End-to-end Characterization of Game Streaming Applications on Mobile Platforms

SANDEEPA BHUYAN, The Pennsylvania State University, USA
SHULIN ZHAO, The Pennsylvania State University, USA
ZIYU YING, The Pennsylvania State University, USA
MAHMUT T. KANDEMIR, The Pennsylvania State University, USA
CHITA R. DAS, The Pennsylvania State University, USA

With the advent of 5G, supporting high-quality game streaming applications on edge devices has become a reality. This is evidenced by a recent surge in cloud gaming applications on mobile devices. In contrast to video streaming applications, interactive games require much more compute power for supporting improved rendering (such as 4K streaming) with the stipulated frames-per second (FPS) constraints. This in turn consumes more battery power in a power-constrained mobile device. Thus, the state-of-the-art gaming applications suffer from lower video quality (QoS) and/or energy efficiency. While there has been a plethora of recent works on optimizing game streaming applications, to our knowledge, there is no study that systematically investigates the $< QoS, Energy >$ design pairs on the end-to-end game streaming pipeline across the cloud, network, and edge devices to understand the individual contributions of the different stages of the pipeline for improving the overall QoS and energy efficiency.

In this context, this paper presents a comprehensive performance and power analysis of the entire game streaming pipeline consisting of the server/cloud side, network, and edge. Through extensive measurements with a high-end workstation mimicking the cloud end, an open-source platform (Moonlight-GameStreaming) emulating the edge device/mobile platform, and two network settings (WiFi and 5G) we conduct a detailed measurement-based study with seven representative games with different characteristics. We characterize the performance in terms of frame latency, QoS, bitrate, and energy consumption for different stages of the gaming pipeline. Our study shows that the rendering stage and the encoding stage at the cloud end are the bottlenecks to support 4K streaming. While 5G is certainly more suitable for supporting enhanced video quality with 4K streaming, it is more expensive in terms of power consumption compared to WiFi. Further, fluctuations in 5G network quality can lead to huge frame drops thus affecting QoS, which needs to be addressed by a coordinated design between the edge device and the server. Finally, the network interface and the decoder units in a mobile platform need more energy-efficient design to support high quality games at a lower cost. These observations should help in designing more cost-effective future cloud gaming platforms.

CCS Concepts: • **Networks** → **Mobile networks**; • **Computer systems organization** → **Client-server architectures**; • **Computing methodologies** → *Rendering*; • **Hardware** → Emerging technologies.

Additional Key Words and Phrases: cloud gaming; 5G ; smartphones; performance; energy efficiency

Authors' addresses: Sandeepa Bhuyan, sxb392@psu.edu, The Pennsylvania State University, W340 Westgate Building, University Park, Pennsylvania, USA; Shulin Zhao, suz53@psu.edu, The Pennsylvania State University, University Park, USA; Ziyu Ying, zjy5087@psu.edu, The Pennsylvania State University, University Park, USA; Mahmut T. Kandemir, mtk2@psu.edu, The Pennsylvania State University, University Park, USA; Chita R. Das, cxd12@psu.edu, The Pennsylvania State University, University Park, USA.

2476-1249/2022/3-ART10 $15.00
https://doi.org/10.1145/3508030

ACM Reference Format:
Sandeepa Bhuyan, Shulin Zhao, Ziyu Ying, Mahmut T. Kandemir, and Chita R. Das. 2022. End-to-end Characterization of Game Streaming Applications on Mobile Platforms. *Proc. ACM Meas. Anal. Comput. Syst.* 6, 1, Article 10 (March 2022), 25 pages. https://doi.org/10.1145/3508030

