IEEE Access
Multidisciplinary ⁝ Rapid Review ⁝ Open Access Journal

Received April 28, 2019, accepted May 22, 2019, date of publication June 26, 2019, date of current version July 31, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2925083

Methods for Objective and Subjective Evaluation of Zero-Client Computing

FATMA ALALI[1,2], TASHA A. ADAMS[3], RIDER W. FOLEY[1], DAN KILPER[3], RONALD D. WILLIAMS[1], AND MALATHI VEERARAGHAVAN[1]

[1] Department of Electrical and Computer Engineering, University of Virginia, Charlottesville, VA 22904, USA
[2] Department of Computer Engineering, Kuwait University, Kuwait City 5969, Kuwait
[3] College of Optical Sciences, The University of Arizona, Tucson, AZ 85721, USA

Corresponding author: Fatma Alali (fha6np@virginia.edu)

This work was supported by the NSF under Grant CNS-1737453.

ABSTRACT Zero clients are hardware-based devices without a central processing unit (CPU) that deliver virtual desktops (VDs) from remote computing systems to users. We measured the performance of applications accessed through zero clients to study the feasibility of using this approach to provide a desktop-pc experience across a network. Performance evaluation is complicated because monitoring software cannot be downloaded to the zero clients. Therefore, we introduce a new methodology and metric to measure zero-client VD performance that is based on network-traffic analysis. We conducted objective and subjective studies to determine the sensitivity of application-specific metrics to different network conditions. The results show that the packet loss rate (PLR) impacts zero-client performance for some applications such as video streaming. Subjective tests showed a greater user sensitivity to the PLR for video streaming than for image viewing or Skype. A strong correlation was found between the objective and subjective measurements but the rate at which these measurements changed with increasing PLR differed depending on the application.

INDEX TERMS Edge-cloud, measurements, objective study, QoE, remote desktops, subjective study, VDI, VDI metrics, virtual desktops, zero clients.

I. INTRODUCTION

Virtual-Desktop (VD) technologies have been growing in popularity as several remote desktop tools, such as VNC, Microsoft Remote Desktop Services (RDS), and TeamViewer, have been developed to support this paradigm. Remote desktop clients have been implemented in custom Application Specific Integrated Circuits (ASICs) or Field Programmable Gate Arrays (FPGAs). For example, Teradici Tera2321, is used in equipment called zero clients, which include interfaces such as USB, DVI, and HDMI ports to connect keyboard, video, mouse (KVM) terminals, and an Ethernet port for network connectivity.

Zero-client hardware has primarily been developed for enterprises, with each user being supported by one Virtual Machine (VM) in an edge- or commercial-cloud host. Thus, zero clients are used in schools, hospitals, libraries, and businesses [1]–[4]. Zero clients have also been used for high-performance graphics applications to allow multiple users to share the more expensive processing engines [5]. Graphics applications frequently exploit co-location of zero clients with graphics servers connected through a LAN. Zero clients also provide secure access for data-center management.

The introduction of edge cloud or fog computing creates a new opportunity to expand the range of potential applications for zero clients into personal or home computing (PC). The use of zero clients with edge clouds is appealing for the potential to provide high-performance computing experiences at favorable costs. Zero-client computing could enable sharing hardware and software licensing costs among many users. Each user or household would only require a zero client having network access to the edge cloud. Zero clients are less expensive than standard personal computers, and this hardware combined with access to edge servers may be a more affordable option than a PC for households without computers. While users can access many applications using their mobile phones, writing documents, applying for jobs and several other tasks still require physical or virtual desktop computing.[1]

The associate editor coordinating the review of this manuscript and approving it for publication was Haibo Wu.

[1] Aaron Smith, "Lack of broadband can be a key obstacle, especially for job seekers," 2015, Pew Research, https://www.pewresearch.org/fact-tank/2015/12/28/lack-of-broadband-can-be-a-key-obstacle-especially-for-job-seekers/

Thin clients have been marketed as affordable computing platforms, citing similar benefits [6]–[9]. However, consumer thin clients compromise performance and offer a lower-quality computing experience for multimedia usage [10], [11]. In contrast, zero clients are designed to deliver the full computer performance of the associated server and can offer high-quality computing experiences that include support for graphic-intensive applications. Moreover, the zero-client approach enhances security because zero clients neither run a standard OS nor expose a CPU on which attackers can install malicious software. However, using a zero client requires an active, reliable, and high-speed network for interaction between the servers and the zero clients. The requirement for zero-client connectivity can impose a mobility challenge as network access must be always available. Fortunately, edge cloud computing is being introduced precisely to provide reliable, high-speed, low-latency user access, and the so-called "mobile" edge clouds are planned to provide similar access for a mobile networking environment.

Even though VD technologies have evolved significantly over the past ten years, commercial zero clients are still limited to business and health sectors today. For example, North Kansas City Hospital deployed 700 zero clients allowing staff members to use any nearby "PC" in the hospital throughout the day to access centrally secured medical data, instead of having one dedicated PC per staff member [4]. In the health or business sector, employees tend to use a specific set of applications. On the other hand, a full PC experience is expected for residential use. As edge-cloud computing becomes available, it is important to understand the viability for zero-client use in this environment, and to characterize the computing experience that can be supported from edge clouds.

Previous work [12], [13] on VD performance was conducted with methods that depend on monitoring the performance at the end-user devices, which is not feasible with zero clients. Other proposed methods depend on monitoring the performance at the server [14]–[16] by monitoring CPU utilization, monitoring the display buffer to detect if a task has completed, or running commercial PC benchmarks on the server. Monitoring at the server does not consider the involvement of the network to transmit the display to the end-user devices. There has been some work on measuring VD performance by analyzing network traffic [13], [15], [17]; however, network traffic was analyzed only to measure video quality.

This paper introduces performance-measurement methods for zero clients that depend on analyzing the network traffic between the zero client and the server, and includes not only video-quality measurements, but also responsiveness as perceived by the end user. We address the following questions: (i) How do we measure zero-client performance without the ability to run monitoring software on the end-user devices? (ii) What is the impact of network conditions on application performance? (iii) Do objective measurements reflect user Quality of Experience (QoE) as determined through subjective measurement studies?

Objective and subjective measurements were obtained to evaluate application performance with zero-client computing. Objective measurements were defined in three categories: response time, video quality, and audio quality. For response time, we defined a new metric, Virtual Desktop Display Update Time (VD-DUT), which depends on analyzing the network traffic between a server and a zero client. Video quality was measured by analyzing network traffic, and by capturing the frame rate. Audio was captured using a hardware recording device connected to the zero client and the captured audio file quality was evaluated. The subjective measurement study involved 115 participants. For both the objective and subjective studies, four activities were performed to evaluate application performance: viewing 2D images, exploring 360-degree images, watching a video, and participating in a video-conference call.

We found that packet loss rate (PLR) impacts zero-client performance for some applications such as video streaming. Subjective tests showed a greater user sensitivity to PLR for video streaming than for image viewing or Skype. A strong correlation was found between the objective and subjective measurements, but the rate at which these measurements changed with increasing PLR differed depending on the application.

Our main contributions are as follows: (i) A new metric, Virtual Desktop Display Update Time (VD-DUT), was defined to measure zero-client responsiveness. (ii) We conducted the first, large-scale subjective study on zero-client performance with 115 participants. (iii) We quantified the correlation between objective and subjective metrics.

Section II describes the zero-client computing approach and the challenges of collecting measurements. Section III describes the objective evaluation approach including metrics, setup, and applications, and Section IV describes the subjective evaluation approach. Section V presents results for both objective- and subjective-study metrics and quantifies correlation between the metrics. After reviewing related work in Section VI, the paper is concluded in Section VII.

II. ZERO CLIENT COMPUTING APPROACH

Fig. 1 illustrates the zero-client computing approach. In this approach, virtual desktops are hosted on cloud or local machines. The zero client uses custom hardware to drive user devices such as KVM terminals. The zero client runs remote desktop protocols with encryption and video decoding. Supporting these protocols without a general-purpose CPU in user owned-and-operated end systems reduces costs while also reducing exposure to cyberattacks.

PC over IP (PCoIP) [18] is a high-performance display protocol used to deliver VDs to end-user devices (e.g., zero clients). Only display pixels and user input (e.g., keyboard strokes and mouse clicks) are sent over the network with all the processing being executed on a remote desktop server.

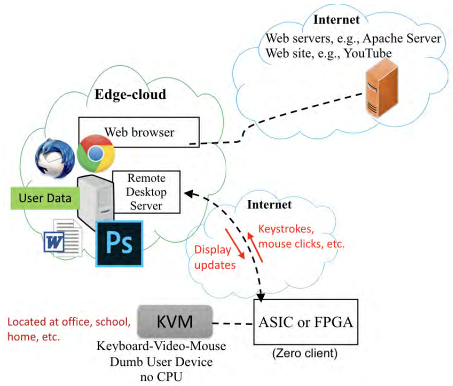

FIGURE 1. Zero client computing approach.

As mentioned earlier, measuring application performance in this zero-client approach is challenging because monitoring software cannot be installed and run on the zero client itself. Also, performance measurement at the edge cloud (i.e. the remote desktop server) alone may not may not suffice. For example, the vSIP study [14] used AppTimer to measure the time needed to load an application at the server. However, this time will not include the time to compress display updates, the time to send display updates over the network, the time for the zero client to receive, decode, and paint the display. Therefore, running measurement software tools at the server alone could lead to inaccurate results. Since monitoring applications cannot be run on the zero client, new monitoring approaches are needed.

