
- Third, the mix of applications is expected to be broader than those of any single commercial deployment. Typical residential applications are expected to include basic office applications such as web browsing, email, and word processing. In addition, residential uses will want media streaming and gaming. As personal applications advance, users will expect more demanding applications such as immersive virtual reality.
- Fourth, because the envisioned residential environment will require sharing of resources, policies and procedures for acquisition and use of those resources must be established. While a user who owns a PC controls the software installed on that PC and the data stored on that PC, the residential users of the virtual desktop system will need a way to decide what software will be installed on the shared servers, how the resources will be shared, and who will control the data.

We provide users with a concise user interface (UI) by deploying commercial VDI solutions. Users with minimum abilities can access their assigned virtual desktops by simply double clicking the icon, whereas technical users are able to change virtual-desktop parameters such as remote display protocols. Further addressing the first challenge is beyond the scope of this work as it requires subjective studies among residential users. We address the second challenge by evaluating residential application performance with real-world thin client machines and discussing the impact of server-assigned computing resources and server capabilities on the client-received application performance. We select video streaming applications to address the third challenge. While video streaming can be found in commercial applications, this application is expected to be central to the residential expectations. Given that residential users vary, different types of videos are used in evaluation, including sporting events, movies, and cartoons. Further, we address the fourth challenge by proposing an engineering solution to maximize the number of residential users sharing edge cloud computing resources.

2.3 Related Work

Several prior efforts have quantified virtual-desktop performance. Nieh *et al.* [7] first developed slow-motion benchmarking to evaluate the performance of thin-client systems by monitoring network traffic traces between a thin client and its server. Results were obtained by comparing traffic traces collected from an ideal environment with traces collected under different network connections. This technique has then been applied in many studies [8], [9], [10], [11] [12], [13], [14]. For example, VDBench, proposed by Berryman *et al.* [10], is a thin-client toolkit that uses the slow-motion technique to evaluate remote user experience in terms of video quality. However, all of the prior work relied heavily on traffic trace monitoring, which does not directly represent user-received virtual-desktop quality. User experience has been studied through subjective tests; however, to the best of our knowledge, our work is the first reported study that applies objective evaluation on end-user client performance for residential use of the virtual-desktop computing approach.

Previous work has considered streaming video quality generally and developed various measurement methodologies. Chan *et al.* [15] proposed a novel method, MPSNR, to address the inaccuracy of the peak signal-to-noise ratio (PSNR) calculation caused by consecutive packet losses. Since PSNR compares every pixel in each frame of a processed video with the corresponding pixel in each frame of the reference video, the consecutive losses of packets lead to the loss of an entire frame, resulting in the comparison of two non-corresponding frames. Laine and Hakala [16] evaluated streaming Quality-of-Service (QoS) performance by collecting pixel information and the display time of video pictures using FFplay. Our work provides a more comprehensive performance assessment by extending metrics for streaming video quality evaluation beyond PSNR and playback duration to include user-received FPS and structural similarity (SSIM).

Virtual Speech Quality Objective Listener, or ViSQOL, proposed by Hines *et al.* [17], is a signal-based full reference metric designed as an alternative to the commercial Perceptual Evaluation of Speech Quality (PESQ) metric [18] for Voice-over-IP (VoIP). ViSQOLAudio, an adaption of ViSQOL, was proposed by Hines *et.al* in 2015 [19] for streaming audio evaluation. Later in 2018, Narbutt *et.al* presented AMBIOUAL [20], a full reference objective quality metric for ambisonic spatial audio. In this study, we measure the user-received audio quality delivered by display protocols using ViSQOLAudio.

Recent research on audio-visual quality perception takes advantage of machine learning techniques. Kumar *et.al* [21] proposed a time-evolution model using features extracted from the speaker's facial area and the discrete cosine image transform to detect audio-visual synchronization in video segments containing a speaker in frontal head pose. Marcheret *et al.* [22] investigated the use of deep neural networks (DNNs) and proposed synchrony DNNs that directly operate on audio and visual features to detect audio-visual synchronization for human speech. However, none of these metrics are applicable to our thin-client case due to the limited computing power available on the thin clients. Instead, we develop our audio-visual synchronization metric by extracting time shifts between the sound and vision using a specialized multimedia file.

3 VMware Horizon 7 Deployment

Our study used a Dell PowerEdge R740 Server configured to operate as the edge-cloud server with VMware ESXi 6.5.0, 24 CPUs $\times$ Intel(R) Xeon(R) Silver 4116 CPU @ 2.10 GHz, 96 GB RAM, and four Network Interface Cards (NIC) of 2x1 GE and 2x10 GE.

3.1 VMware Horizon 7

VMware Horizon 7 is a Virtual Desktop Infrastructure (VDI) solution provided by VMware intended to offer flexibility, reliability, and security to end users. It is a VDI platform that delivers secured desktops to remote end users from a data center. End users can access on-demand or up front provisioned desktops using laptops, zero-clients, thin-clients, tablets, and even smartphones. Desktops are managed, updated, and destroyed centrally from a VMware vSphere server.

VMware vSphere is a suite of virtualization products that provides a scalable platform for running virtual desktops and applications. vSphere contains three main components: ESXi, vCenter, and vSphere Client. ESXi is a type-1 hypervisor functioning as the virtualization server on which all virtual machines are installed. VMware vCenter Server is a centralized management application through which multiple connected hosts in a network are managed and host resources are pooled. vCenter Server is installed as a virtual machine on top of ESXi. vSphere Client, an HTML5-based management portal, is required to install, manage, and access virtual desktops. It resides above ESXi. It can also access vCenter Server for management purposes.

3.2 Anvil Deployment

Our deployment of VMware Horizon 7 is shown in Figure 2. The physical ESXi server, named Anvil, is connected to a network switch through a 1 GE NIC for management, and a 10 GE NIC for the VM network and Internet access. The VM network contains five components: Active Directory Domain Controller, MSSQL Server, VMware vCenter Server, VMware Horizon Connection Server, and pfSense Software Firewall. Users access virtual desktops by establishing connections to the Anvil Connection Server via the Horizon Client installed on user devices.

4 Evaluation Approach

In this study, we quantify the performance of edge cloud virtual-desktop computing for video-streaming applications.

4.1 Input Media

We selected three videos from the Waterloo Quality-of-Experience Database [23], a well-designed RAW HD video database, as the references for our video-quality evaluation. We compressed the reference videos using an H.264 codec in the constant-quality mode with CRF (constant rate factor) 18. The slow preset was applied for video compression to achieve a balance of encoding speed and relatively good quality. Duration and resolution were kept the same as those of the reference. We used PSNR and SSIM to measure the quality of compressed videos. PSNR (peak signal-to-noise ratio) is the ratio between the maximum possible power of an image and the power of corrupting noise that affects the quality of its representation, and SSIM (structural similarity index) is a method of predicting the perceived quality digital images. PSNR and SSIM are full-reference metrics that positively correlate with Mean Opinion Score (MOS), which is a measure used in the domain of Quality of Experience that represents the overall quality of a stimulus or system [24], as shown in Table 1. Compressed videos are usually considered excellent in quality with a PSNR greater than 37, or SSIM equals to or greater than 0.99 [25]. Table 2 shows the information and quality of our input videos, ranging from high-motion to static, illustrating heavy to light workload.

TABLE 1: PSNR and SSIM to MOS mapping

PSNR (dB)	SSIM	MOS	Quality
≥ 37	≥ 0.99	5	Excellent
≥ 31 & < 37	≥ 0.95 & < 0.99	4	Good
≥ 25 & < 31	≥ 0.88 & < 0.95	3	Fair
≥ 20 & < 25	≥ 0.5 & < 0.88	2	Poor
< 20	< 0.5	1	Bad

To evaluate application performance with audio included, we selected multimedia files from the UnB-AVQ-2013 database [26]- [27], a publicly-available digital video library designed for audio-visual assessment. Details of our selected multimedia files are shown in Table 3.

A special multimedia file designed specifically for synchronization testing was selected for audio-video synchronization evaluation. [28] This file was further trimmed and cropped for use in our measuring environment. The