1 INTRODUCTION

Over the past five decades, the video game industry has evolved tremendously – from arcade game stores to home-consoles to PC and handheld consoles and now to mobile gaming. A zeal for creation of realistic graphics for consumer entertainment, products and architecture visualization, and medical applications has motivated research advancement in computer graphics techniques and graphics acceleration hardware, which in turn has led to an exponential growth in the video gaming industry with more than 2.7 billion gamers worldwide in 2020 [61]. Further, a recent Verizon study reveals that in 2020, during the pandemic, the consumption of video games has increased by 115% in the USA [51]. However, enjoying video games with high graphics fidelity on desktop PCs requires high-end graphics cards. Exacerbating this requirement, the Covid-19 pandemic has induced a global semiconductor chip scarcity, thus making it difficult to procure high-end graphics cards or gaming consoles [46]. To salvage the situation, a myriad of recently emerged cloud gaming services, such as the NVIDIA GeForce Now [1], Google Stadia [27], Sony PlayStation Now [67], Microsoft xCloud [49], and Amazon Luna [2], have taken gaming to a new-level by allowing access to powerful cloud servers and offering Games-as-a-Service (GaaS) [71], encompassing hundreds of games. This has incentivized people to move towards cloud gaming services. According to a Newzoo report, the global cloud gaming market revenue is forecasted to reach $6.53 billion in 2024 [19].

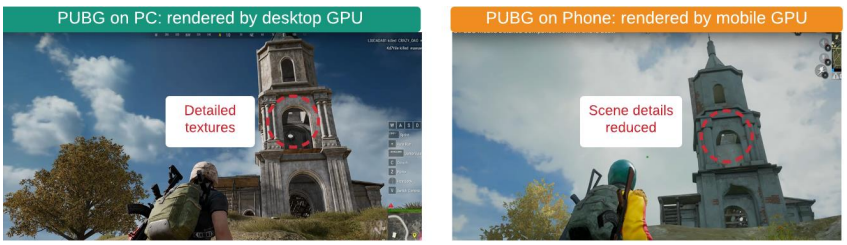

Fig. 1. Two versions of PUBG game: on a PC (left) and on a mobile device (right).

Additionally, cloud gaming services[1] have proffered the perks of enjoying graphically-intensive games on the mobile platforms, which was not possible earlier. Although a few popular PC games such as Fortnite [21], PUBG [42], and Minecraft [50] have been ported to mobile platforms, due to a limited power budget and resource constraints, the mobile versions typically offer diminished graphics effects compared to their PC versions. As seen in Fig. 1, compared to a desktop PC, the PUBG mobile[41] version has noticeably reduced scene details. Moreover, mobile GPUs cannot handle advanced graphics-intensive techniques such as raytracing (which incorporates realistic lighting in the frame by simulating the physical behavior of light). Such advanced graphics laden games that offer an enhanced and realistic user experience can be provided through cloud gaming services. Unfortunately, a major challenge for the cloud gaming applications is their high network bandwidth demand and their latency sensitivity. Even using high-speed WiFi to access these services leads to frame drops at higher bitrates. Luckily, the introduction of 5G has broken the barriers of internet speed offered via WiFi, delivering high-speed broadband internet to the masses. This is

[1]We refer cloud gaming services and game streaming applications interchangeably throughout this paper.

positioned to revolutionize 4K UHD gameplay on mobile platforms leveraging the cloud gaming services.

However, the performance perks offered by 5G comes at the price of increased battery drain, which is a major drawback for mobile platforms that are resource-constrained and have limited power budgets. As an example, around an hour of 4K gameplay on 5G using cloud gaming services on mobile platforms drains around ≈ 36% of battery on a Pixel 5 phone. Due to the "real-time" and "interactive" nature of these applications, prior video streaming energy optimizations to reduce network energy drain [6, 66] are not directly applicable to cloud gaming. Furthermore, for an immersive experience, maintaining a decent *Quality of Service (QoS)* is also a crucial necessity during gameplay. Our preliminary profiling shows that playing 4K UHD games at times suffers from low frame rate. Since these are interactive applications, the bottleneck could be in the i) server, ii) network, or iii) client mobile platform. Therefore, it is first critical to understand the end-to-end performance and energy behavior of gaming applications to aid in designing the next generation edge devices for providing acceptable QoS, while improving the energy efficiency. Isolated characterizations of network performance with 4G, WiFi and 5G [54] and mobile platform performance [17, 39] have been conducted in prior research. However, to our knowledge, there is *no* holistic characterization of gaming applications – including server, network, and edge device – to shed light on the current bottlenecks from performance and energy consumption standpoints. Such a characterization study is essential for mitigating the current bottlenecks in cloud gaming pipelines.