III. OBJECTIVE EVALUATION APPROACH

An objective evaluation of the zero-client computing model was conducted by measuring the performance of the system while running different applications. User input was emulated using `Autoit` [19], which is a scripting language used to automate Windows GUI input by simulating keystrokes, mouse movements and clicks. Measurements were obtained from within the edge-cloud host (server), from packet traces between the edge-cloud host and the zero client, and by capturing the audio output of the zero client.

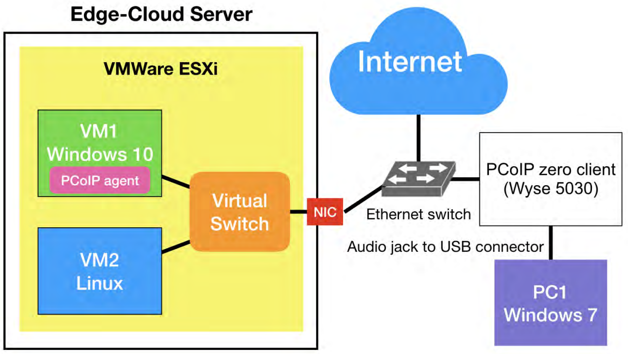

FIGURE 2. Experimental setup.

A. SETUP

Fig. 2 shows the setup used to run experiments and obtain measurements. An ASUS STRIX laptop was configured to operate as the edge-cloud server with VMWare ESXi 6.5.0, 16 GB RAM, four Intel i7 2.80GHz multithreading CPUs, and a 1 GE Network Interface Card (NIC). Two VMs were created within the server: (i) VM1 with Windows 10 OS, and (ii) VM2 with Ubuntu 16.04. The two VMs were connected via a virtual switch within the server. Port mirroring cannot be configured within an ESXi virtual switch, so we configured the virtual switch to operate in the promiscuous mode for packet capture on VM2. to allow VM2 to receive all packets. This configuration supported monitoring and processing of VM1-client network traffic. We used the Dell Wyse 5030 zero client, which supports PCoIP and is equipped with the Teradici TERA2321 chip. Both the server and the zero client were connected via an Ethernet switch with 1 Gbps ports. We used a USB audio capture device to obtain the audio output from the zero client. The audio jack from the zero client was connected through USB to a PC (PC1) running Audacity to record the audio from the zero client.

B. METRICS

Different metrics were used to measure the performance of the system. The metrics can be divided into 3 categories:

- Response Time (RT)
- Video Quality (VQ)
- Audio Quality (AQ)

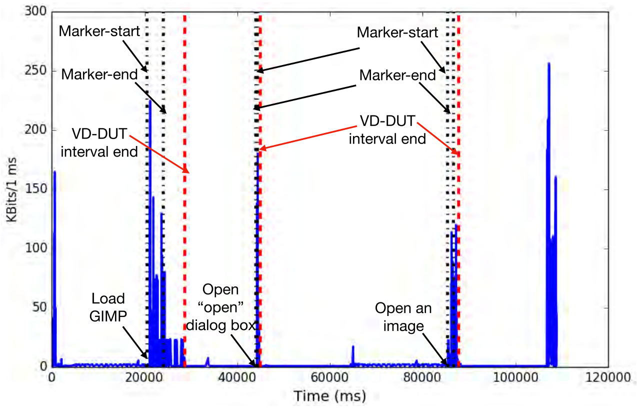

FIGURE 3. Network traffic capture from server to the client when three tasks were performed.

1) RESPONSE TIME

We used the slow-motion technique proposed by Nieh *et al.* [20] to measure the system response time. The (emulated) user initiates a single task and then waits for the server to perform the task. The user further waits for the response to be received and painted on the display before the user initiates another task. This technique allows for the network packets associated with each task to be identified within the traffic trace captured between the server and client. For example, Fig. 3 shows the network traffic from a server to a zero client with a 20-sec gap between the first and second task, and a 40-sec gap between the second and the third tasks.

Response times are computed from the time instants of traffic spikes. RT metrics include: RT-Autoit, RT-Marker and VD-DUT.

RT-Autoit is measured at the VM running in the edge-cloud host using an `Autoit` built-in function to detect when a task has completed. For example, `Autoit` uses the `WinWaitActive()` function to detect the rendering of a window for a launched application. This function locks the execution thread until `Autoit` detects that the application window has appeared on the display. When `Autoit` calls the `WinWaitActive` function, it is checking the VM frame buffer at the server to determine whether or not the application window has appeared on the display. Our traffic analysis showed that this metric does not include the time taken to send all the display updates to the client and the time taken by the client to process and draw the display.

RT-Marker is measured using the VDBench [21] method. This method requires that a marker packet (a UDP packet to a predefined port) be sent. When `Autoit` detects that the task has been performed, another marker packet is sent to indicate the end of the task. From the collected network trace, RT-Marker is obtained by computing the time between the two marker packets. Since there is background traffic between the edge-cloud and zero client to support the protocol, we need a threshold (τ) to identify the end of a display update.

To compute VD-DUT, our newly defined metric, instead of relying on an end-marker packet, our method finds the last display update packet in the traffic trace.

Example: Fig. 3 shows the network traffic trace from a server to a client when the `Autoit` script instructs the server to: (i) sleep for 20 sec, (ii) send a start-marker packet, (iii) load GIMP application (which is a photo editor), (iv) send an end-marker packet, (v) sleep for 20 sec, (vi) send a end-marker packet, (vii) trigger the "Open" dialog box, (viii) send an end-marker packet, (ix) sleep for 20 sec, (x) select a picture for the GIMP application, (xi) send a start-marker packet, (xii) trigger the "open" button to load the picture, (xiii) send an end-marker packet, and (xiv) sleep for 20 sec. Fig. 3 shows that, for task 1 (loading GIMP), even after the end-marker packet was sent, more packets were sent from the server to the client. Hence, we used a simple heuristic to decide which packets were part of the display update and should be considered when computing VD-DUT.

To determine an appropriate value for τ, we examined the network traffic from the server to the client. We found two types of packets: (i) periodic, small PCoIP communication packets, and (ii) display update packets. To understand the characteristics of the small periodic packets, we characterize the traffic from the server to the client under idle condition.

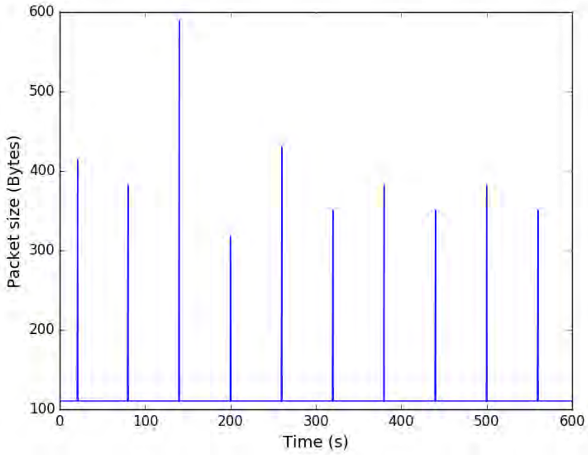

FIGURE 4. Packet size from the server to the client under idle condition with no display updates.

Fig. 4 shows a 10-min snippet of a collected packet trace showing packets sent from the server to the client under idle condition. The horizontal line in the plot represents small continuous 110-byte packets sent with a short inter-arrival time (on the order of hundreds of milliseconds). On the other hand, each vertical line represents two larger packets (of size 200 - 600 bytes) sent back-to-back approximately every 1 minute (the two packets overlap in the plot). In the illustrated 10-min packet trace, there were 1417 packets of size 110 bytes, and only 20 packets with a size larger than 110 bytes. Therefore, our heuristic assumes that any packet with a size greater than 110 bytes could be a display update packet.

The threshold τ was based on the time interval between arrivals of packets with a size greater than 110 bytes. If the inter-arrival time between two consecutive, larger-than-110 bytes packets (p_i and p_{i+1}) is greater than 500 ms, then the p_{i+1} packet is not part of the display update and VD-DUT is determined from the time at which the start-marker packet was sent to the time at which packet p_i was sent.

We chose 500 ms as our threshold after examining the inter-arrival time between the large packets (>110-byte) in many captured packet traces between the edge-cloud host and the zero client. We know that large packets sent between the start- and end-marker packets were task-related packets since we only send a start-marker packet after performing a task on the VM running in the edge-cloud host. We found the mean inter-arrival time between these large display update packets was 5 ms, the 99th percentile was 175 ms, and the maximum inter-arrival time was 490 ms. Since the inter-arrival time between the large packets under idle condition was approximately 1 min, we chose a number larger than 490 ms but smaller than 1 min. Specifically we chose 500 ms. To conduct a sensitivity analysis, we redid the analysis with 1 sec, but found the same results.

VD-DUT has its limitations in measuring the display time because it does not consider: (i) the time from a user mouse click until the packet carrying the mouse click is sent by the zero client, (ii) the time taken for the mouse click to reach the server, and (iii) the time the zero client takes to receive, decode, and paint the display with the update received from the server. The second time component can be considered by adding half a zero-client-to-server Round Trip Time (RTT) to

VD-DUT. However, the first and third time components are difficult to measure because the zero client has no general-purpose CPU on which to run monitoring software.