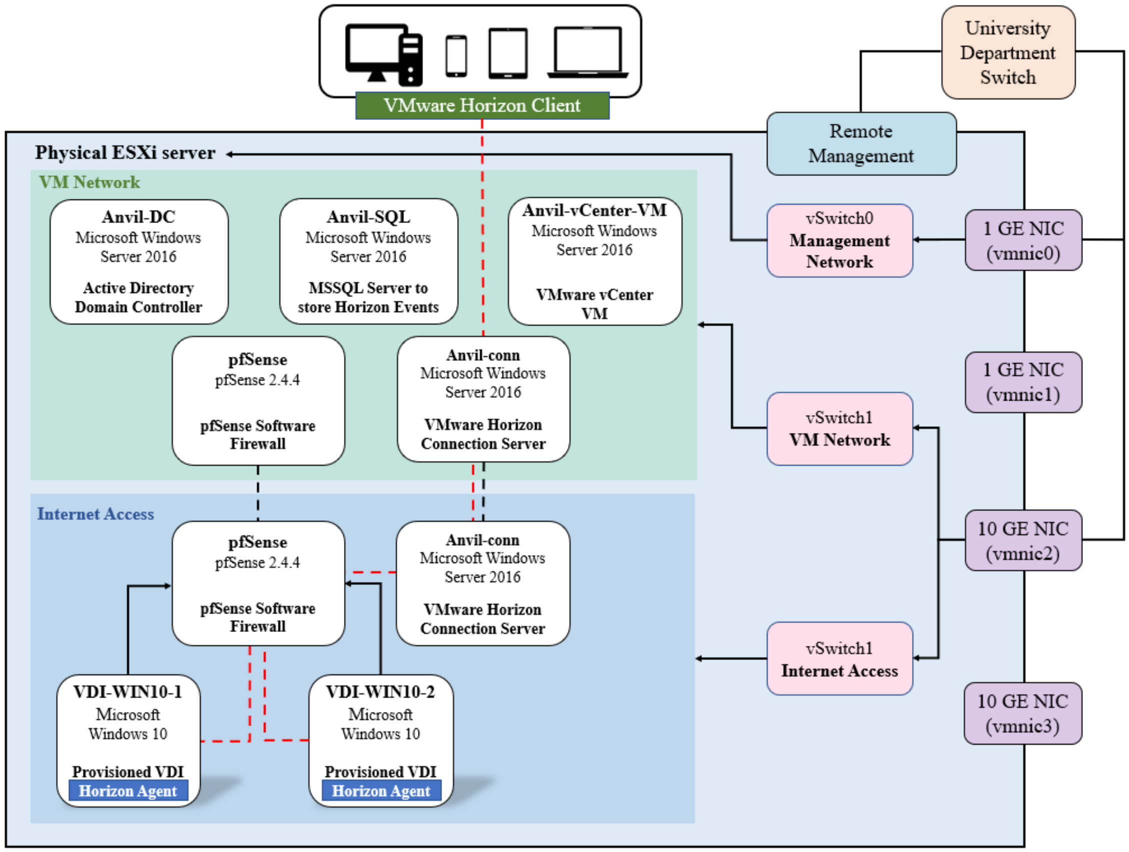


Fig. 2: VMware Horizon 7 Anvil deployment

TABLE 2: Input video parameters

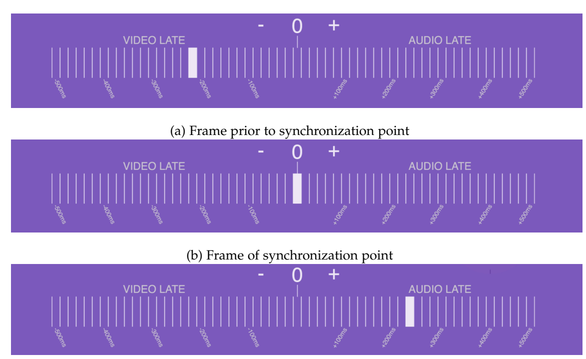
	Basketball	Squirrel	Transformer
Codec	H.264	H.264	H.264
Mode	CQ (crf 18)	CQ (crf 18)	CQ (crf 18)
Resolution	1920 × 1080	1920 × 1080	1920 × 1080
Duration(s)	10	10	10
Scene	Sports	Animation	Movie
Characteristic	High-motion	Intermediate	Static
File size (MB)	30.3	11.7	7.7
Avg. FPS	25/1	25/1	24/1
Avg. bit rate (Mbps)	24.22	9.33	6.16
PSNR (dB)	39.22	42.07	45.35
SSIM	0.9983	0.9980	0.9982
Video quality	Excellent	Excellent	Excellent

TABLE 3: Input multimedia parameters

	Crowd Run	Basketball
Resolution	1280 × 720	1280 × 720
Duration(s)	8	8
Scene	Running	Playing basketball
Video Characteristic	High-motion	High-motion
Audio Characteristic	Music with ambient noise	Ambient noise
Avg. FPS	30/1	30/1

	Music	Reporter
Resolution	1280 × 720	1280 × 720
Duration(s)	8	8
Scene	Playing the guitar	Human talking
Video Characteristic	Static	Static
Audio Characteristic	Music	Speech
Avg. FPS	30/1	30/1

modified multimedia file is a 10-second long, 60-FPS video containing 10 evenly distributed synchronization points. Each video synchronization point occurs at the time when an audio beep occurs. Further, the video contains a tick mark strip and a white bar that moves along the tick mark strip during video playback, as shown in Figure 3. For each audio synchronization point, the moving white bar is expected to appear at a fixed position along the tick mark bar, and that position is used as the audio synchronization reference in the evaluation process.



(a) Frame prior to synchronization point

(b) Frame of synchronization point

(c) Frame after synchronization point

Fig. 3: Frames of multimedia for synchronization evaluation

4.2 Virtual-desktop Specifications

Table 4 shows specifications for the virtual desktops deployed for performance evaluation in this study. We chose 1 vCPU (2100 MHz) with 2 GB vRAM as the minimum specification because our guest operating system, 64-bit Windows 10, required a minimum of a single 1 GHz processor and 2 GB of RAM. We increased the processor of our minimum specification to 2 GHz (i.e., 1 vCPU) because our Anvil server had no GPU and in qualitative observations performance was unacceptable and difficult to measure for less than 2 GHz. Here we purposefully do not consider GPU support in order to evaluate the potential for using lower cost commodity servers for these applications. However, in Section 7, we examine the case of servers with GPU support.

TABLE 4: Virtual-desktop specifications

Abbreviation	Description
1C2R	1 vCPU (2100 MHz) 2 GB vRAM
2C2R	2 vCPU (4200 MHz) 2 GB vRAM
3C2R	3 vCPU (6300 MHz) 2 GB vRAM
3C3R	3 vCPU (6300 MHz) 3 GB vRAM
3C4R	3 vCPU (6300 MHz) 4 GB vRAM

We limited the maximum specification to 3 vCPU (6300 MHz) with 4 GB vRAM as we expected relatively low resource consumption for the video-playback task.

4.3 Remote Display Protocols

Teradici PC-over-IP protocol (PCoIP) and Blast Extreme display protocol (Blast Extreme) are two high-performance display protocols developed to deliver virtual desktops to end-user devices. PCoIP uses a lossless codec to deliver end users high-quality desktops via zero clients, whereas Blast Extreme takes advantage of the thin-client computing approach by enabling audio-video buffering at the client side. In this study, we measured and compared the performance of the lossless zero-client protocol with the thin-client protocol using two codecs on residential use, as shown in Table 5.

TABLE 5: Remote display protocols

Abbreviation	Description
PCoIP	Teradici PC-over-IP protocol
Blast JPEG/PNG codec	VMware Blast Extreme Display Protocol
Blast H.264 codec	VMware Blast Extreme Display Protocol

4.4 Media Player

We used Media Player Classic (MPC) (packaged in K-Lite Codec Pack), which is a lightweight open-source media player designed for Microsoft Windows. MPC provided most options and features expected for modern videos while consuming less processor resources than other open-source media players. Among all video renderers, we selected the basic Overlay Mixing Renderer to avoid unnecessary virtual-desktop computational overhead, and used DirectSound as the audio renderer.

4.5 Metrics

Different Metrics were applied to measure the media quality delivered to the client and the overall system performance. The metrics can be divided into the following categories: (i) video quality, (ii) multimedia quality, and (iii) resource consumption.

4.5.1 Video quality

4.5.1.1 User-received frames per second (%)

Frames per second (FPS) is the frequency at which consecutive images (or frames) appear on a display. We defined the user-received FPS in percentage (recv-FPS(%)) as the ratio of the average rate of user-received frame updates during a complete run of video playback to the average FPS of the original encoded video:

$$\textit{recv-FPS(\%)} = \frac{Average\ UserReceived\ FPS}{Original\ Video\ FPS} \tag{1}$$

4.5.1.2 Received PSNR and received SSIM

PSNR and SSIM are two commonly used full-reference metrics for compression codecs. However, it is usually hard to use any full-reference metrics for videos that are transmitted over a network in a virtual-desktop environment, due to lost frames. To incorporate PSNR and SSIM into our measurements for the video quality, we defined two new metrics, received PSNR (recv-PSNR) and received SSIM (recv-SSIM), which depend on only user-received frames. We summarize our process for calculating recv-PSNR and recv-SSIM as follows:

a) We annotate each frame with its frame number, and extract all labeled frames as reference.
b) We take fast screenshots at the client when playing annotated videos via virtual desktops.
c) We find the corresponding original frame from reference based on each frame number in our captured screenshots, and calculate recv-PSNR and recv-SSIM for each captured screenshot. Note that this process was automated. Full details are described in Baseline Application Performance.

No noticeable measuring overhead was observed during experiments. We further note that this methodology could be applied in a local environment. The accuracy of this methodology is illustrated in Section 5.1.2 by performing evaluation on video streaming locally.

4.5.1.3 Playback duration

Video playback was observed to occur in slow motion when the virtual desktop had limited computing resources. Therefore, we used the actual playback duration of a complete video as one metric for video quality because any extension in duration indicates that the allocated resource is not sufficient for the virtual desktop to process and deliver videos in time.

4.5.2 Multimedia quality

4.5.2.1 ViSQOLAudio

The quality of the user-received audio signal is measured by the objective ViSQOLAudio [29] metric where the magnitudes of the reference and the test spectrograms are calculated using a 32-band Gammatone filter bank. The output similarity is then mapped into a MOS value within the range of 1 to 5.

4.5.2.2 Audio-visual sync

The synchronization of audio and video when playing a multimedia file is evaluated in terms of audio-visual time shift. The measurement process can be summarized as follows:

a) We record both the video and audio of each complete multimedia file viewed on the client.
b) We extract the timestamp of each audible marker (beep) from the recorded audio.
c) We find the nearest extracted frame and the corresponding moving bar position based on the timestamp of each audible marker.
d) We calculate the audio-visual shift as the difference between the extracted bar position in step c) and that of the ground truth from the original multimedia file.
e) We convert the audio-visual shift into a time-based scale based on the calibrated scale on which the bar position was measured.

The application of this methodology in a local environment would be restricted by the recording FPS of the recorder.

4.5.3 System resource monitoring

System resource consumption is included as metrics of virtual-desktop capabilities that yield user-received application performance. The metrics are listed as follows: (i) CPU usage, (ii) Memory usage, (iii) Network usage, (iv) GPU usage.

5 User-received Media Quality Evaluation

An objective evaluation of the virtual-desktop computing model was performed by measuring the quality of media delivered to the client. Experimental data was collected from the client device, from within the server, and from packet traces between the server and the client.

5.1 Baseline Application Performance

An experiment was first conducted to identify differences in user-received application quality with distinctive display protocols, varied virtual-desktop specifications, and various types of videos.

5.1.1 Setup and Automation

A high-end client device, HP EliteDesk 800 G3 Desktop Mini Business PC, Intel(R) Core(TM) i7-8700T CPU @ 2.40 GHz and 16 GB RAM with the Windows 10 Operating System, was used as the client device to access virtual desktops hosted on the server Anvil to avoid any performance limitation caused by the client machine. The HP EliteDesk was directly connected to the Anvil Connection Server through a university network environment with a round-trip time of less than 1 ms. The device was connected to a monitor that supports 1920×1080 display resolution.

In the baseline experiment, we used the three input videos, five virtual-desktop specifications, and three display protocols as shown in Table 2, Table 4, and Table 5. To obtain the quality of the delivered videos, we used the four video-quality metrics:

- recv-FPS
- recv-PSNR/recv-SSIM
- Playback duration

Measuring user-received FPS requires taking measurements on the client machine display. We accomplished this using TCBench [30], a sub-program of the VirtualGL open-source program, to capture the rate of pixel change in a specified region in the client window. We selected an area at the center of the window for capture. For each complete run of video playback, we took the average recv-FPS. The experiment was repeated 50 times for each video playback.