Towards this, in this paper we conduct an extensive end-to-end study of cloud gaming apps on mobile platforms by examining the effects of various configurable parameters across the cloud, network and end user platforms on QoS as well as energy. We use seven game streaming applications to conduct a measurement-based study to analyze two important parameters – performance and power consumption. For the performance analysis, we measure latency of different stages of the gaming pipeline, QoS (frame rate and image quality in terms of signal-to-noise-ratio (PSNR)) and, for the energy estimation, we primarily measure the energy consumption of different stages.

Our in-depth evaluation of the entire game streaming pipeline has led to the following observations: (i) On the cloud/server end, the rendering stage and the encoding stage consume a significant amount of time and often misses the target 16.66 ms (60 FPS) frame deadline for several applications. This suggests minimizing the latency of these two stages through novel hardware and software optimizations. (ii) WiFi and 4G networks are capable of supporting 1080p quality video games to a great extent without significant frame drops. However, the frame drops increase significantly for 4K videos. Due to high throughput/bandwidth, as expected, 5G networks can support 4K video games. However, playing high-quality 4K games at higher bitrates on the 5G networks can still suffer significant frame drops. Also, 5G signal strength is reported to be highly fluctuating by various studies [5, 40, 64]. This is because the 5G signals operate at high GHz range frequencies (> 20 GHz), and hence suffer higher path and propagation loss as the 5G signals cannot penetrate solid objects such as cars, trees, and walls as easily as 4G, which operates below 2 GHz frequency. This results in high variation in 5G network throughput, thus impacting video quality. Thus, a coordinated design to support variable bit rate between the server and an edge device for real-time and interactive applications (e.g., cloud gaming) as well as optimizations in the 5G networks' backend to support high-bitrate game streaming is essential to utilize 5G more effectively. (iii) On the mobile end, the energy consumption of the 5G network module is significantly higher (about 36% more compared to WiFi on our Pixel 5 phone) followed by the decoding stage, thereby making it less energy-efficient. This in turn needs a fresh look at minimizing the compute and memory energy consumption in addition to what has been proposed in the context of video streaming applications [17, 77]. This is non-trivial, but essential to enhance the battery life for supporting such applications.

2 BACKGROUND AND MOTIVATION

In this section, we first illustrate the end-to-end game streaming pipeline, compare it against the video streaming pipeline, and highlight the main differences between the two. We then present the performance and energy-inefficiencies of the game streaming applications on the mobile platforms to motivate the need for a detailed and thorough end-to-end characterization study of game streaming applications, that would identify bottlenecks in the application pipeline. Finally, we also discuss the various types of gaming workloads used in our study.

2.1 Game Streaming Pipeline

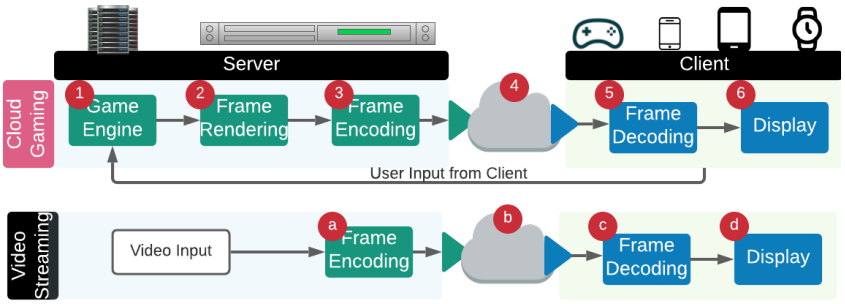

Fig. 2. A typical cloud gaming pipeline (top) and a video streaming pipeline (bottom).