VD-DUT can be broken down into the server processing time (or RT-Autoit), transmission time, and retransmission time as shown in (1). T_{trans} is computed by dividing the total display-update size by the link rate.

$$\text{VD-DUT} = T_{proc} + T_{trans} + T_{retrans} \tag{1}$$

2) VQ

To measure video quality, we used two metrics: (i) received PCoIP frames per second (recv-PCoIP-fps), and (ii) video quality measured with the slow-mo-VQ approach (2) developed by Nieh et al. [20]. The frame rate is a good representation of video quality because PCoIP adjusts fps based on network conditions. Teradici offers a tool called Session Statistics Viewer (SSV) to measure received fps data, but this tool works with only PCoIP. On the other hand, slow-mo-VQ can be used more broadly, as it is independent of the remote desktop protocol. This slow-mo-VQ metric is computed by analyzing network traffic, where P in (2) represents the tested video. Slow-mo-VQ compares the slow-motion video to the regular-speed video to quantify how many frames were dropped, or not transmitted, by examining the total bytes transferred and the time required to play the video. A video is first played back at 1 fps and a network trace is captured. The video is then replayed at regular speed. The video playback at the low fps rate is used to establish a reference data-transfer rate (total data transferred divided by the total playback time), which corresponds to a perfect video playback with no dropped frames. This rate is then compared with the data-transfer rate of the regular-speed video.

$$\text{slow-mo-VQ} = \frac{\dfrac{Data\ Transferred(P)\ /\ Playback\ Time(P)}{Ideal\ FPS(P)}}{\dfrac{Data\ Transferred(\text{slow-mo})\ /\ Playback\ Time(\text{slow-mo})}{Ideal\ FPS(\text{slow-mo})}} \tag{2}$$

3) AQ

A survey on perceptual quality assessment for audio/visual services [22] showed that video is the dominant factor in determining the Quality of Experience (QoE) of streamed videos, while in a video-conference call, the dominant factor is audio quality. Therefore, for video playback testing, only the video quality was measured, and for video-conference-call testing only audio quality was considered when measuring the performance.

To measure audio quality, we used three objective audio/speech evaluation metrics: (i) Weighted Spectral Slope (WSS) [23], (ii) Log-Likelihood Ratio (LLR) [24], and (iii) Virtual Speech Quality Objective Listener, (ViSQOL) [25]; all of which are signal-based, full-reference metrics. WSS and LLR were not developed with VoIP as a target; on the other hand, ViSQOL was designed to be a general objective speech quality metric for VoIP. ViSQOL was developed as an alternative to commercial, industry-standard speech quality metrics: Perceptual Evaluation of Speech Quality PESQ [26] and Perceptual Objective Listening Quality (POLQ) [27]. Both WSS and LLR measure the distance between the reference signal and the degraded signal. ViSQOL uses the Neurogram Similarity Index Measure (NISIM) to determine the similarity between the two signals, which is then mapped to a value within the range of 1 to 5.

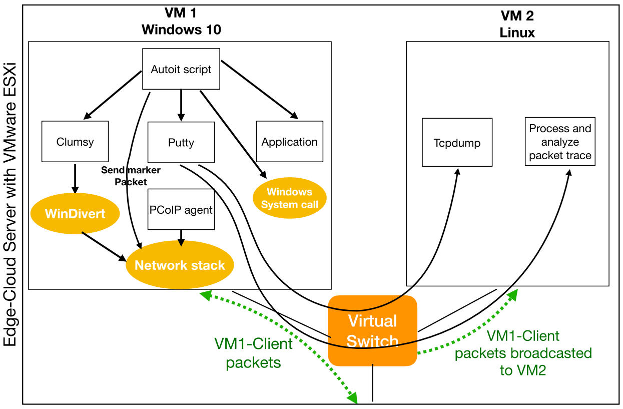

FIGURE 5. Application automation process.

C. APPLICATION AUTOMATION

Fig. 5 shows how different components were used to automate and execute the experiments, and how these components interacted with each other. VM1 has five main components: (i) `Autoit` script, which emulates different user activities, (ii) the application, which is being tested and controlled via an `Autoit` script, (iii) Clumsy [28], which is a network emulation tool used to change the network conditions within the VM, (iv) Putty, to allow VM1 to remotely access VM2 and initiate the network traffic monitoring script, and (v) PCoIP agent, which is used to enable the remote desktop access.

Four activities were considered for the performance evaluation study: 2D image viewing via Windows `Photos`, video playback via `MPlayer`, 360-degree image exploration via `Chrome`, and video-conference call via `Skype`. Five packet loss rate (PLR) values of 0%, 0.5% 3%, 5%, and 10% were used for the evaluation.

1) 2D IMAGE VIEWING AND 360-DEGREE IMAGE EXPLORING

Each activity was automated by an `Autoit` script as follows: (i) Instruct Clumsy to change the network configuration. Clumsy leverages Windows WinDivert package that allows user-layer applications to manipulate sent and received network packets. (ii) Initiate tcpdump at VM2 via an ssh connection with Putty. (iii) Send a UDP start-marker packet. (iv) Instruct the application to perform a specific task (e.g., open an image or play a video), and wait for a Windows system call to provide a signal when the task completes execution. (v) Send an end-marker packet. (vi) Stops Clumsy.

(vii) Stop packet capture at VM2 via Putty. (viii) Initiate an analysis script at VM2 to parse the captured packet trace, compute the total number of bytes sent on PCoIP, compute VD-DUT, and save the results.

For 2D image viewing, six images were used with a high resolution between 1024 × 1545 to 5719 × 3803, and unique pixels[2] in the range 309K - 877K. During this test, all images were opened in order to pre-load the data into the RAM before taking measurements. In each run, all six images were viewed sequentially with a 20-sec inter-image gap. Each run was repeated 70 times, the geometric mean of VD-DUT and total sent bytes of the six images were computed, and then the arithmetic mean values were computed across runs.

For the 360-degree image exploration application, we used a web-hosted 360-degree image tool to explore images, instead of running a tool locally on the edge-cloud server. Our edge-cloud host is not equipped with virtualization support for the GPU, and hence the application performance was poor. Since our goal is not to evaluate the computing resources of the edge cloud, but rather to evaluate the zero-client approach, we used a commercial-cloud 360-degree image viewing application. Three images were explored via Pano2VR running on a web server.[3] In each run, for each image, the `Autoit` script would open a new tab in a Chrome web browser, visit the image URL, drag-and-drop to change the scene, and zoom-in and zoom-out. A 20-sec gap was introduced after every two tasks. This sequence was repeated 90 times. For each run, VD-DUT was computed for each task performed on each image, the geometric mean of VD-DUT for all images across all tasks was computed for each run, and then the arithmetic mean was computed across runs.

2) VIDEO

To obtain video measurements, the following steps were followed: (i) at VM1, the `Autoit` script initiates tcpdump at VM2, (ii) plays a video via MPlayer at 1 fps rate, (iii) after the video playout completes, the script logs the time that was taken to play the video and stops tcpdump at VM2, (iv) configures the network with Clumsy, (v) initiates tcpdump at VM2, (vi) starts SSV at VM1 to capture fps, (vii) starts the video at the regular fps rate, (viii) after the video playout completes, the script logs the time that was taken to play the video and stops tcpdump at VM2, and (ix) executes the analysis script at VM2 to parse the collected packet traces by extracting total bytes for the two playout sequences and uses this data to compute video quality `slow-mo-VQ` (2). Steps (iv-viii) are repeated for different network conditions. Each run was repeated 75 times and mean values were computed. A 36-sec video from the animated movie "Zootopia" with 23.9 fps rate was used.

3) SKYPE

Evaluating Skype performance involves metrics that do not require collecting network packet traces. The audio received on the AUX jack of the zero client was captured and processed to measure call audio quality. A USB audio capture device was used to capture the audio output of the zero client. The captured audio was forwarded to PC1 which runs `Audacity` to record the audio.

AQ metrics require comparing the recorded audio file to a reference file. To obtain a reference file, a Skype call between two PCs was performed. An audio file was fed to Skype at one PC, and the audio output of the second PC was recorded. The recorded audio file represented the reference audio. The above steps were repeated four times with two audio files with male and female speakers from ITU-P.862 conformance data (`u_am1s03` and `u_af1s02`). Each audio file was recorded twice to account for the variability of Skype calls. Four reference audio files were obtained from the above experiment and were used as the reference to compute the audio quality metrics (`u_af1s02_f_ref1`, `u_af1s02_f_ref2`, `u_am1s03_m_ref1`, and `u_am1s03_m_ref2`).

The following steps were taken to conduct the zero client Skype experiment: (i) a Skype call was initiated from PC1 (Fig. 2) to VM1 in the edge-cloud host. (ii) A master `Autoit` script was executed on VM1. (iii) The master script starts by configuring the network using Clumsy, then connects to PC1 via Microsoft RDP. (iv) The master script then starts another `Autoit` script (play-and-record script) at PC1, (v) The script at PC1 starts recording by running `Audacity` and then instantly feeds an audio file to the Skype call (`u_am1s03` or `u_af1s02`). (vi) After the audio file ends, the `Autoit` script at PC1 stops recording, exports the recorded audio as a `wav` file, and analyzes the collected audio file by computing the AQ metrics. Each run consists of repeating the above steps five times for each network condition. Five runs were executed before changing the network condition to give PCoIP enough time to adapt to the network changes. The experiment was repeated 50 times for each audio file (`u_am1s03` and `u_af1s02`), and the two corresponding references were used in each run to compute the AQ metrics.