To automate the process of measuring recv-PSNR/recv-SSIM, an AutoIt [31] script was used to send screen-capture commands and save screenshots in the BMP file format at the client device. The starting time and time interval of screen-capturing commands varied for each complete run of video playback. We captured 20 screenshots for each run, and each experiment was repeated 20 times. This process yielded 400 received frames for each video playback.

We used a Convolutional Neural Network (CNN) to automate the frame-matching process based on the frame number marked at the upper right corner of each frame. The CNN was built using the ResNet architecture, and was trained using a dataset consisting of 10,000 30×30 grayscale blurred digit images ranging from 0 to 9. We obtained those digit images by cropping annotated video frames. Among those 10,000 images, 2,000 were left-shifted, 2,000 were slightly right-shifted, 2,000 were slightly left-rotated, 2,000 were slightly right-rotated, and 2,000 remained unchanged. All digits in those images were of the same font and size as that of the video annotations. All digit images were blurred prior to training as degradation was expected in user-received frames. We intended to obtain an over-fitting model since the model would only be used to identify digits that are exactly the same font and size as those of digits in the training dataset. The CNN model proved to be so as it achieved perfect 1.0 accuracy, 1.0 precision, and 1.0 recall. In the frame-matching process, for each captured screenshot, each digit was cropped out, blurred, and then identified by the CNN. Recognized digits of each frame were then put back in order to form the original frame number.

The packet trace for a complete run of video playback was obtained using an AutoIt script to emulate a user's behavior when playing videos via the virtual desktop. Traffic capture was enabled at the client device prior to executing the script in the virtual desktop. A marker packet was sent prior to the start of each video playback. The overall process can be summarized as follows:

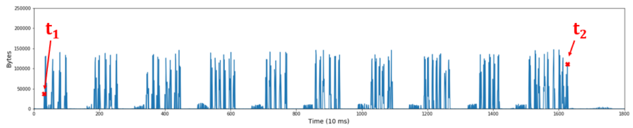
a) Start traffic capture at the client device.
b) Execute the script in the virtual desktop
c) The script opens Media Player Classic, goes to "Quick Open File..." menu, types the video name in the "File name" field, clicks on "Open", and immediately sends a marker packet to the client
d) Stop traffic capture when video playback ends.

The captured packet trace was analyzed using a Python script, following these steps:

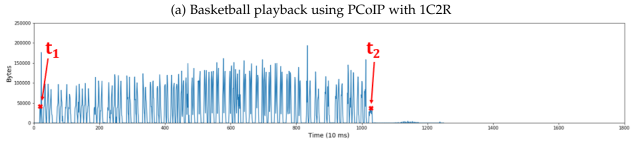
a) Filter out background traffic irrelevant to screen updates based on the packet size.
b) Find the timestamp of the marker packet t_0.

c) Find the timestamp t_1, that corresponds to the first packet of the continuous screen updates after t_0, and the timestamp of the last packet t_2
d) Calculate the actual playback duration as $t_2 - t_1$.

For example, Figure 4 shows t_1 and t_2 extracted from a packet trace of playing the Basketball video using PCoIP with different virtual-desktop specifications. In the experiment, each video playback was repeated 20 times, generating 20 playback duration measurements.



(a) Basketball playback using PCoIP with 1C2R



(b) Basketball playback using PCoIP with 2C2R

Fig. 4: Timestamp extracted from packet traces

5.1.2 Performance Benchmark

Performance benchmarks of recv-FPS, recv-PSNR, and recv-SSIM were obtained to show the accuracy of the measurement tools. Using the high-end client device, we played three videos locally via MPC, and measured user-received video quality applying the same automation techniques as previously discussed. Results are shown in Table 6.

TABLE 6: Performance benchmark

	Basketball	Squirrel	Transformer
True FPS	25/1	25/1	24/1
Avg. recv-FPS	23.77	24.26	22.98
Avg. recv-FPS(%)	95.08	97.04	95.75
recv-PSNR (dB)	43.00	43.09	45.48
recv-SSIM	0.9957	0.9960	0.9937
user-received quality	Excellent	Excellent	Excellent

The slight difference between captured FPS and the source video FPS may come from the usage of the TCBench tool as it captures the rate of pixel change in one specified region rather than the entire client screen. Choices of the capturing region will cause a small variation in the resulting recv-FPS. The difference could also come from limitations in the MPC capabilities.

Values of recv-PSNR and recv-SSIM for local video playback are related with the selection of video renderer. Though our choice of Overlay Mixing Renderer may lead to small degradation in frame quality, the overall user-received video quality is excellent as we expected.

5.1.3 Results

Figure 5 shows the user-received video quality with the high-end user device. The output is organized so that each column shows results of one display protocol, and each row shows the comparison of each metric output among the three display protocols. In general, PCoIP delivers frame updates with the lowest recv-FPS but the highest frame quality, while the Blast H.264 codec provides frame updates with the highest recv-FPS but lower frame quality as compared to PCoIP, especially for the high-motion video. The Blast JPEG/PNG codec, on the other hand, yields videos with the recv-FPS in between PCoIP and Blast H.264, while producing the lowest frame quality. Across different video types, the three display protocols show the highest performance for the static video, Transformer, and the lowest performance for the high-motion video, Basketball. This situation occurs because high-motion videos involve much more frequent changes in pixel values among frames, leading to more computational overhead at the server.

Though PCoIP sends frame updates more frequently with increased virtual-desktop resources, large variance in recv-FPS is observed in PCoIP for the three videos. For Blast JPEG/PNG codec and H.264 codec, recv-FPS increases with virtual-desktop resources, and the corresponding variance decreases. When the virtual-desktop resources rise from 3C3R to 3C4R, little improvement in performance is observed for all three display protocols.

PCoIP delivers frames of the highest and most consistent quality with values very close to the benchmark recv-PSNR and recv-SSIM for the most static video, Transformer. Interestingly, unexpected degradation in recv-PSNR and recv-SSIM is observed for both Blast Extreme protocols with the increase of virtual-desktop resources, especially for Basketball, the high-motion video. For 3C3R and 3C4R virtual desktops using Blast JPEG/PNG codec and Blast H.264 codec, we observed a larger variance in recv-SSIM. One possible interpretation is that when frame updates are frequent and computing resources are very limited, Blast Extreme protocol reduces the rate of delivering frame updates to provide relatively good frame quality. With the increase of available computing resources, Blast Extreme balances frame-delivery FPS and frame quality so that it reduces the computational overhead for frames. Overhead is reduced by reducing quality of delivered frames occasionally to maintain relatively high FPS delivered.

Results for playback duration, as shown in the last row of Figure 5, suggest that 1 vCPU (2100 MHz) is not capable of processing high-motion videos. When increasing the assigned processor to 4200 MHz (2 vCPU), all display protocols are able to deliver videos without a slow-motion effect.

As the human visual system can process 10 to 12 images per second while perceiving them as individual images, we consider 12 FPS as the minimum requirement for a video to be perceived as motion by the observer. That is 48% recv-FPS for Basketball and Squirrel, and 50% for Transformer, as represented by dashed lines in recv-FPS(%) in Figure 5. We denote thresholds of excellent and good frame quality by two dashed lines for both recv-PSNR and recv-SSIM, respectively, as suggested by Table 1. We thus conclude that PCoIP provides higher frame quality but requires more virtual-desktop processing resources as compared to Blast Extreme protocol. It is limited in frame-delivering rate by the high motion video even with sufficient computing resources. Whereas Blast Extreme protocol appears to strike a balance between the frame quality and updating rate, sacrificing a small amount of frame quality to ensure high delivering rate for all videos when given adequate amount

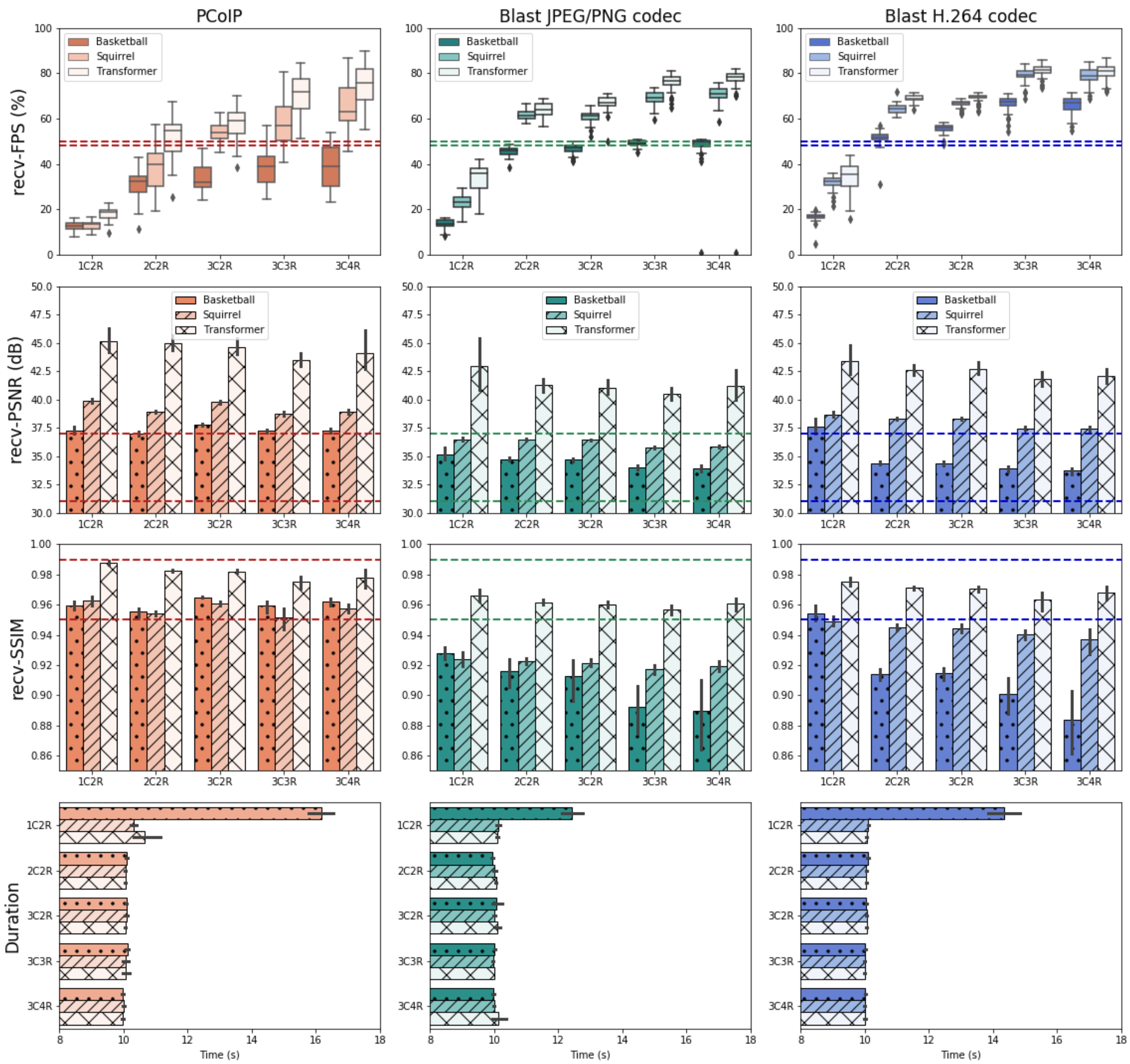


Fig. 5: User-received video quality using a high-end user device

of processing resources. Further, reducing virtual-desktop resources to 1 vCPU is insufficient for any videos.