In a typical game streaming pipeline (illustrated in the top half of Fig. 2), the game engine (shown in ①), which comprises the game logic, evaluates the "next state" of the game environment and invokes graphics API calls (more specifically, draw calls) to generate the next frame at a preset frame-rate (assuming a dynamic game environment). On reception of a user input from the client, the game engine integrates it into the next game state evaluation and frame generation. A draw call is a graphics API command that contains information regarding the rendered objects, textures, shaders, etc., in a frame. These draw calls are handled by the GPU for frame rendering (shown in ②). As a raw rendered frame size can be around $\approx 8Mpixels * 3BytesPerPixel = 24MBytes$, it is *compressed* using the frame encoding stage (shown in ③) to prepare it for transmission to the client. Before the transmission of the encoded frame over the network (shown in ④) to the client, the frame is *pre-processed* to fit into network packets either by splitting the frame across packets or by combining a few frames into a single packet (depending on the encoded frame size). Finally, the packets are *prepended* with routing headers and *dispatched* over the network. At the client, the received packets are first *processed* to extract the actual encoded video frame data by stripping the headers and assembling all the payload information from every packet associated with the encoded video frame. The encoded video frame is then *decoded* and *stored* in the framebuffer in the frame decoding stage (shown in ⑤). Finally, the display processor, which probes the framebuffer at its refresh rate, *reads* the pixel information from the framebuffer and *displays* it on the client's screen.

Video streaming pipeline comparison: While in a typical video streaming application (shown in the bottom half of Fig. 2), when the client requests to watch a video, the frames encoded in the frame encoding stage (shown in ⓐ) follow the same subsequent stages as the game streaming pipeline to process the received frame and decode the encoded video frame data (shown in ⓒ) before displaying on the client's screen (ⓓ). Although the pipeline stages at the client device are similar for both the applications, the performance and energy-efficiency optimizations for the well-studied video streaming applications [6, 29, 35, 43, 66] are *not* directly applicable to the game streaming application. This is because, unlike video streaming where the videos uploaded on the server can

be pre-encoded and maintained at the server and reused for several clients (users) upon request, the video game frames *cannot* be pre-rendered and encoded as they heavily depend on the user input which varies from user to user. Thus, the interactive and immersive experience offered by the game streaming application imposes a "real-time" frame generation (rendering) and encoding pressure on the server.

4K UHD gameplay: In the case of video streaming, watching a 4K UHD resolution video can be bandwidth heavy as it requires larger frame sizes to be transmitted to the user. In contrast, playing games through the game streaming services at 4K UHD resolution is not only bandwidth demanding but is also latency-sensitive failing which might lead to a "Game Over" scenario for the user, thus degrading the user's *Quality of Experience (QoE)*. As a result, game streaming applications are most popular among users utilizing a desktop/laptop/game console (xCloud on Xbox) equipped with a stable and high-speed internet connection.

Game streaming on mobile platforms: Due to the aforementioned reasons, game streaming applications lack widespread adoption by mobile users. However, the emergence of 5G mmWave network has broken the barrier of current broadband data speed, thus making high bandwidth and low latency connections available to the masses along with the added benefit of mobility, in contrast to WiFi and Ethernet LAN connections. Hence, 5G will be a major driver for adoption of game streaming services on mobile platforms and will eventually lead to more game streaming services being available for Android and iOS mobile platforms. Moreover, the ongoing Covid-19 pandemic has induced a worldwide semiconductor chip shortage [53], leading to a scarcity of graphics cards and gaming consoles (Playstation 5 and Xbox Series X). This has driven more users towards adoption of cloud gaming services [65].

2.2 Challenges

Although game streaming is an interesting use case for mobile platforms, there are several critical performance as well as energy efficiency concerns that mostly root from the real-time and interactive nature of these applications. Addressing these concerns can potentially result in more widespread adoption of game streaming. We list below several of the challenges.

Faster battery drain: The high bandwidth and low latency perks of 5G come at the expense of energy. For instance, just one hour of game streaming application usage on a Pixel 5 phone using 5G can drain ≈36% of its battery life, compared to ≈18% on WiFi. This 2× battery drain on 5G compared to WiFi is a heavy price to pay for the resource-constrained and power budget limited mobile platforms. Thus, it is imperative to first understand how much benefit we can reap from leveraging 5G and at what cost. Further, the most widely adopted optimizations for video streaming applications to reduce network system processing energy on client platforms such as pre-fetching and batch downloading future videos frames [6, 66] cannot be directly leveraged for these video game streaming applications. This is due to the following reasons: ⓐ In these game applications, current user input decides the subsequent frames to be generated by the game server and how frequent the user input is sent to the server varies by game. For example, for a shooter game, the user can move around very quickly, thus sending each input event to the server at a higher rate. In the case of a turn-based strategy game, the game will not register new user inputs unless it is the user's turn to take actions. ⓑ Further, since the user input events are non-deterministic in nature, and the client demands and receives the frames at an intended frame rate set for any particular game, the network subsystem can not be put into a sleep or low-power mode when using game streaming applications.