IV. SUBJECTIVE EVALUATION APPROACH

We conducted studies to evaluate users' subjective experiences with the zero-client computing approach, and to quantify the relationship between objective and subjective measurements. A total of 115 participants (66 males and 49 females) at the University of Virginia completed the subjective study in the fall of 2018. Participants rated their experiences using the Mean Opinion Score (MOS) with a 5-point Absolute Category Rating (ACR) scale following ITU-T Recommendation P.800 and P800.1 [29], [30]. Each participant was asked to assess the Quality of

[2] In the RGB color model, the color of every pixel in an image is defined by a particular combination of pure red, green, and blue color values. The number of unique pixels in an image is computed by finding the number of unique combinations of RGB values. PCoIP reuses repeated pixels at the client to reduce the number of bytes sent by the server, i.e., the larger the number of unique pixels, the higher the number of bytes sent from the server to the client. The number of unique pixels was determined using a Linux tool called `magick`.

[3] https://rodedwards.com/interactive-files/Chatsworth_House/index.html

Experience (QoE) for each application on a scale from 1 (bad) to 5 (excellent). The same applications were used to evaluate performance in both the objective and subjective studies.

A. SETUP

Two testing stations (cubicles) were configured with identical keyboards, mice, and monitors (Dell LCD) with 1680 × 1050 resolution. A Wyse 5030 zero client was used in one cubicle, and an LG CBV42-B PCoIP zero client was used in the other. Both zero clients are equipped with the same Teradici TERA2321 PCoIP processor.

The arrangement for this study was similar to that used for the objective study shown in Fig. 2. This study includes an additional VM (VM3) with the same configuration as VM1. Also, the LG PCoIP zero client was connected to the physical Ethernet switch shown in the setup. VM2 was not used to collect packet traces during this subjective study. Subjective and objective experiments were executed during different time periods using the same arrangement.

B. METHODOLOGY

Upon arrival, each participant was seated in one of the cubicles and directed to click on an application icon located in the middle of the Desktop. This application icon executed our master script, which then initiated the test applications and collected user input. Participants first read the informed consent agreement and, if they agreed to the terms, they were directed to a short survey that captured information about their experience. The actual test started by asking the participant to execute multiple activities in sequence.

Four applications (each with multiple activities) were used to evaluate the experience: (i) image viewing via Windows `Photos`, (ii) 360-degree image exploration via `Chrome`, (iii) video playback via Windows `Movies & TV`, and (iv) a video-conference call via `Skype`. The first two applications were used to collect data related to the responsiveness of the system. The other two applications collected data related to audio and video qualities. During each activity, PLR was changed to test the performance under different network conditions as emulated by Clumsy.

1) IMAGE VIEWING

Each participant was asked to view the same six images that were used in the objective study, and rate the quality of each image without considering the image content. Each participant was asked to look at each of the images three times with the PLR changed for each of the views.

2) 360-DEGREE IMAGE EXPLORATION

Each participant was asked to explore three 360-degree images. Every 15 seconds during the exploration, a window appeared asking the participant to rate the responsiveness of the system and the quality of the images. A new PLR value was configured before each image was presented for exploration.

3) WATCHING A VIDEO

Each participant was asked to watch a 36-sec video three times with the PLR changed for each iteration. Each participant was asked to rate each viewing iteration based on the quality of the video and audio without considering the content.

4) SKYPE

Each participant was asked to join a 2-min video call via Skype with one of two research assistants. Every 40 seconds during the call, a window appeared asking the participant to rate the call quality. Call variability was limited during the test by locating the research assistant receiving the call always in the same room, using the same laptop and connected to the same network. We controlled the call conversation by asking the participant to play the "20 questions" game. The participant would think of a person or item, and the research assistant would be allowed to ask up to 20 questions to identify the person or item. We chose this interaction instead of using a written script because we wanted the participant to focus on the call quality rather than reading a script. The research assistant asked the questions to the participant to ensure that the participant was listening and paying attention to the audio quality.

Upon completion of each application test, the participant was invited to continue with the next application or opt out. Thus, participants had the option of rating their experiences for one, two, or three applications. The activities were automated using an `Autoit` script. Participant interaction with the study personnel was not necessary, and this helped to reduce any influence of study personnel over participant ratings. The automation also maintained test consistency and controlled the testing time. The automation script performs the following tasks: (i) shows the participant a dialog box to describe the activity, (ii) runs the test application (e.g., visits the 360-degree image URL, or opens the directory that includes the 2D images that need to be explored), (iii) interrupts the test at specific time intervals to ask about user experience, and (iv) changes the network configuration.

C. DATA ANALYSIS APPROACH

MOS values for each combination of application and PLR value were computed and are reported in Sec. V-B. To study the subjective measurements, we conducted a pairwise t-test to check if the MOS values across different PLR values are significantly different. The t-test can reject the null hypothesis that there is no difference between the mean values (MOS) across PLR.

Because of the repeated measures required in our study, where the same subject rates the experience under different network conditions, a subject dependency is expected. The subject dependency in the results occurred because each participant provided experience ratings for 3 PLR values; the subject might have rated the experience with a 3% PLR while recalling the previous experience with a 0% PLR baseline. To account for this subject dependency in our study, we performed further analysis using a Linear Mixed effect Model (LMM). LMM has two types of effects: fixed and random. We used PLR as our fixed effect and the subject as our random effect ($QoE \sim PLR{+}subject$). We conducted Tukey's Honestly Significant Difference (HSD) test to check if the differences between the groups (PLR) were significant. HSD adjusts the p-value based on the total number of pairwise comparisons. It is very conservative with respect to Type I error (rejecting the null hypothesis when it is true). We used R package "lme4" version 1.1-19 to fit the data to LMM.

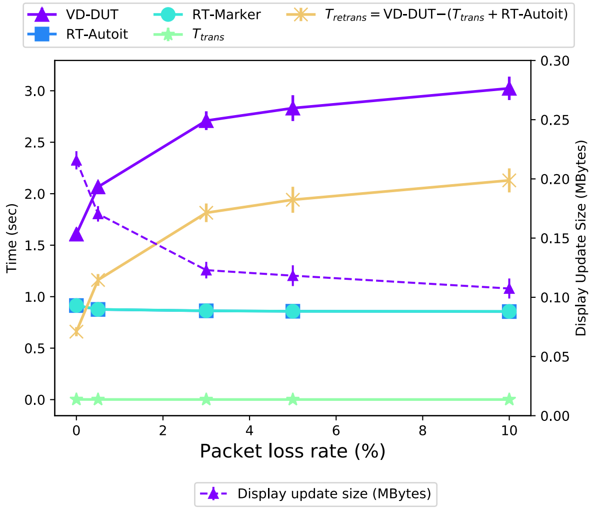

FIGURE 6. Response Time breakout for image viewing. RT-Autoit and RT-Marker plots overlap.

V. RESULTS

A. OBJECTIVE EVALUATION RESULTS

1) IMAGE VIEWING

Fig. 6 shows the mean of the different RT metrics. When the packet loss rate increases, the most significant impact on VD-DUT. Both RT-Marker and RT-Autoit remain unaffected by packet loss rate because both these timers are based on monitoring the server frame buffer, which means network activities do not impact these timers. RT-Autoit and RT-Marker will be affected by the processing time on the server, e.g., if many applications share the VM CPU resources, then RT-Autoit and RT-Marker are likely to be higher.

Since processing time (RT-Autoit) and transmission delay (T_{trans}) are unaffected by PLR, using (1), we conclude that the increase in VD-DUT is due to an increase in retransmission time. This time also includes the time for processing packets at the client since retransmissions require participation of both ends of the protocol.

While VD-DUT increases with PLR, the total number of sent bytes decreases. If the total bytes decreases, one would expect VD-DUT to decrease as fewer packets are sent. However, VD-DUT increases because of the extra time required to retransmit lost packets. Also, the total number of transmitted bytes decreases due to PCoIP decreasing the display resolution when detecting a high packet loss rate. Therefore, users would notice an increase in display update time and a decrease in display resolution at high packet loss rates.

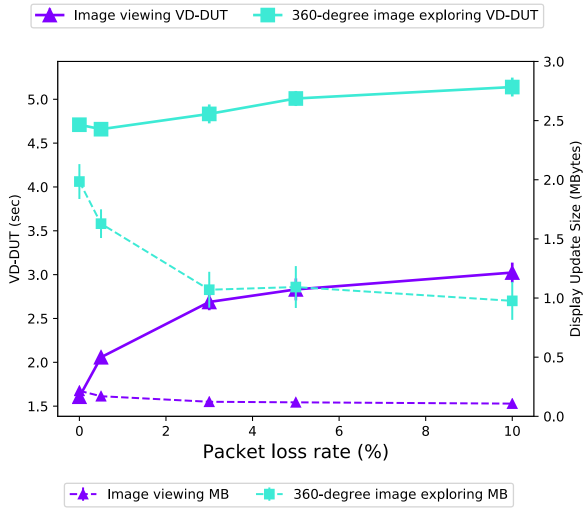

FIGURE 7. 360-degree image exploring.