5.2 Application Performance with Audio Inclusion

Applications, services, and demand for digital multimedia communication have increased significantly in recent years [32]. Increased usage of multimedia applications has been accompanied by expectations for better Quality of Experience. Therefore, we evaluate the quality of multimedia delivered by edge cloud from the aspects of audio quality and audio-visual synchronization as supplements to our previous assessment of the video quality.

5.2.1 Setup and Automation

An HP EliteDesk 800 G3 was used as the client device that was directly connected to the Anvil Connection Server through Ethernet with an RTT less than 1 ms. We used four multimedia files for the audio-quality measurement, as shown in Table 3, and a specialized multimedia file for the audio-visual synchronization evaluation, as described in Section 4. We used four virtual-desktop specifications: (i) 1C2R, (ii) 2C2R, (iii) 3C2R, and (iv) 3C3R as described in Table 4, and three display protocols as shown in Table 5. Metrics used are listed as follows:

- ViSQOLAudio
- Audio-visual sync

Degraded audio files were obtained by using Audacity [33] at the client device to record user-received audio signals. Audacity is an open-source audio recording application that runs on multiple platforms. An AutoIt script was developed to automate the measurement process, which can be stated as follows:

a) Start Audacity prior to multimedia playback at the

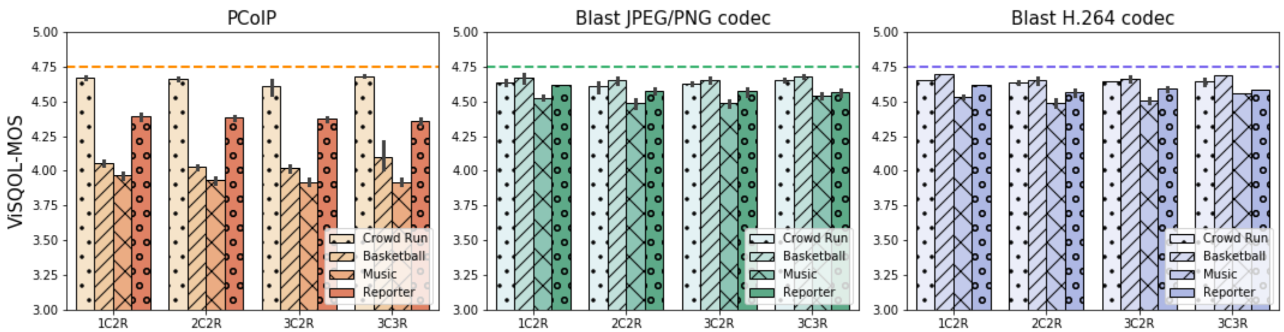


Fig. 6: Audio quality obtained using a high-end user device

client device.

b) Play the multimedia file using the virtual desktop.
c) Stop audio recording when the playback ends.
d) Cut off any silent period from the recorded audio file using -30 dB as the threshold.
e) Use ViSQOLAudio to generate the ViSQOL-MOS score by comparing the processed recorded audio file with the original audio file.

In the experiment, each multimedia playback using one display protocol with one virtual-desktop specification was repeated 20 times, and thus generating 20 ViSQOL-MOS scores.

To measure audio-visual synchronization, we used the specialized multimedia file as described in Section 4. OBS studio [34], an open-source streaming recording program, was applied to record both video (at 30-FPS as the highest FPS achieved by any of the display codecs in Baseline Application Performance is below 30) and audio (at 48 kHz) of multimedia playback. We ran an AutoIt script similar to the previous audio recording to automate our recording process. Another Python script was developed for data processing after extracting audio signals from the recorded file, with steps summarized below:

a) Call functions from Librosa library [35] to obtain timestamps of extracted audio signals.
b) Find the nearest video frame and the corresponding moving bar position for each extracted audio timestamp.
c) Calculate the audio-visual shift as the difference of the moving bar position between the recorded file and the reference for each audio signal.
d) Convert the position shift to a time-based shift.

For each multimedia playback, the experiment was repeated 25 times. As each multimedia playback contains 10 audio signals, a total number of 250 shifts were obtained for each display codec per virtual-desktop specification. We also performed benchmark test by measuring the audio-visual shift of playing the multimedia locally on the client device. The median value of the audio-visual time shift obtained from the benchmark test was subtracted from the experimental data.

5.2.2 Results

Figure 6 shows the quality of audio signals delivered by edge cloud virtual desktops. Support vector regression is used to map similarity scores to ViSQOL-MOS scores in the range of 1 to 5. For audio quality, the maximum ViSQOL-MOS value is expected to be 4.75. Blast JPEG/PNG codec and Blast H.264 codec deliver all four types of audio signals with excellent quality. A degradation in performance is observed for increasing audio frequency range of the audio signals when using PCoIP. A decrease in the audio quality is also observed when audio signals are accompanied by high-motion video content. Further, few differences in the audio quality are found among various virtual-desktop specifications.

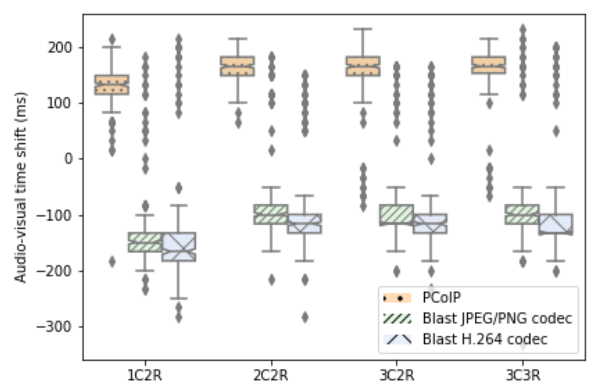


Fig. 7: Audio-visual synchronization evaluation using a high-end user device

Figure 7 shows the audio-visual time shift after calibration using three remote display codecs with different virtual-desktop specifications, where positive values indicate that video is advanced with respect to sound. Though no obvious differences in time shifts are observed for PCoIP with different virtual-desktop specifications, low computing resources (i.e., 1C2R) introduce larger time shifts for both Blast H.264 and JPEG/PNG codec. In general, PCoIP delivers audio signals late, while both Blast H.264 and JPEG/PNG codec send audio signals prior to the vision with an overall smaller time shift. This is expected since unlike PCoIP, Blast enables audio-video buffering at the client side. It is noteworthy that all positive outliers observed for two Blast codecs have values close to PCoIP, and are related to the first audio signals in multiple playbacks, for which our interpretation is that the client has an empty buffer at the beginning of each multimedia playback.

5.3 Network Impact

We next vary network parameters to determine the minimum conditions necessary for edge cloud virtual desktops to deliver videos with a good user-received quality. We cover a wide range of real-world situations showing virtual-desktop performance with latency and packet loss rate from extremely low to comparatively annoying. Since edge computing moves the processing power closer to residential users in our case, we do not consider network bandwidth as one of the performance bottlenecks in this study. The actual network usage of delivering videos via edge cloud is shown in Figure 12.

5.3.1 Setup and Automation

An HP EliteDesk 800 G3, directly connected to the Anvil Connection Server was used as the high-end client device for virtual-desktop performance evaluation with network control. We chose the minimum virtual-desktop specification with the highest performance using Blast H.264 codec in Baseline Application Performance, 3C3R, and the high-motion video, Basketball, as shown in Table 2, to investigate network impact on user-received video quality.

TABLE 7: Network Control Parameters

Fixed latency	Normally distributed latency
10 ms	$\mathcal{N}(10\ ms, 1\ ms)$
20 ms	$\mathcal{N}(20\ ms, 2\ ms)$
50 ms	$\mathcal{N}(50\ ms, 5\ ms)$
100 ms	$\mathcal{N}(100\ ms, 10\ ms)$
Periodic packet loss	**Gilbert-Elliott loss**
0.10%	$peak\ loss = 0.10\%$
0.50%	$peak\ loss = 0.50\%$
1.00%	$peak\ loss = 1.00\%$
2.00%	$peak\ loss = 2.00\%$
5.00%	$peak\ loss = 5.00\%$

Four fixed and four normally distributed latency values were applied for the evaluation, as shown in Table 7. Specifically, we picked 10 to 100 ms latency to demonstrate virtual-desktop performance under the upcoming fifth-generation (5G) wireless cellular network and the current fourth-generation (4G) wireless cellular network conditions, as suggested by Chen *et al.* [36]. For packet loss rate, we chose to compare performance of periodic loss and the Gilbert-Elliott loss. The Gilbert-Elliott model [37], is a 2-state Markov chain that provides estimation for real-time services on the Internet by enabling burst loss. In the Gilbert-Elliott loss settings, we assigned 0.00% loss rate to the good state (state G) with a 10.0% transition probability, and assigned 0.10%, 0.50%, 1.00%, 2.00%, 5.00% as the bad-state (state B) peak loss rate with a transition probability of 90.0%, respectively. The time slot for transitions were set as 10 milliseconds. We used these metrics:

- recv-FPS
- recv-PSNR/recv-SSIM
- Playback duration

We used the Network Emulator for Windows Toolkit [38], a software-based network-environment emulator. The emulator was executed in the virtual desktop, and network control was applied to the downstream (or outgoing) traffic. Computational resources consumed by this emulator could be ignored as compared to that of running MPC for video playback. Also, we took advantage of the automation techniques described in the previous Baseline Application Performance.