QoS degradation: The two main QoS parameters that affect a user's gameplay experience are *Frame per Second (FPS)* and *Image Quality*. Our preliminary measurements show that, for a few games, the client views the frame at ≈32 FPS. At first glance, one could hypothesize that the network

Table 1. Game Workloads

Game	Scene Details	Scene Variation/ Camera Motion	Description
Destiny 2 [9]	High	High	Shooter game; detailed textures
Return to Castle Wolfenstein [8]	Low	High	Shooter game; less textures
Skyrim Special Edition [7]	High	Low	Role playing game
Age of Empires-II: Definitive Edition [47]	Low	Low	Real time strategy; less textures
Age of Empires-III: Definitive Edition [48]	High	Low	Real time strategy; detailed textures
Hollow Knight [69]	Low	Low	Platform; dark background
Dead Cells [52]	Low	Low	Platform; vivid background

connection may be the bottleneck. But, owing to the real-time nature of the application, (recall the game streaming pipeline in Fig. 2) any stages in the pipeline prior to the network could also potentially trigger low frame rates and at times even the stages on the client device may be the culprit. Hence, addressing such QoS degradation concerns for game streaming on mobile devices requires a detailed end-to-end study of the pipeline to i) identify the bottlenecks and ii) propose optimizations to enhance QoS. Further, the tolerance limit of the QoS parameters can be different for different games. For example, a user playing a shooter game mostly moves quickly inside the game environment and thus may not be able to discern the loss in image quality of the frames, whereas a user playing a role playing game might be able to notice a (human-eye perceptible) comparable loss in image quality.

Large search space: There are many configurable/modifiable parameters affecting the QoS as well as energy consumption, such as graphics rendering quality, encoding quality and encoding bitrate settings, frame resolution, game nature, etc., thus making the search space immense. Investigating these QoS metrics and energy expenditure via a thorough parameter sweep is a non-trivial effort.

2.3 Salient Characteristics of Gaming Workloads

In Sec.2.2, we discussed how inherent game characteristics or features can impose different QoS requirements. To further examine them, we choose seven representative gaming workloads, listed in Table 1, across a broad range of video game genres as well as based on amount of texture/information details in the game scene and frequency of scene variation. A brief description of the selected gaming workloads is provided below.

- *Destiny 2 and Return to Castle Wolfenstein*: Both of these are shooter (first person view) games which involve high scene variation or camera motion/panning; Destiny 2 has much more detailed graphics compared to Return to Castle Wolfenstein.
- *Skyrim*: Skyrim is another popular texture-rich role playing game with a requirement of less frequent scene variation than shooter games.
- *Age of Empires-II (AoE-II) and Age of Empires-III (AoE-III)*: Both of these games are real-time strategy games. While AoE-II uses 2D rendering, AoE-III utilizes 3D rendering and is much more texture-rich compared to AoE-II.
- *Hollow Knight and Dead cells*: Both these games are platform games which involve skillful movement of the player character between points in a rendered environment. Dead cells presents vivid backgrounds, whereas Hollow Knight's background colors are mostly dark and muted. Additionally, Hollow Knight uses 2D layered assets to create a perception of 3D environment in the game.

Furthermore, we believe different games will respond differently to the modifications or changes in the different stages of the game streaming pipeline. Therefore, we choose a variety of gaming workloads mentioned above to conduct a comprehensive study.

3 METHODOLOGY

In this section, we first provide a brief overview of various cloud gaming services. To study the game streaming pipeline stages and inspect the bottlenecks for QoS and energy consumption (as discussed in Sec. 2), we state our platform of choice. Next, we illustrate the detailed working of our platform with respect to the game streaming pipeline. And, finally, we present our experimental setup for data collection and analysis.