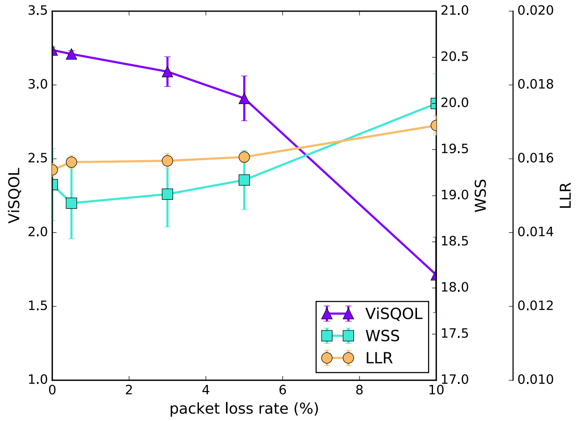

FIGURE 8. Skype AQ measurements obtained via three different metrics.

2) 360-DEGREE IMAGE EXPLORING

Fig. 7 shows mean values of VD-DUT and display update size. Similar to the 2D image viewing application, VD-DUT increases with packet loss rate. However, the rate of increase was lower when compared to 2D image viewing (9.1% increase rate between 0 to 10% packet loss rate for the 360-degree images, while for 2D images, the increase rate was 46.7%). This behavior could be due to the nature of the 360 images in which the display is changing rapidly; hence PCoIP is not taking time to retransmit the lost packets, as with the 2D images. In the latter case, the display stays unchanged for some time.

3) SKYPE

Fig. 8 shows Skype performance measured using different AQ metrics. In general, AQ metrics values decrease with increasing PLR. When PLR increased from 0% to 10%, VisQOL dropped by 47.04%, WSS increased by 4.59%, and LLR by 7.65%. At PLR of 0%, ViSQOL was computed to be 3.2, which is approximately the average value since ViSQOL is designed to range between 1 and 5. This indicates that video-conference calls via zero clients have an average quality under even ideal network conditions.

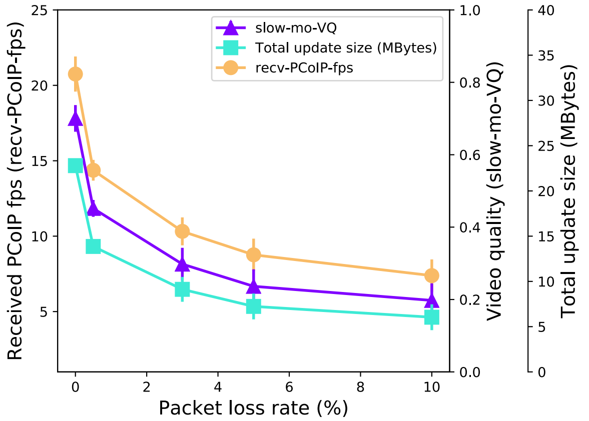

FIGURE 9. **Video quality across different PLR.**

4) VIDEO

Fig. 9 shows the video quality measured via slow-mo-VQ and recv-PCoIP-fps rates, where both metrics decrease as PLR was increased. At PLR value 3% and higher, the changes in recv-PCoIP-fps and slow-mo-VQ were smaller than when PLR was increased from 0 to 3% Video quality dropped when PLR was changed from 0% to 10% at a rate of 64.4% for recv-PCoIP-fps and 71.8% for slow-mo-VQ. At 0% PLR, the video quality reached 20.75 recv-PCoIP-fps rate, which is less than the original video rate (23.9), and slow-mo-VQ achieved 70%. The amount of received data decreased to 6 MB with a PLR setting of 10% (network traffic captured after packets were dropped by the network emulator). The decreased volume of received data implies that some frames were dropped or partially dropped (when a few pixels are dropped, frame construction becomes difficult). Such frame/pixel drops explain the low recv-PCoIP-fps rate of 7.4 at 10% PLR.

B. SUBJECTIVE EVALUATION RESULTS

Table 1 shows the total number of collected QoE values corresponding to each tested PLR case. The 0% PLR case has a higher number of collected QoE values because both stations had 0% PLR as an initial condition, whereas the tests at other PLR values were divided between the two stations. The minimum number of collected QoE values is 24 (ITU recommends that a minimum of 15 participants are required to evaluate image quality on a screen [31]).

TABLE 1. **Total number of collected QoE values for each activity and packet loss rate value.**

	packet loss rate (PLR) %				
Activity	0	0.5	3	5	10
Image viewing	75	31	44	44	31
360-degree image exploring	58	24	34	34	24
Video playback	66	28	38	38	28
Skype	53	26	27	27	26

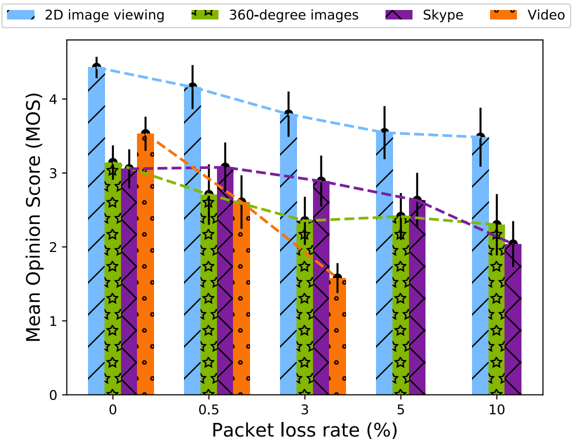

FIGURE 10. **MOS values of different applications with 95% confidence interval.**

1) MOS ANALYSIS

Fig. 10 shows MOS values for different applications with 95% confidence intervals (CI). Image viewing is the only application that achieved MOS value higher than 4 with 0% PLR (MOS = 4.43). Even with the high PLR of 10%, image viewing MOS value did not drop below 3. This indicates that participants were less sensitive to decreased-resolution, still images. For 360-degree images, MOS with an ideal network condition (0% PLR) had a lower rating (MOS = 3.31) when compared to 2D image viewing. The highest drop rate of MOS occurred when PLR increased from 0 to 3%. After 3% PLR, MOS value continued to decrease but at a lower rate.

For Skype, it is interesting to note that even with the very high PLR value of 10%, MOS value did not drop below 1. This could be due to: (i) participants expecting video-conference calls to have poor quality, and (ii) audio quality being the dominant factor when evaluating video-conference calls as shown by a previous study [22]. More bandwidth is assigned to the audio channel during video-conference calls, and the impact of 10% PLR was not as high as we expected.

The MOS for video playback showed the most dramatic decrease to 1.58 with 3% PLR. Video applications require larger data transfers because the display changes rapidly; the objective measurements showed that the video playout had the highest number of sent bytes among the tested applications. Therefore, video playback should be the most sensitive application to PLR when compared to the other tested applications. We did not collect QoE values beyond 3% PLR because the MOS value had already dropped to a low value of 1.58.

TABLE 2. T-test pairwise p-value for different applications.

(a) Image Viewing

	0	0.5	3	5
0.5	1.99e-01			
3	6.74e-04	1.07e-01		
5	2.74e-06	6.95e-03	2.25e-01	
10	7.73e-06	6.13e-03	1.69e-01	7.85e-01

(b) Skype

	0	0.5	3	5
0.5	9.28e-01			
3	4.53e-01	4.69e-01		
5	5.70e-02	8.59e-02	3.13e-01	
10	1.25e-05	1.08e-04	1.26e-03	2.37e-02

(c) 360-degree image

	0	0.5	3	5
0.5	0.075699			
3	0.000317	0.188981		
5	0.000837	0.272655	0.810676	
10	0.000231	0.141992	0.834638	0.659491

(d) Video playback

	0	0.5
0.5	1.02e-05	
3	1.18e-19	8.65e-06

2) T-TEST ANALYSIS

Table 2 shows the p-value for a pairwise t-test conducted on the different PLR values across each application with a cutoff of 0.05. Highlighted table cells have p-values < 0.05 indicating a significant difference between the two PLR values. For image viewing, we cannot reject the null hypothesis for the consecutive pairs (0,0.5), (3,5), and (5,10). We could also see in Fig. 10 that the decrease rate was low, i.e., MOS values were close. These results were expected considering the activity of browsing through still images. Participants could sense a difference between no packet loss (0%) and packet loss (3, 5, or 10%), but they could not necessarily sense a difference in performance for PLR values higher than 0%.

In contrast, for the interactive Skype application, the t-tests among the groups of PLR showed more significant differences. By comparing 0, 0.5, 3, and 5% loss rate values, we cannot conclude that participants noticed a difference in the performance. However, with the higher PLR of 10%, we can conclude that the QoE rating was significantly different when compared to the lower PLR of 0, 0.5, 3 and 5%. The results from 360-degree image exploration allow for the rejection of the null hypothesis only when comparing 0% PLR to the other PLR values, and we can conclude that the participants had a better experience with no packet loss rate when compared to any tested PLR case. Fig. 10 supports these results as the QoE mean values (MOS) of 3, 5, and 10% PLR, are close. On the other hand, the video playback t-test results showed a significant difference between the QoE values across the different PLR cases with a very low p-value between 0 and 3% (1.18e-19).

3) LMM WITH HSD TEST ANALYSIS

For different applications, Fig. 11 shows the 95% CI of the difference between the mean values for each PLR pair. The mean values were obtained using the LMM, and CI lines were computed by applying an HSD test between the mean values for each group (PLR case). In the plots, if CI includes zero within its range, then we cannot conclude that the two means are statically significantly different because there is a chance that the difference between the two mean values of a PLR pair is zero. Fig. 11 shows more conservative results compared to the t-test results reported in Table 2. For the video application, the HSD test on the fitted LMM showed similar results to the t-test such that there were significant differences between the mean across all PLR pairs. For the other three applications, there were statically significant differences between the mean values when PLR was 0% and when PLR was greater than 0%. Even though for the other PLR pairs, we cannot reject the null hypothesis, Fig. 11 shows that the difference between MOS values across many pairwise PLR groups is near-marginal significance (for some CI, only a small part of the line crossed 0).