5.3.2 Results

Figure 8 shows the mean values of recv-FPS, playback duration, recv-PSNR, and recv-SSIM with different latency. With fixed latency, while recv-FPS remains unaffected, degradation in both recv-PSNR and recv-SSIM is observed. Though recv-PSNR drops when fixed latency goes beyond 50 ms, the mapped video quality remains good within 100 ms of fixed latency. As a comparison, video files show stronger dependency on fixed latency for the recv-SSIM metric as it decreases when fixed latency starts to increase from 0, leading to degradation in the mapped video quality with fixed latency greater than 50 ms. Playback duration remains unaffected until fixed latency goes beyond 50 ms, which reflects the fact that Blast Extreme takes advantage of audio-video buffering at the client side. Therefore, we conclude that with Blast H.264 codec, the performance of video playback delivered by edge cloud virtual desktops is preserved with fixed latency no greater than 50 ms.

With normally distributed latency, a decrease in recv-FPS, recv-PSNR, recv-SSIM, and an increase in playback duration are observed. Particularly, when the mean value of normally distributed latency goes beyond 20 ms, recv-FPS drops below 50% and the video delivered is no longer perceived as motion by the observer. A slow-motion effect becomes more serious as well due to the varied delay in frame delivery. Though the frame quality in PSNR remains good as recv-PSNR only drops from $34 - 35$ to 32, the frame quality in SSIM drops significantly from fair to poor with the increase of normally distributed latency. In general, the acceptable highest mean value of normally distributed latency is no more than 20 ms.

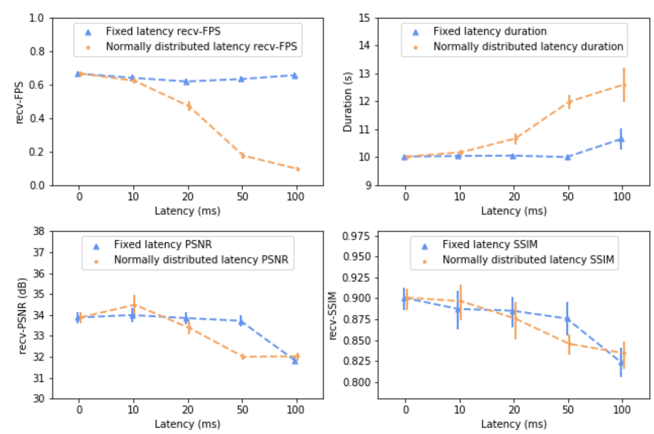


Fig. 8: Video quality with injected latency

Figure 9 shows the user-received video quality with packet loss. When packet loss rate increases, for both periodic loss and Gilbert-Elliott loss, the largest impact is on recv-FPS and playback duration. With periodic loss, a sharp drop in recv-FPS and a rapid burst in playback duration are observed when packet loss rate increases beyond 2%. With Gilbert-Elliott loss, recv-FPS drops below 50% for a 2% peak loss rate. It is then concluded that a packet loss rate with peak loss below 2% is required for cloud-based virtual desktops to deliver videos with good quality.

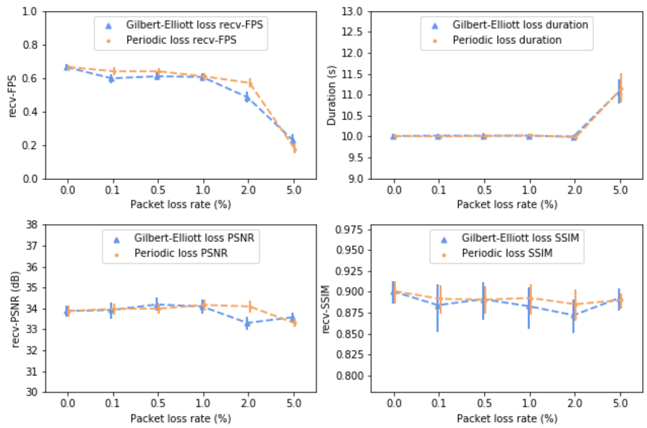


Fig. 9: Video quality with injected packet loss rate

A relatively large variance is observed in recv-SSIM with either latency or packet loss rate. This is consistent with our previous observation in Baseline Application Performance where Blast H.264 codec drops frame quality occasionally to maintain a relatively high delivered FPS when provided with sufficient computing resources.

5.4 Residential-deployment Emulation Experiments

Switching from standard personal computers to affordable thin clients using VDIs over edge clouds is appealing for the potential to provide quality computing experiences at favorable costs. A Chromebook differs from a standard personal computer because the Chromebook is designed to perform a variety of tasks taking advantages of the Google Chrome browser with most applications and data residing in the cloud. The low cost of the hardware, software, and upkeep make Chromebook an affordable residential computing platform. However, Chromebooks traditionally suffer from poor computing performance due to the limited thin client computational capabilities. Thus, the addition of edge cloud based VDI has the potential to overcome this shortcoming by utilizing the cloud based processing. Similar arguments can be made for other thin-client devices such as Microsoft Surface Go. Surface Go is a 10-inch 2-in-1 detachable laptop, or tablet, equipped with Windows 10 Home in S-mode. As compared to ChromeOS, the Windows operating system may provide users more flexibility in task performance. We also note that zero clients offer similar cost savings and performance using VDI without the benefit of a local operating system. Zero clients have been studies elsewhere [14]. This section quantifies the performance of the edge cloud virtual-desktop solution using two thin-client devices, a Chromebook and a Surface Go. Table 8 shows the corresponding specification and processor/integrated GPU capability as compared to the high-end desktop. As a side note, the Surface Go is installed with standard Windows 10 Home rather than S-mode. We reference processor and GPU scores from Geekbench Browser [39].

TABLE 8: User device specification

	High-end desktop
Manufacturer	HP
Processor	Intel(R) Core(TM) i7-8700T CPU @ 2.40GHz
Number of cores	6
Number of threads	12
RAM (GB)	16
Processor score (multi-core)	4949
Professor graphics	Intel(R) UHD Graphics 630
Professor graphics score	4903
Operating system	Windows 10 Pro
	Chromebook
Manufacturer	Acer
Processor	Intel(R) celeron(R) CPU N2840 @ 2.16GHz
Number of cores	2
Number of threads	2
RAM (GB)	4
Processor score (multi-core)	373
Professor graphics	Intel(R) HD Graphics 500
Professor graphics score	1222
Operating system	ChromeOS
	Surface Go
Manufacturer	Microsoft
Processor	Intel(R) Pentium(R) CPU 4415Y @ 1.60GHz
Number of cores	2
Number of threads	4
RAM (GB)	4
Processor score (multi-core)	885
Professor graphics	Intel(R) HD Graphics 615
Professor graphics score	3539
Operating system	Windows 10 Home

5.4.1 Setup and Automation

The Acer Chromebook and Microsoft Surface Go were used as VDI client devices to access virtual desktops hosted on the server Anvil. Both devices were connected to the Anvil Connection Server via the university wireless connection with a round-trip time less than 5 ms. As limited by the original display resolution, both thin-client devices were connected to external monitors that support 1920×1080 resolution via HDMI cables to enable 1080p video playback.

We study the application performance of using a Chromebook as the VDI interface with different display protocols and virtual-machine specifications. We further compare the user-received video quality of using thin-client devices with the result we obtained in Baseline Application Performance using 3C3R, the minimum specification with the highest performance. We used three videos as described in Table 2.

5.4.1.1 Chromebook

Due to ChromeOS limitation, only the following metrics were used to evaluate the video quality:

- recv-PSNR/recv-SSIM
- Playback duration

To measure recv-PSNR/recv-SSIM, 20 screenshots were taken for each complete run of video playback. The experiment was repeated 20 times and a total number of 400 screenshots were obtained per video playback. No noticeable overhead was observed during screenshot capturing. Captured frames were then used to calculate recv-PSNR and recv-SSIM using the same automation techniques as described in Baseline Application Performance.

Playback duration was measured using an AutoIt script, which records the starting time and ending time of MPC. The measurement was repeated 20 times per video playback.

5.4.1.2 Surface Go

As Surface Go runs the Windows 10 operating system, the measurement tools and scripts developed in Baseline Application Performance also apply to Surface Go. Therefore, all three video-quality metrics were included:

- recv-FPS
- recv-PSNR/recv-SSIM
- Playback duration

We used the same automation techniques described in Baseline Application Performance to quantify user-percv FPS and recv-PSNR/recv-SSIM. No noticeable overhead was observed during screenshot capturing. However, to measure playback duration, we used the AutoIt script, as described above, to record the starting and ending time of MPC execution in the virtual desktop rather than capturing the corresponding traffic trace at the client device since we observed that running network capturing tools on the thin-client significantly increases the processor workload, introducing a slow-motion effect to video playback.

5.4.1.3 Calibration on Playback Duration

Given that the playback duration obtained by running the AutoIt script includes not only the actual playback duration, but also the execution time of starting the MPC program and the time consumed to load the video file, we perform calibration on the obtained data, which is summarized as follows:

a) We measure playback duration by running the AutoIt script using the high-end desktop.
b) We estimate the extra processing time by comparing the duration obtained from the above step to that in Baseline Application Performance.
c) We calculate the true playback duration using thin-client devices as the difference between the obtained duration and the extra processing time.

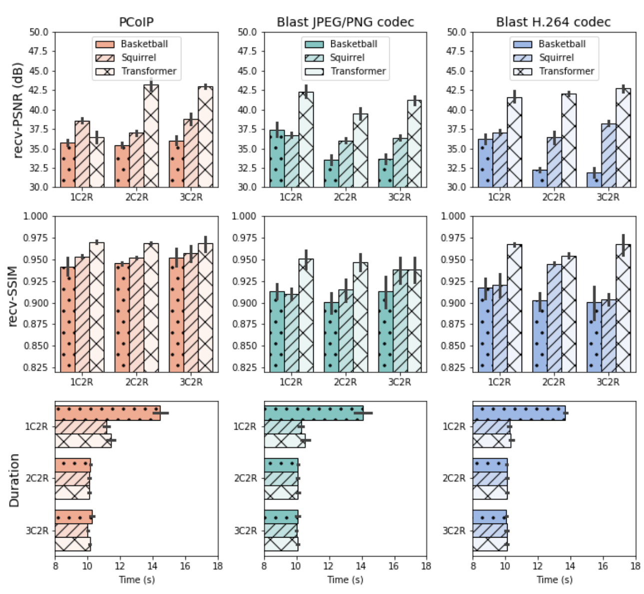


Fig. 10: User-received video quality using a Chromebook client device

5.4.2 Results

Figure 10 shows the video quality with the 1C2R, 2C2R, and 3C2R vitual-desktop specification using a Chromebook. Similar to Baseline Application Performance, we conclude that PCoIP provides users with the highest frame quality, and the 1C2R virtual-desktop specification is not able to handle video playback tasks as it introduces serious slow-motion effect. Comparing to the result obtained in Baseline Application Performance, small degradation in the frame quality is observed with low specifications, suggesting that a high-end user device may provide users with better performance when limited computing resources are allocated for the virtual desktop.