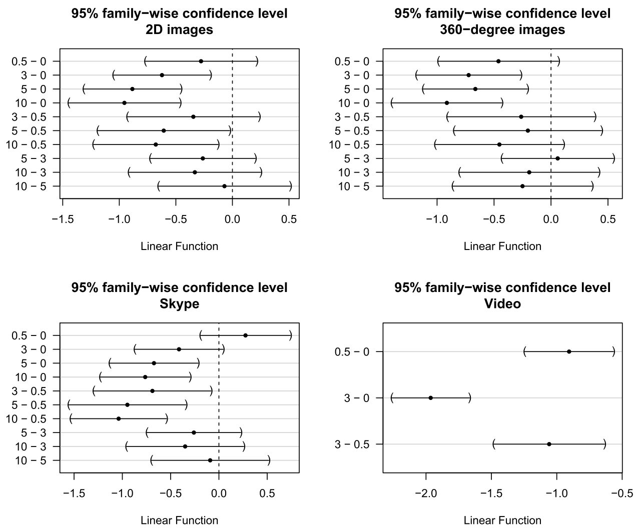

FIGURE 11. Pairwise 95% confidence interval of the difference between every two mean values across PLR using a Linear Mixed-effect Model with HSD test.

4) SUBJECTIVE AND OBJECTIVE CORRELATION

To study the correlation between the objective (VD-DUT, VQ, AQ) and subjective MOS values, we used Pearson's correlation coefficient (r). For the tested applications, r values were found to be as follows: image viewing ($r = -0.990$), 360-degree image exploration ($r = -0.733$), Skype-LLR ($r = -0.963$), Skype-WSS ($r = -0.946$), Skype-ViSQOL ($r = 0.974$), video-recv-PCoIP-fps ($r = 0.987$), video-slow-mo-VQ ($r = 0.986$). These results imply strong correlations between the objective metrics and the subjective MOS values. However, the rates of increase/decrease in the objective and the corresponding subjective metric were different when PLR increased.

Table 3 shows the slope of a linear model for each application and method (LMM was used to fit the subjective results). For image viewing, both objective and subjective

TABLE 3. The slope of fitted linear models of each application for both objective and subjective measurements.

App	Method	slope
Image viewing	Objective-VD-DUT	0.1260
	Subjective	-0.1041
360-degree image	Objective-VD-DUT	0.0483
	Subjective	-0.0866
Skype	Objective-ViSQOL	-0.1494
	Objective-WSS	0.0961
	Objective-LLR	0.0001
	Subjective	-0.0979
Video	Objective-recv-PCoIP-fps	-2.9060
	Objective-slow-mo-VQ	-0.1102
	Subjective	-0.6565

metrics have approximately the same absolute line slope indicating that PLR impacted both metrics at the same rate. For Skype, subjective measurements, ViSQOL, and WSS have approximately the same increase/decrease rate caused by PLR increase. On the other hand, LLR has a smaller slope value, indicating a minimum impact of PLR on performance (i.e., LLR underestimated the effect of PLR when compared to the subjective results). For 360-degree image exploration, the subjective metric (MOS) decreased faster than the rate at which the objective metric (VD-DUT) increased indicating a different impact by PLR (i.e., the subjective metric was more sensitive to PLR changes when compared to the objective metric (VD-DUT)). For the video analysis, we only used the objective data points collected at PLR of 0, 0.5, and 3% since we only collected subjective data at these three PLR values. Comparing the objective recv-PCoIP-fps and slow-mo-VQ to the subjective QoE results, the subjective metric decreased when PLR increased at a rate higher than the rate of decrease of slow-mo-VQ, and at a rate lower than the rate of decrease of recv-PCoIP-fps. In conclusion, objective metrics showed a decreasing trend as PLR was increased (which matches the subjective results); some objective metrics underestimated while others overestimated the impact of PLR on performance.

VI. RELATED WORK

Many studies have been conducted on objective measurements to quantify virtual desktop performance and/or to evaluate new solutions for virtual desktops. Nieh *et al.* [20] proposed a methodology to measure the performance of thin clients via slow-motion techniques based on monitoring network traffic. Packet traces are collected in an ideal environment as a baseline and then compared against packet traces collected under different network conditions and server loads. Slow-motion techniques have been used by other researchers as well [13], [15], [17], [21], [32].

VDBench [21] is a thin-client benchmarking tool that uses slow-motion techniques. Realistic loads of multiple applications were generated to compare different remote desktop protocols by measuring video quality and application response time under various server loads and network conditions (round-trip time and packet loss rate). CloudRank-V [33] is another benchmarking tool that uses network traces to find response time (latency). A method to generate complex workloads was proposed by mixing nine applications, and the maximum number of VMs the server can execute before user performance degradation is noticeable was determined. Our objective study is similar to VDBench and CloudRank-V, where we measured response time based on analyzing network traffic by defining our own metric VD-DUT (VDBench response time is based on when the action is performed at the server side, and the CloudRank-V threshold to define start and end of display updates was not listed in the paper). To the best of our knowledge, ours is the first study on virtual desktops accessed through zero clients as prior studies used thin clients or other computing platforms.

VNCPlya [34] and DeskBench [12] are VD benchmarking tools that measure application response time by monitoring the status of the display buffer. When a specific task is performed (e.g., opening a text editor), the display buffer is captured and used as a reference. To measure the response time, the tool performs the same specific task, and then keeps comparing the test-case display buffer to the reference display buffer until they match (DeskBench uses a hash function of the display buffer instead of the image itself). Pandey *et al.* [35] proposed a framework to facilitate VD benchmarking and allow adaptation to changes in VDI software architecture. Song *et al.* [13] presented FastDesk, a remote desktop system for multiple tenants, which was evaluated by measuring CPU utilization, response time and video quality for different applications. However, all this prior work relied on running software on the client to monitor the display buffer or capture mouse clicks, which is not feasible in a zero-client setup.

Sui *et al.* [14] evaluated their proposed virtual scheduler for interactive performance (vSIP) at the server side without considering the remote desktop protocol or the client end-device. The evaluation metrics included video quality measured by the rate of dropped frames at the server side, cold and warm launch time of applications, and web-page loading time. Server-side measurements were also used in the Zhou *et al.* study [16]. Our goal is to evaluate the end-user perceived performance; thus, server-side measurements alone are not sufficient.

Some studies focused only on video quality and developed measurement methods [15], [16], [32], [36], [37]. For example, Laine and Hakala [36] used displayed image frames, the frame rate and play duration as performance metrics. Yu *et al.* [32] used a modified slow-motion video quality metric.

Subjective assessments have been used to measure user quality of experience [14], [38]–[44]. The number of participants on the assessments varied between 10 to 40, and Mean Opinion Score (MOS) was uwebsed to evaluate the performance. These studies focused only on the quality of video or gaming experience [45]; other applications in addition to video were considered in our subjective study.

Casas *et al.* [46] undertook a subjective study with 52 participants to measure QoE using Citrix technologies. Each participant performed several tasks (text editing, drag and drop, scroll down, and web browsing) and evaluated the experience using MOS under different RTTs. The authors characterized traffic by collecting packet traces, measured the response time, and compared the response time when using Citrix with response time when running the applications on a desktop. Our paper is similar to the Casas *et al.* paper as different applications were considered. However, our study focused on zero-client performance, and in addition to the subjective study, we conducted a correlation study of subjective and objective results. Also, we conducted a Skype subjective conversation-opinion test on VDs specifically with the zero-client setup.

In the VDpilot study [47], VD performance was evaluated with 38 participants who assessed application QoE in comparison to performance in a physical desktop. Five applications were evaluated; the participants used their own devices to access the VDs over a WAN connection. In contrast, our study focused on zero-client performance and also quantified the impact of the network on QoE via subjective and objective studies.

Prior work on subjective Voice over IP (VoIP) performance evaluation was conducted with audio or video files fed into the call, without interaction with a person on the other side of the call [48]–[52]. This test is defined as the listening quality test [29]. This approach a lower degree of realism when compared to the more-complex conversation-opinion test. Very few studies have been conducted using the conversation-opinion test. For example, Cano and Cerdan [53] conducted a subjective study of Skype performance with real calls, but with a different purpose of comparing multiple VoIP applications. Khitmoh *et al.* [54] and Daengsi *et al.* [55] performed subjective studies on VoIP service where every two subjects had a 3-5 minute conversation in a controlled laboratory setup to develop a model for VoIP quality evaluation for the Thai language.

VII. CONCLUSIONS

Objective and subjective measurements were obtained to evaluate zero-client performance in which four activities were performed: (i) image viewing, (ii) 360-degree image exploration, (iii) video playback, and (iv) video-conference calling. Many objective metrics were used: Virtual Desktop Display Update Time (VD-DUT), Audio Quality (AQ) metrics, and Video Quality (VQ) metrics. VD-DUT, our newly defined metric, is measured by analyzing the network traffic between the zero client and the edge cloud, and was used to measure system responsiveness for the first two activities. Methodologies to measure AQ and VQ were also described, and experiments were conducted to measure the video quality of a playback application and the audio quality of a Skype call. The first large-scale subjective study on zero-client performance was conducted at the University of Virginia with 115 participants, in which Mean Opinion Score values were collected and analyzed. Network conditions were altered during the objective and subjective studies by increasing the packet loss rate (PLR).

The PLR impact on the zero-client performance varied based on the application. By analyzing the objective measurements, we found that VD-DUT increased and both AQ and VQ decreased when PLR increased, as expected. The video-playout application experienced the highest impact from packet losses as the video quality (measured by recv-PCoIP-fps) decreased by 71.8% when the PLR was changed from 0% to 10%. Statistical analysis conducted on the subjective measurements showed that the MOS values at 0.5% PLR and higher were not statistically significantly different, implying that the subjects interpreted the quality at 0.5% PLR and higher in a similar way. Video playout was the exception, where MOS values across different PLRs were significantly different and the impact of packet loss on quality of experience was the highest as MOS dropped from 3.53 (at 0% PLR) to 1.58 (at 3%), decreasing at a rate of 55.2%. A strong correlation between the objective and subjective measurements was found but the rate at which the objective and subjective measurements changed with increasing PLR differed depending on the application.

We plan to extend this work by executing a scalability study using our newly proposed metric, in which both network and computing resources are considered. A small experimental study is required to first characterize and model the network traffic and computing-resource usage. A subsequent simulation study can be carried out to quantify scalability.

ACKNOWLEDGMENT

The authors acknowledge Sam Nadjari, John Eshirow, and Paul Codjoe for their assistance in conducting the subjective trials. We also thank Clayton Banks, Bruce Lincoln and the Silicon Harlem team for their help in developing and testing the subjective testing protocols, and leading community engagement related to this technology. Thanks also to Teradici for providing licenses for these experiments.

REFERENCES

[1] VMware. (2014). *Key Considerations in Choosing a Zero Client Environment for View Virtual Desktops in VMware Horizon*. [Online]. Available: https://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/techpaper/vmware-top-five-considerations-for-choosing-a-zero-client-environment.pdf

[2] Teradici PCoIP. (2016). *Moulton College Deploys Teradici PCoIP Hardware Accelerator With Zero Clients to Deliver Flawless Video Performance While Sharply Decreasing the Management Burden for Its Endpoints*. [Online]. Available: http://www.teradici.com/docs/default-source/resources/case-studies/moulton_college_case_study.pdf

[3] Teradici PCoIP. (2014). *Lewiston Library Offers Patrons Quiet, High-Performance Zero Clients and Virtual Desktops*. [Online]. Available: https://www.teradici.com/docs/default-source/resources/case-studies/cs_cityof-lewiston-case-study.pdf

[4] Teradici PCoIP. (2014). *North Kansas City Hospital Delivers Secure, Point-of-Care Computer Access Anytime, Anywhere*. [Online]. Available: http://www.teradici.com/docs/default-source/resources/case-studies/nkch_case_study.pdf

[5] Teradici PCoIP. (2015). *PCoIP Zero Clients, Hardware Accelerators, & GPUs Match Graphic-Intensive Workloads for 2D/3D CAD*. [Online]. Available: https://www.teradici.com/docs/default-source/resources/case-studies/cs_construction_industry_case_study.pdf

[6] D. Brinkley, "Thin-clients in the classroom; software compatibility and a survey of systems," in *Proc. E-Learn, World Conf. E-Learn. Corporate, Government, Healthcare, Higher Educ.*, T. Reeves and S. Yamashita, Eds. Honolulu, HI, USA: AACE, Oct. 2006, pp. 383–390.

[7] N. Tolia, D. G. Andersen, and M. Satyanarayanan, "Quantifying interactive user experience on thin clients," *Computer*, vol. 39, no. 3, pp. 46–52, Mar. 2006.

[8] Technology Advice for Small Businesses. (2017). *How Thin and Zero Clients Save Money*. [Online]. Available: https://www.techadvisory.org/2017/07/how-thin-and-zero-clients-save-money/

[9] M. Remer. (2017). *Reduce Your I.T. Costs With Thin Clients*. [Online]. Available: https://www.businessmagazinegainesville.com/reduce-your-i-t-costs-with-thin-clients/

[10] B. Value. (2009). *The Pros and Cons of Thin Clients*. [Online]. Available: https://www.houkconsulting.com/2016/06/pros-cons-thin-clients/

[11] A. Wood. (2009). *Seven Deadly Sins When Deploying Thin Clients*. [Online]. Available: https://www.astroarch.com/tvp_strategy/seven-deadly-sins-when-deploying-thin-clients-1544/

[12] J. Rhee, A. Kochut, and K. Beaty, "DeskBench: Flexible virtual desktop benchmarking toolkit," in *Proc. IFIP/IEEE Int. Symp. Integr. Netw. Manage. (IM)*, Jun. 2009, pp. 622–629.

[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 Gener. Comput. Syst.*, vol. 81, pp. 478–491, Apr. 2018.

[14] Y. Sui, C. Yang, N. Jia, and X. Cheng, "vSIP: Virtual scheduler for interactive performance," in *Proc. ACM Int. Conf. Comput. Frontiers*, 2016, pp. 222–231.

[15] B. Song, M. M. Hassan, Y. Tian, M. S. Hossain, and A. Alamri, "Remote display solution for video surveillance in multimedia cloud," *Multimedia Tools Appl.*, vol. 75, no. 21, pp. 13375–13396, 2016.

[16] Y. Zhou, W. Tang, D. Zhang, and Y. Zhang, "Software-defined streaming-based code scheduling for transparent computing," in *Proc. Int. Conf. Adv. Cloud Big Data (CBD)*, Aug. 2016, pp. 296–303.

[17] M. A. Layek, T. Chung, and E.-N. Huh, "Adaptive desktop delivery scheme for provisioning quality of experience in cloud desktop as a service," *Comput. J.*, vol. 59, no. 2, pp. 260–274, 2016.

[18] *PCoIP*. Accessed: Nov. 2018. [Online]. Available: https://www.teradici.com/what-is-pcoip

[19] *Autoit*. Accessed: Nov. 2018. [Online]. Available: https://www.autoitscript.com/site/

[20] J. Nieh, S. J. Yang, and N. Novik, "Measuring thin-client performance using slow-motion benchmarking," *ACM Trans. Comput. Syst.*, vol. 21, no. 1, pp. 87–115, 2003.

[21] A. Berryman, P. Calyam, M. Honigford, and A. M. Lai, "VDBench: A benchmarking toolkit for thin-client based virtual desktop environments," in *Proc. IEEE 2nd Int. Conf. Cloud Comput. Technol. Sci. (CloudCom)*, Nov./Dec. 2010, pp. 480–487.

[22] J. You, U. Reiter, M. M. Hannuksela, M. Gabbouj, and A. Perkis, "Perceptual-based quality assessment for audio–visual services: A survey," *Signal Process., Image Commun.*, vol. 25, no. 7, pp. 482–501, Aug. 2010.

[23] D. Klatt, "Prediction of perceived phonetic distance from critical-band spectra: A first step," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, vol. 7, May 1982, pp. 1278–1281.

[24] R. Viswanathan, J. Makhoul, and W. Russell, "Towards perceptually consistent measures of spectral distance," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, vol. 1, Apr. 1976, pp. 485–488.

[25] A. Hines, J. Skoglund, A. Kokaram, and N. Harte, "ViSQOL: The virtual speech quality objective listener," in *Proc. Int. Workshop Acoust. Signal Enhancement (IWAENC)*, 2012, pp. 1–4.

[26] 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 *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, vol. 2, May 2001, pp. 749–752.

[27] *Assessment, Perceptual Objective Listening Quality*, document ITU-T Rec. P.863, 2011.

[28] *Clumsy 0.2*. Accessed: Nov. 2018. [Online]. Available: https://jagt.github.io/clumsy/index.html

[29] *Methods for Subjective Determination of Transmission Quality*, document ITU-T Rec. P.800, 1996.

[30] *Mean Opinion Score (MOS) Terminology*, document ITU-T Rec. P.800.1, 2016.

[31] *Methodology for the Subjective Assessment of the Quality of Television Pictures*, document ITU-R Rec. BT.500-13, 2012.

[32] W. Yu, J. Li, C. Hu, and L. Zhong, "Muse: A multimedia streaming enabled remote interactivity system for mobile devices," in *Proc. 10th Int. Conf. Mobile Ubiquitous Multimedia*, 2011, pp. 216–225.

[33] L. Cai, Z. Jia, Y. Qi, and L. Wang, "CloudRank-V: A desktop cloud benchmark with complex workloads," in *Proc. IEEE 10th Int. Conf. High Perform. Comput. Commun., IEEE Int. Conf. Embedded Ubiquitous Comput. (HPCC_EUC)*, Nov. 2013, pp. 415–421.

[34] N. Zeldovich and R. Chandra, "Interactive performance measurement with VNCPlay," in *Proc. USENIX Annu. Tech. Conf. (FREENIX) Track*, 2005, pp. 189–198.

[35] A. Pandey, L. Vu, V. Puthiyaveettil, H. Sivaraman, U. Kurkure, and A. Bappanadu, "An automation framework for benchmarking and optimizing performance of remote desktops in the cloud," in *Proc. Int. Conf. High Perform. Comput. Simulation (HPCS)*, Jul. 2017, pp. 745–752.