Figure 11 shows the comparison of user-received video quality using different client devices with 3C3R virtual-desktop specification. Overall, results from three client devices are consistent where PCoIP delivers frame updates with the lowest FPS but the highest frame quality, and Blast H.264 codec sends frame updates most frequently but with lower quality as compared to PCoIP. It is worth noting that the large variance in recv-SSIM, especially for Basketball, was again observed using all user devices.

While using a high-end client device may result in more stable performance, no remarkable difference in video quality is observed among three user devices. Though the video quality delivered by PCoIP shows a slight drop in values with thin-client devices, the mapped quality based on Table 1 keeps almost the same with that using the high-end user device.

We conclude that the performance of edge cloud virtual desktops using a thin-client model shows little to no dependence on the capability of user devices when sufficient computing resource is allocated for virtual desktops. In the study, using a Chromebook or a Surface Go as the user device obtains nearly the same video quality as that of using a high-end user device with the 3C3R virtual-desktop specification.

6 Enabling Resource Sharing Using Virtual Desktops

We next determine how to maximize the number of users that share edge cloud resources while maintaining good performance. We make this determination by experimenting with resource allocation shares in the edge cloud virtual-desktop service. The resource-allocation sharing functionality is supported by VMware vSphere. Shares specify the relative priority of a virtual desktop. A virtual machine is entitled to consume more resources if it is assigned with higher shares when there is resource over-commitment among virtual desktops in the same resource pool.

6.1 Setup and Automation

A resource pool containing two virtual desktops, VM 1 and VM 2, was created and assigned 3 vCPU (6300 MHz) and 6 GB vRAM. Each virtual desktop was initially allocated with 3 vCPU (6300 MHz) and 3 GB vRAM. Those values were determined so that when only one virtual desktop is powered on, it could be provided as much processor power as it was assigned. However, when the two virtual desktops are both powered on at the same time, the total available processing power would be limited to the amount allocated to the resource pool. Therefore, processor over-commitment

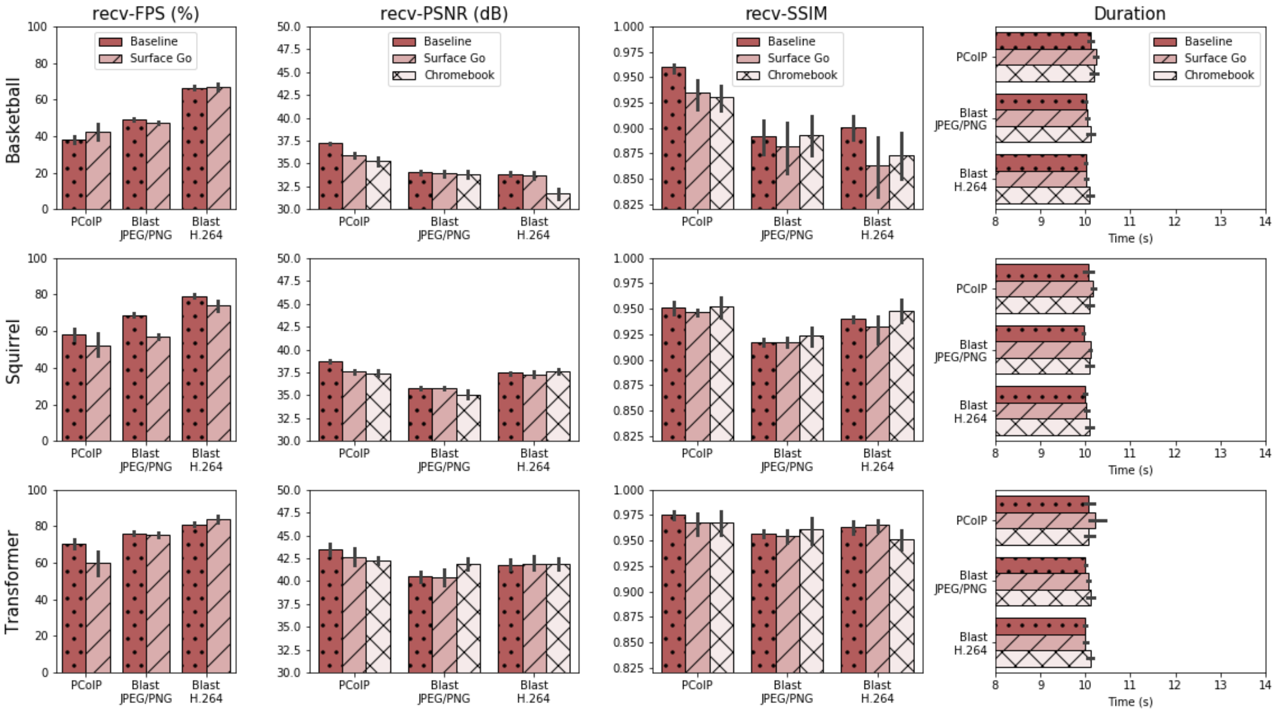


Fig. 11: User-received video quality using a high-end user desktop, a Surface Go, and a Chromebook as the client device with the 3C3R virtual-desktop specification

could occur between the two virtual desktops. In the experiment, two machines, an HP EliteDesk 800 G3 and a Microsoft Surface Pro 6, Intel(R) Core(TM) i5-8250U CPU @ 1.60 GHz and 8 GB RAM with Windows 10 Operating System, were used as two client devices to connect to VM 1 and VM 2 via Blast H.264 codec, respectively.

TABLE 9: Phases of user-behavior simulation

	VM 1	VM 2
Phase I	Basketball playback	Notepad typing
Phase II	Basketball playback	YouTube watching
Phase III	Basketball playback	Basketball playback
Phase IV	idle	Basketball playback

Two AutoIt scripts were executed in virtual desktop VM 1 and VM 2 simultaneously to simulate four phases of user behavior, as shown in Table 9. Each phase lasted 200 seconds. Performance evaluation was carried out on the high-end client device, HP EliteDesk 800 G3, for VM 1 only. Metrics used in this section are listed as follows where the resources consumed by virtual desktops were monitored and recorded in real-time by vSphere:

- Video quality
  - recv-FPS
  - recv-PSNR/recv-SSIM
- System resource monitoring
  - CPU usage
  - Memory usage
  - Network usage

To automate the measurement process, the Basketball video was played in the repeated mode for the entire evaluation period. TCBench was configured to continuously monitor a 5-second average FPS. Meanwhile, a screenshot of video playback was taken each 5.6 seconds. Both capturing techniques were activated from the beginning to the end of VM 1 video playback at the client. The experiment was repeated twice: once with the same shares specified for VM 1 and 2, and the other with high shares for VM 1 and normal shares for VM 2. As high and normal shares specify share values with a 2:1 ratio, we expected VM 1 to exhibit twice as much processor usage as that of VM 2 when CPU over-commitment occurrs.

6.2 Results

Figure 12 shows the resources consumed by VM 1 and VM 2 during the experiment, and the user-received video quality of VM 1. Tasks assigned to VM 2 in Phase I and II are not CPU-consuming. Therefore, no CPU over-commitment occurs in the first two phases, and two virtual desktops can use resources independently in response to the task-execution requirement. However, we observe a burst in the resource consumed by VM 2 when switching tasks, which corresponded to a drop in the resource consumed by VM 1 and results in a sudden decrease in recv-FPS. On the other hand, recv-PSNR and recv-SSIM remained unaffected.

Starting from Phase III, VM 2 switches to a video-playback task and starts to require more computational resources, leading to CPU over-commitment. When no preemption is given, VM 1 and VM 2 share the computational resource evenly. Degradation in recv-FPS is observed at the start of Phase III due to the decrease in the allocated resources. An increase in recv-PSNR happens at the start of Phase III with the decrease of allocated resources. This is consistent with our previous result when allocated resource becomes limited, Blast H.264 codec reduces frame-delivery rate to keep a relatively high frame quality. Similarly, though there is an occasional drop in recv-SSIM as observed in the

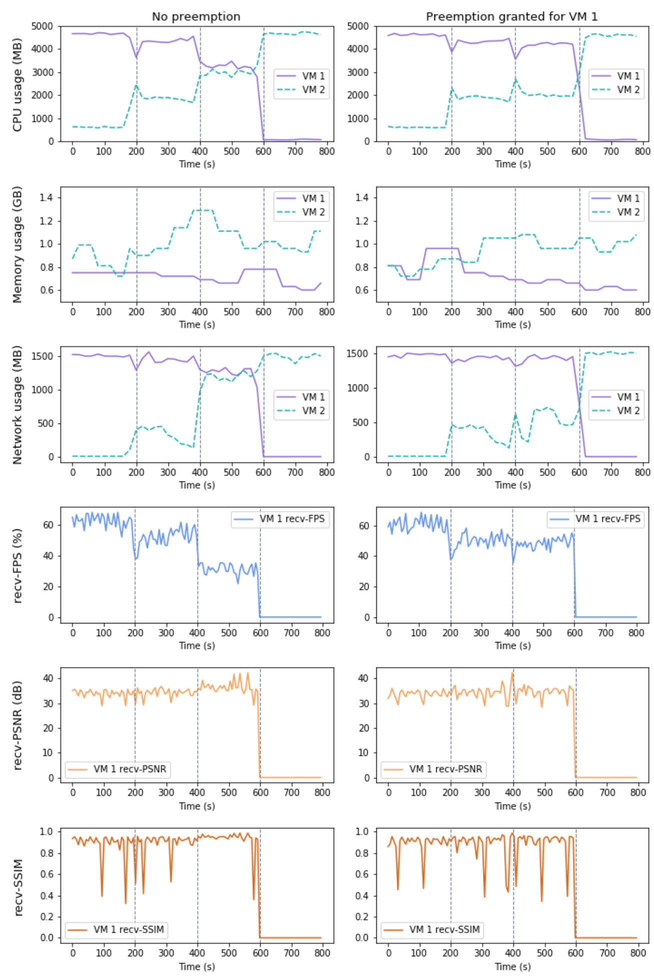


Fig. 12: Resource consumption and performance evaluation of resource sharing

baseline experiment, the overall recv-SSIM remains stable. In contrast, when preemption is granted for VM 1, VM 1 is able to consume the same amount of resources even when CPU over-commitment happens, and thus preserving the video quality delivered to the client.