[36] S. Laine and I. Hakala, "H.264 QoS and application performance with different streaming protocols," in *Proc. 8th Int. Conf. Mobile Multimedia Commun.* Chengdu, China: ICST, 2015, pp. 32–38.

[37] 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 *Proc. IEEE INFOCOM*, Mar. 2010, pp. 1–9.

[38] R. K. P. Mok, E. W. W. Chan, and R. K. C. Chang, "Measuring the quality of experience of HTTP video streaming," in *Proc. IFIP/IEEE Int. Symp. Integr. Netw. Manage. (IM)*, May 2011, pp. 485–492.

[39] T. Abar, A. Ben Letaifa, and S. El Asmi, "Objective and subjective measurement QoE in SDN networks," in *Proc. 13th Int. Wireless Commun. Mobile Comput. Conf. (IWCMC)*, Jun. 2017, pp. 1401–1406.

[40] B. P. Bondzulic, B. Z. Pavlovic, V. S. Petrovic, and M. S. Andric, "Performance of peak signal-to-noise ratio quality assessment in video streaming with packet losses," *Electron. Lett.*, vol. 52, no. 6, pp. 454–456, Mar. 2016.

[41] F. Battisti, M. Carli, and P. Paudyal, "QoS to QoE mapping model for wired/wireless video communication," in *Proc. Euro Med Telco Conf. (EMTC)*, Nov. 2014, pp. 1–6.

[42] Z. Duanmu, K. Zeng, K. Ma, A. Rehman, and Z. Wang, "A quality-of-experience index for streaming video," *IEEE J. Sel. Topics Signal Process.*, vol. 11, no. 1, pp. 154–166, Feb. 2017.

[43] Z. Deng, Y. Liu, J. Liu, X. Zhou, and S. Ci, "QoE-oriented rate allocation for multipath high-definition video streaming over heterogeneous wireless access networks," *IEEE Syst. J.*, vol. 11, no. 4, pp. 2524–2535, Dec. 2017.

[44] J. Pokhrel, B. Wehbi, A. Morais, A. Cavalli, and E. Allilaire, "Estimation of QoE of video traffic using a fuzzy expert system," in *Proc. IEEE Consum. Commun. Netw. Conf. (CCNC)*, Jan. 2013, pp. 224–229.

[45] M. Jarschel, D. Schlosser, S. Scheuring, and T. Hoßfeld, "An evaluation of QoE in cloud gaming based on subjective tests," in *Proc. 5th Int. Conf. Innov. Mobile Internet Services Ubiquitous Comput. (IMIS)*, Jun./Jul. 2011, pp. 330–335.

[46] P. Casas, M. Seufert, S. Egger, and R. Schatz, "Quality of experience in remote virtual desktop services," in *Proc. IFIP/IEEE Int. Symp. Integr. Netw. Manage. (IM)*, May 2013, pp. 1352–1357.

[47] P. Calyam, A. Berryman, D. Welling, M. Saravanan, R. Ramnath, and J. Ramanathan, "VDPilot: Feasibility study of hosting virtual desktops for classroom labs within a federated university system," *Int. J. Cloud Comput.*, vol. 3, no. 2, pp. 158–176, 2014.

[48] W.-H. Chiang, W.-C. Xiao, and C.-F. Chou, "A performance study of VoIP applications: MSN vs. skype," in *Proc. MULTICOMM*, 2006, pp. 13–18.

[49] G. Lisha and L. Junzhou, "Performance analysis of a P2P-based VoIP software," in *Proc. Int. Conf. Internet Web Appl. Services/Adv. Int. Conf. Telecommun. (AICT-ICIW)*, Feb. 2006, p. 11.

[50] P. Calyam, E. Ekici, C.-G. Lee, M. Haffner, and N. Howes, "A 'GAP-model' based framework for online VVoIP QoE measurement," *J. Commun. Netw.*, vol. 9, no. 4, pp. 446–456, 2007.

[51] N. Kushman, S. Kandula, and D. Katabi, "Can you hear me now?!: It must be BGP," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 37, no. 2, pp. 75–84, 2007.

[52] M. Goudarzi, L. Sun, and E. Ifeachor, "Modelling speech quality for NB and WB SILK codec for VoIP applications," in *Proc. 5th Int. Conf. Next Gener. Mobile Appl., Services Technol. (NGMAST)*, Sep. 2011, pp. 42–47.

[53] M.-D. Cano and F. Cerdan, "Subjective QoE analysis of VoIP applications in a wireless campus environment," *Springer Telecommun. Syst.*, vol. 49, no. 1, pp. 5–15, Jan. 2012.

[54] N. Khitmoh, P. Wuttidittachotti, and T. Daengsi, "A subjective—VoIP quality estimation model for G.729 based on native Thai users," in *Proc. 16th Int. Conf. Adv. Commun. Technol.(ICACT)*, Feb. 2014, pp. 48–53.

[55] T. Daengsi, P. Wuttidittachotti, C. Strategy, C. Wutiwiwatchai, and S. Sukparungsee, "VoIP quality of experience: A proposed subjective MOS estimation model based-on Thai users," in *Proc. 5th Int. Conf. Ubiquitous Future Netw. (ICUFN)*, Jul. 2013, pp. 407–412.

FATMA ALALI received the B.S. degree in computer engineering from Kuwait University, in 2011, and the M.S. and Ph.D. degrees in computer engineering from the University of Virginia, in 2016 and 2019, respectively. She is currently an Assistant Professor with the Department of Computer Engineering, Kuwait University. Her research interests include software-defined networks, enabling virtual circuits (VC) services, infiniband networks, and cloud computing.

TASHA A. ADAMS was born in Colorado. She received the bachelor's degree in optical engineering from Norfolk State University, Norfolk, VA, USA, in 2013, and the master's degree in optical science from The University of Arizona, Tucson, AZ, USA, in 2018, where she is currently pursuing the Ph.D. degree in optical science.

In the past, she was an Intern with the National Oceanic and Atmospheric Administration and the Air Force Research Laboratory. She is a Student Member of the OSA.

RIDER W. FOLEY received the bachelor's degree in environmental science from The University of New Hampshire, the master's degree in environmental management from Harvard University, and the Ph.D. degree in sustainability from Arizona State University. He is currently an Assistant Professor of science, technology, and society program with the Department of Engineering and Society, The University of Virginia. He is the Co-PI of the recently funded NSF Project titled A Novel Architecture for Secure, Energy-Efficient Community-Edge-Clouds With Application in Harlem (SEEC HARLEM) with partners in Harlem, The University of Arizona, and Fordham University. He is a Research Collaborator of the Sustainability Science Education Program with the Biodesign Institute.

DAN KILPER received the Ph.D. degree in physics from the University of Michigan, in 1996. He is currently the Director of the Center for Integrated Access Networks and a Research Professor with the College of Optical Sciences, The University of Arizona, Tucson, where he holds a joint appointment in electrical and computer engineering, an Adjunct Professorship with the College of Engineering, Trinity College Dublin, and an adjunct faculty position in electrical engineering with Columbia University. From 2000 to 2013, he was a Member of Technical Staff with Bell Labs. He holds seven patents. He has authored five book chapters and more than one hundred fifty peer-reviewed publications. He is on the Steering Committee of the IEEE Sustainable ICT Initiative. He has given plenary or keynote presentations at the IEEE Green Communications Conference 2011, E-Energy Conference 2011, and the IEEE/IFIP ONDM 2017. He has served as the General Chair for the IEEE Green Communications Conference 2014 and 2015 and a Technical Program Committee Co-Chair for the IEEE/OSA Photonics in Switching 2013. He is the TPC Chair of Photonics in Switching and Computing 2019. He has organized workshops and served on technical program committees at numerous international conferences including CLEO/IQEC, OFC, INFOCOM, ICC, GLOBECOM, ICTON, IFIP, CLEO Europe, and COIN/ACOFT. His research is aimed at solving fundamental and applied problems in communication networks in order to create a faster, more affordable, and energy efficient Internet, addressing interdisciplinary challenges for smart cities, sustainability, and digital equality. His work has been recognized with the Bell Labs President's Gold Medal Award. He has served on the Bell Labs President's Advisory Council on Research. He is an Editor of the IEEE Transactions on Green Communications and Networking (TGCN) and the IEEE/CiC China Communications.

RONALD D. WILLIAMS received the B.S. and M.S. degrees in electrical engineering from the University of Virginia and the Ph.D. degree from the Massachusetts Institute of Technology. He is a Faculty Member of the Department of Electrical and Computer Engineering, University of Virginia. He has authored over 100 technical papers based on his research. He is an Inventor on six U.S. patents. His research and teaching is primarily in the areas of embedded computing with biomedical and infrastructure applications.

MALATHI VEERARAGHAVAN received the B.Tech. degree from IIT Madras, and the M.S. and Ph.D. degrees from Duke University.

She is currently a Professor with the Charles L. Brown Department of Electrical and Computer Engineering, University of Virginia. 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 is currently a Professor of electrical and computer engineering. Her research work has been primarily in high-speed networking, wireless networking, and network security. Her research funding has been mainly from the National Science Foundation, the U.S. Department of Energy, and DARPA. She holds 30 patents. She has over 138 publications. She has 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 has received six best-paper awards. She is currently an Associate Editor of the IEEE/ACM Transactions on Networking. She was as an Associate Editor of the IEEE Transactions on Reliability, from 1992 to 1994.