In Phase IV, we show that when VM 1 becomes idle, no matter whether the preemption is given, VM 2 grabs as much resources as it is initially assigned to perform processor-consuming tasks.

We demonstrate only CPU-sharing in this study. Given that video playback is not memory-intensive, very limited memory resources would need to be initially assigned to the resource pool to trigger memory over-commitment among virtual desktops. In that case, VMware Memory Ballooning is activated from the moment when virtual desktops are powered on, resulting in a large degradation in virtual-desktop performance. Therefore, we conclude that allocating memory below our minimum specification in order to trigger memory allocation sharing is unnecessary for low memory-consuming tasks.

The results suggest that resource allocation shares can be scaled to a much larger resource pool containing dozens of users to maximize the number of users that share edge cloud resources, and that preemption should be given to users who need to to perform high resource-consuming tasks to preserve good user-received performance.

7 Impact of Edge-Cloud Server Capabilities

A commodity datacenter sever differs from those designed for commercial virtual-desktop offerings because the former typically has no GPU. Therefore, we compared virtual-desktop capabilities on graphics processing for video playback tasks with GPU-virtualization disabled and enabled. We conducted measurements on server image-rendering performance. We further addressed the issue of balancing cost and performance of edge cloud residential virtual-desktop computing.

7.1 Setup and Automation

A Dell R730 was configured to function as the edge-cloud server with VMware ESXi 6.5, 12 CPUs $\times$ Intel(R) Xeon(R) E5-2670 v3 @ 2.30GHz, and 64 GB RAM. The server was equipped with an Nvidia GRID K2 that supports GPU hardware virtualization and allows multiple users to share a single GPU using virtual-desktop solutions. Measurement was conducted in two modes:

- GRID K2 disabled mode: no GPU virtualization
- GRID K2 enabled mode: set Nvidia GRID K2 as the primary GPU for VSGA

Three videos, as shown in Table 2, and five virtual-desktop specifications, as shown in Table 4, were included in server capability evaluation. Differing from the previous experiments, virtual desktops were accessed via VMware remote console and the measurement was taken on the local server given that not only can the remote display protocol affect the video quality but also the rendering processing at the server side. Therefore, no remote display protocols were involved in the measuring process. Metrics used to evaluate server capabilities are:

- Video quality
  - recv-FPS
  - recv-PSNR/recv-SSIM
- System resource monitoring
  - CPU usage
  - GPU usage

The MPC video render filter was applied to measure recv-FPS for each complete run of video playback. During playback, a Python script that cooperates with PyAutoGUI [40] and psuti [41] was used for screen capturing and CPU monitoring, respectively. Measurements for recv-FPS and CPU/GPU consumption were repeated 5 times per video playback, and for each video, 20% of frames were captured for frame quality evaluation. The same techniques as described in Section Baseline Application Performance was used to quantify recv-PSNR and recv-SSIM. Further, Unigine Heaven Benchmark [42] was used to evaluate GPU-virtualization performance of the 3C4R virtual desktop with 1920×1080 resolution and Direct3D9 API.

TABLE 10: Unigine Heaven Benchmark for the 3C4R virtual-desktop specification

GRID K2 disabled	Min. FPS	Max. FPS	Avg. FPS
	1.1	1.6	1.3
GRID K2 enabled	Min. FPS	Max. FPS	Avg. FPS
	8.3	134.3	45.3

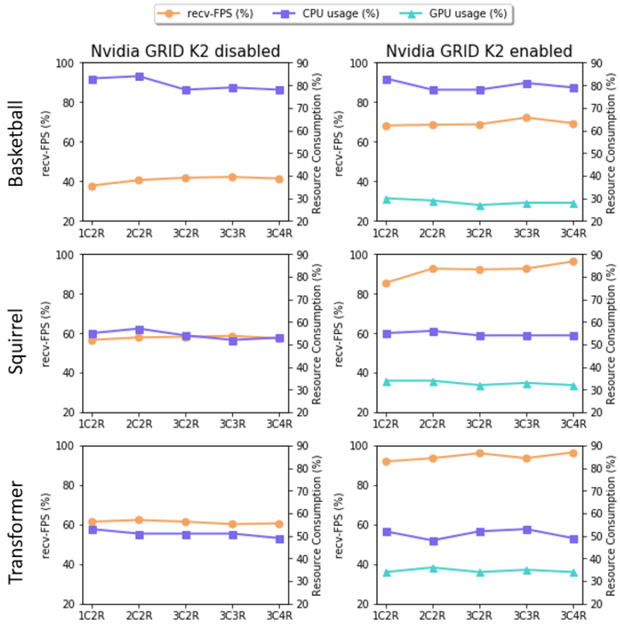


Fig. 13: recv-FPS(%) and resource consumption of virtual-desktop video playback with GPU-virtualization disabled and enabled

7.2 Results

Table 10 shows Nvidia GRID K2 GPU-virtualization performance. By enabling GPU-virtualization, virtual desktops are capable of processing frames at a much higher rate, with an average FPS reaching nearly 35 times as that of disabling GPU-virtualization.

Figure 13 shows the recv-FPS and corresponding CPU/GPU consumption of video playback. Though CPU usage remains almost the same, enabling the Nvidia GRID K2 GPU-virtualization sharply raises the FPS for all virtual-desktop specifications. Figure 14 shows the frame quality produced by various virtual-desktop specifications. A decrease in the frame quality, especially in terms of recv-PSNR, is observed when enabling Nvidia GRID K2 GPU virtualization, though the mapped quality remains excellent for all three videos.

We conclude that with GPU-virtualization enabled, even the virtual desktop with the lowest specification, i.e., 1C2R, can deliver end-users videos of excellent quality. However, the issue of balancing cost and performance exists as GPUs are usually high in cost, ranging from hundreds of dollars to thousands of dollars. With VMware ESXi 6.5, Nvidia GRID K2 supports 2 vGPU and only serves two virtual desktops at the same time, dramatically raising the cost per end-user. Previous results in Baseline Application Performance suggest that without GPU virtualization, edge cloud virtual desktops with sufficient computing resources and appropriate display protocols are capable of providing users with good performance in video delivery. Therefore, we conclude that though the virtual-desktop performance limitation goes away with GPU-virtualization, careful considerations should be given to the cost and the number of users that can be supported to determine whether GPU-virtualization is necessary for residential virtual-desktop computing.

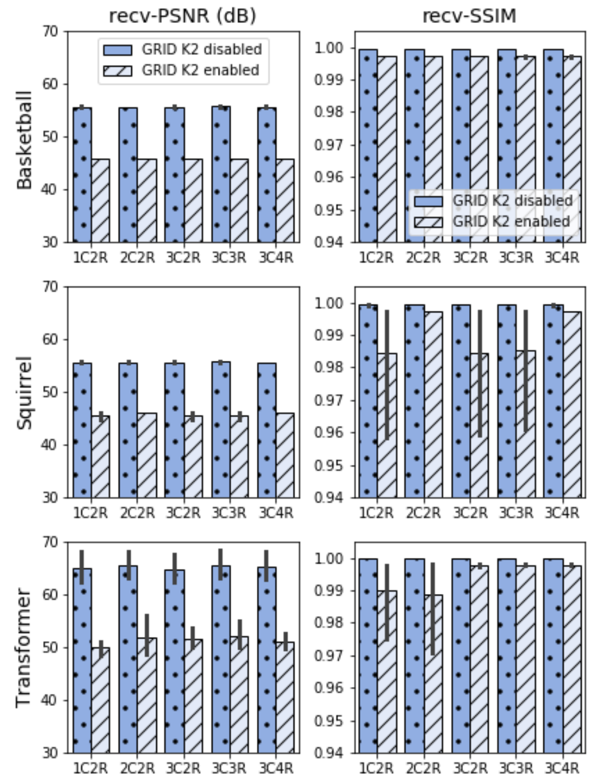


Fig. 14: Frame quality of virtual-desktop video playback with GPU-virtualization disabled and enabled

8 Conclusion

This study provides objective measurements to quantify edge cloud based virtual-desktop computing performance for residential use. The experimental results can be summarized as follows:

- Comparison of user-received performance was made among three display codecs and various virtual-desktop specifications using different types of videos. In general, PCoIP provides higher frame quality with a lower updating rate, whereas Blast Extreme protocol makes a trade-off between the frame quality and the frame-delivering rate based on available computing resources.
- The good quality of video playback delivered by the virtual desktop via edge cloud is preserved within 50 ms of fixed latency, 20 ms (mean value) of normally distributed latency, and 2% of peak packet loss rate.
- The delivered video quality shows very little reliance on the capabilities of client devices when sufficient computing resources are granted for the virtual desktop.
- Resource allocation sharing makes it possible to maximize the number of users that share edge cloud resources while maintaining good application performance.
- Using a server equipped with GPU, even low-specification virtual desktops could deliver users videos of excellent quality.

We plan to extend our study by evaluating residential application performance across various VDI solutions and performing quality evaluation on highly interactive residential-use applications, including Skype calls and gaming. The performance of open-source remote display protocols is another promising direction which potentially reduces the cost of edge cloud residential virtual-desktop computing.

ACKNOWLEDGMENTS

This research was funded by NSF grant #CNS-1737453. We thank Dawit Kahsay for his engagement in configuring Server Anvil, and Yizhe Zhang for her help in developing the CNN model. We also thank the Silicon Harlem team for their support. The authors would like to dedicate this paper to Malathi Veeraraghavan for her tremendous insight, passion, and leadership in this project.

REFERENCES

[1] VMware, *VDI: A New Desktop Strategy*, https://www.vmware.com/pdf/vdi_strategy.pdf.
[2] *Teradici PCoIP Zero Client*, https://www.teradici.com/products/desktop-performance-solutions/zero-clients.
[3] M. Louie, *The Angry Birds Movie' Created With Support by Teradici PCoIP*, https://connect.teradici.com/blog/the-angry-birds-movie-created-with-support-by-teradici-pcoip, 2016.
[4] M. Simsek, A. Aijaz, M. Dohler, J. Sachs, and G. Fettweis, "5G-enabled tactile internet," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 3, pp. 460–473, 2016.
[5] M. Antonakakis, T. April, M. Bailey, M. Bernhard, E. Bursztein, J. Cochran, Z. Durumeric, J. A. Halderman, L. Invernizzi, M. Kallitsis *et al.*, "Understanding the mirai botnet," in *26th USENIX Security Symposium (USENIX Security '17)*, 2017, pp. 1093–1110.
[6] A. M. Research, *Cloud-based VDI Market by Deployment Model, User Type and End User: Global Opportunity Analysis and Industry Forecast 2017-2023*, https://www.researchandmarkets.com/research/pm9p5q/global?w=4, 2017.
[7] J. Nieh, S. J. Yang, and N. Novik, "Measuring thin-client performance using slow-motion benchmarking," *ACM Transactions on Computer Systems (TOCS)*, vol. 21, no. 1, pp. 87–115, 2003.
[8] R. A. Baratto, L. N. Kim, and J. Nieh, "THINC: A virtual display architecture for thin-client computing," in *Proceedings of the twentieth ACM symposium on Operating systems principles*, 2005, pp. 277–290.
[9] N. Tolia, D. G. Andersen, and M. Satyanarayanan, "Quantifying interactive user experience on thin clients," *Computer*, vol. 39, no. 3, pp. 46–52, 2006.
[10] A. Berryman, P. Calyam, M. Honigford, and A. M. Lai, "VDBench: A benchmarking toolkit for thin-client based virtual desktop environments," in *2010 IEEE Second International Conference on Cloud Computing Technology and Science*. IEEE, 2010, pp. 480–487.
[11] W. Yu, J. Li, C. Hu, and L. Zhong, "Muse: A multimedia streaming enabled remote interactivity system for mobile devices," in *Proceedings of the 10th International Conference on Mobile and Ubiquitous Multimedia*, 2011, pp. 216–225.
[12] B. Song, M. M. Hassan, Y. Tian, M. S. Hossain, and A. Alamri, "Remote display solution for video surveillance in multimedia cloud," *Multimedia Tools and Applications*, vol. 75, no. 21, pp. 13 375–13 396, 2016.
[13] T. Song, J. Wang, J. Wu, R. Ma, A. Liang, T. Gu, and Z. Qi, "FastDesk: A remote desktop virtualization system for multi-tenant," *Future Generation Computer Systems*, vol. 81, pp. 478–491, 2018.
[14] F. Alali, T. A. Adams, R. W. Foley, D. Kilper, R. D. Williams, and M. Veeraraghavan, "Methods for objective and subjective evaluation of zero-client computing," *IEEE Access*, vol. 7, pp. 94 569–94 582, 2019.
[15] A. Chan, K. Zeng, P. Mohapatra, S.-J. Lee, and S. Banerjee, "Metrics for evaluating video streaming quality in lossy IEEE 802.11 wireless networks," in *2010 Proceedings IEEE Infocom*. IEEE, 2010, pp. 1–9.
[16] S. Laine and I. Hakala, "H. 264 QoS and application performance with different streaming protocols," *EAI Endorsed Transactions on Future Intelligent Educational Environments*, 2015.
[17] A. Hines, J. Skoglund, A. Kokaram, and N. Harte, "ViSQOL: The virtual speech quality objective listener," in *IWAENC 2012; International Workshop on Acoustic Signal Enhancement*. VDE, 2012, pp. 1–4.
[18] A. W. Rix, J. G. Beerends, M. P. Hollier, and A. P. Hekstra, "Perceptual evaluation of speech quality (PESQ)-a new method for speech quality assessment of telephone networks and codecs," in *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No. 01CH37221)*, vol. 2. IEEE, 2001, pp. 749–752.
[19] A. Hines, E. Gillen, D. Kelly, J. Skoglund, A. Kokaram, and N. Harte, "ViSQOLAudio: An objective audio quality metric for low bitrate codecs," *The Journal of the Acoustical Society of America*, vol. 137, no. 6, pp. EL449–EL455, 2015.
[20] M. Narbutt, A. Allen, J. Skoglund, M. Chinen, and A. Hines, "AMBIQUAL-a full reference objective quality metric for ambisonic spatial audio," in *2018 Tenth International Conference on Quality of Multimedia Experience (QoMEX)*. IEEE, 2018, pp. 1–6.
[21] K. Kumar, J. Navratil, E. Marcheret, V. Libal, G. Ramaswamy, and G. Potamianos, "Audio-visual speech synchronization detection using a bimodal linear prediction model," in *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*. IEEE, 2009, pp. 53–59.
[22] E. Marcheret, G. Potamianos, J. Vopicka, and V. Goel, "Detecting audio-visual synchrony using deep neural networks," in *Proc. Interspeech 2015*, 2015, pp. 548–552.
[23] Z. Duanmu, K. Zeng, K. Ma, A. Rehman, and Z. Wang, "A quality-of-experience index for streaming video," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 1, pp. 154–166, 2016.
[24] ITU-T, *Vocabulary for performance, quality of service and quality of experience*, http://handle.itu.int/11.1002/1000/13408, 2017.
[25] A. N. Moldovan, I. Ghergulescu, and C. H. Muntean, "A novel methodology for mapping objective video quality metrics to the subjective MOS scale," in *2014 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting*. IEEE, 2014, pp. 1–7.
[26] H. B. Martinez and M. C. Farias, "Full-reference audio-visual video quality metric," *Journal of Electronic Imaging*, vol. 23, no. 6, p. 061108, 2014.
[27] H. B. Martinez and M. C. Farias, "A no-reference audio-visual video quality metric," in *2014 22nd European Signal Processing Conference (EUSIPCO)*. IEEE, 2014, pp. 2125–2129.
[28] *Audio Video Sync Test 60 FPS*, https://www.youtube.com/watch?v=TjAa0wOe5k4, 2017.
[29] M. Chinen, F. S. Lim, J. Skoglund, N. Gureev, F. O'Gorman, and A. Hines, "ViSQOL v3: An open source production ready objective speech and audio metric," *arXiv preprint arXiv:2004.09584*, 2020.
[30] *VirtualGL 2.0 User's Guide*, https://virtualgl.org/vgldoc/2_0/.
[31] *AutoIt*, https://www.autoitscript.com/site/.
[32] Z. Akhtar and T. H. Falk, "Audio-visual multimedia quality assessment: A comprehensive survey," *IEEE Access*, vol. 5, pp. 21 090–21 117, 2017.
[33] *Audacity*, https://www.audacityteam.org/.
[34] *OBS Open Broadcaster Software*, https://obsproject.com/.
[35] *librosa*, https://github.com/librosa/librosa.
[36] H. Chen, R. Abbas, P. Cheng, M. Shirvanimoghaddam, W. Hardjawana, W. Bao, Y. Li, and B. Vucetic, "Ultra-reliable low latency cellular networks: Use cases, challenges and approaches," *IEEE Communications Magazine*, vol. 56, no. 12, pp. 119–125, 2018.
[37] G. Hasslinger and O. Hohlfeld, "The Gilbert-Elliott model for packet loss in real time services on the internet," in *14th GI/ITG Conference - Measurement, Modelling and Evalutation of Computer and Communication Systems*, 2008, pp. 1–15.
[38] *Standalone Network Emulator Tool*, https://docs.microsoft.com/en-us/archive/blogs/juanand/standalone-network-emulator-tool.
[39] *Geekbench*, https://browser.geekbench.com/.
[40] *PyAutoGUI*, https://pyautogui.readthedocs.io/en/latest/.
[41] *psutil documentation*, https://psutil.readthedocs.io/en/latest/.
[42] *Heaven Benchmark*, https://benchmark.unigine.com/heaven.

Hongying Dong received the B.S. degree in chemistry from the University of Science and Technology of China, Hefei, China, in 2014, the M.A. degree in chemistry from the University of Virginia, Charlottesville, U.S., in 2019, and the M.S. degree in computer science from the Georgia Institute of Technology, Atlanta, U.S., in 2019. She is currently pursuing the Ph.D. degree in computer engineering with the University of Virginia. Her research interests include network security, computer networks, and cloud computing.

Aaron T. Kinfe received the B.S. in information technology and engineering from Mekelle Institute of Technology, Ethiopia. He is currently pursuing his Ph.D. degree in computer engineering with the University of Virginia. His current research interests include computer network, cloud computing and Internet of Things.

Jiakai Yu is pursuing his PhD degree in electrical and computer engineering from the University of Arizona. He received the B.S. degree from Nankai University in 2013 and the M.S. degree from SUNY at Buffalo in 2015. His research interests include software-defined networking, 5G wireless & optical converged networks, low-margin fiber-optic network, and machine learning.

Qi Liu received the B.S. degree in communication engineering from University of Science and Technology Beijing, Beijing, China. She is currently pursuing her Ph.D. degree in Computer Engineering with the University of Virginia. Her current research interests are cloud computing, network security, and Internet of Things. Her recent work focuses on ubiquitous computing and Internet of Things security.

Dan Kilper is Professor of Future Communication Networks at Trinity College Dublin and Director of the CONNECT Center. Recently he was a research professor in the College of Optical Sciences and Electrical and Computer Engineering at the University of Arizona, Tucson. He holds an adjunct faculty position at the Columbia University Data Science Institute and was a faculty appointee at NIST in 2020. He received MS (1992) and PhD (1996) degrees in Physics from the University of Michigan. From 2000-2013, he was a member of technical staff at Bell Labs, and he co-founded Palo Verde Networks, a startup developing optical technologies for AI controlled software-defined optical networks. He holds thirteen patents and authored six book chapters and more than one hundred sixty peer-reviewed publications. His research is aimed at solving fundamental and real-world problems in communication networks, addressing interdisciplinary challenges for smart cities, sustainability, and digital equity.

Ronald D. Williams is a faculty member in Electrical and Computer Engineering at the University of Virginia. He has B.S. and M.S. degrees in electrical engineering from the University of Virginia. His Ph.D. in electrical engineering is from the Massachusetts Institute of Technology. His research and teaching is in the area of cyber physical systems.

Malathi Veeraraghavan received the B.Tech. degree from IIT Madras, and the M.S. and Ph.D. degrees from Duke University. After receiving the Distinguished Member of Technical Staff Award and a ten-year career with Bell Laboratories, she joined Polytechnic University, Brooklyn, NY, as a Faculty Member, where she was Associate Professor of electrical engineering, from 1999 to 2002. She joined the University of Virginia, in 2003, where she was a Professor with the Charles L. Brown Department of Electrical and Computer Engineering. Her research work was primarily in high-speed networking, wireless networking, and network security. Her research funding was mainly from the National Science Foundation, the U.S. Department of Energy, and DARPA. She had over 138 publications. She served as the Technical Program Committee Co-Chair from the High-Speed Networking Symposium at the IEEE ICC 2013 and the Technical Program Committee Chair for the IEEE ICC 2002. She received six best-paper awards. She was elevated to IEEE Fellow in 2020. She was an Associate Editor of the IEEE/ACM Transactions on Networking. She was an Associate Editor of the IEEE Transactions on Reliability, from 1992 to 1994.