3.1 Cloud Gaming Platforms

To identify the bottlenecks in the game streaming pipeline, an extensive study is necessary at both the server and the client ends. Over the past few years, various cloud gaming services, such as NVIDIA GeForce Now [1], Google Stadia [27], Amazon Luna [2], Microsoft xCloud [49], etc. have emerged that offer Games-as-a-Service (GaaS). Fig. 3 shows a high-level overview of such cloud gaming platforms where the game engine, frame rendering and encoding are hosted by the cloud servers. However, the proprietary nature of such platforms makes them implausible for research usage. Moreover, the Android client applications on mobile for each of these services do not yet support 4K UHD resolution. Further, Gaming Anywhere [30, 31], an open-source cloud gaming system, which has been extensively used in the past [11] for cloud gaming research has been discontinued.

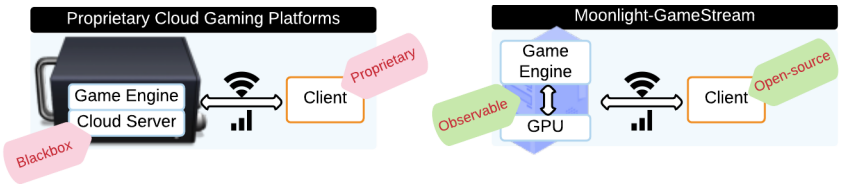

Fig. 3. Proprietary cloud gaming platforms (left) vs. a Moonlight-client based GameStream platform (right).

Apart from the above platforms, NVIDIA provides a game streaming service, called GameStream [57], which utilizes its GeForce graphics cards and streams to NVIDIA Shield devices. An open-source [12] client application named Moonlight-Stream [13] is available that mimics the client side protocols of this streaming service and can run on multiple platforms, such as Microsoft Windows, Linux, Android, etc. On the server side, the GeForce graphics card is utilized in frame rendering and its hardware video encoders encode the frames before transmitting to the client. The Moonlight-Stream client uses the hardware decoders of the client device to decode the received frames and transmit player input back to the server. The above game streaming pipeline exposes more details than the proprietary cloud gaming services discussed earlier. Hence, for our evaluation, we use Moonlight-Stream (referred to as Moonlight) as the client and a desktop equipped with a GeForce GPU as our cloud gaming server. Henceforth, we refer to our game streaming evaluation platform as *GameStream* platform.

3.2 GameStream Evaluation Platform Across Server and Client

Fig. 4 illustrates the pipeline stages in our GameStream platform. The server and the client communicate via the NVIDIA GameStream protocol. The NVIDIA GeForce Experience application installed on the server handles the server side communication, while the Moonlight-Stream app present on the mobile device does so for the client side. When playing the games on the client devices, the user inputs captured by the client device's touch screen/peripherals are sent to the NVIDIA GeForce Experience application on the server, which in turn forwards them to the game engine. Based on the user inputs, the game engine logic invokes graphics API calls for rendering the next frame. Frame rendering takes place on the GPU. The rendered frame is then encoded using

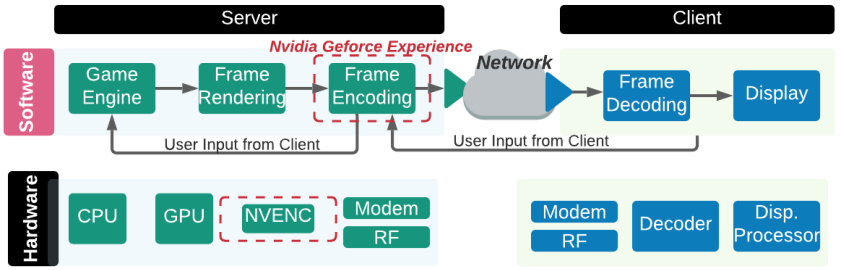

Fig. 4. GameStream platform.

the NVENC cores, which are high speed, low latency hardware video encoding cores present on the NVIDIA GPUs, at a desirable bitrate setting requested by the client, and the encoded frame is transmitted to the client over the network. The client receives the frame and uses its hardware decoder to decode the received frames and then finally displays the decoded frame on the client device screen.

3.3 Experimental Methodology

Hardware Infrastructure: The cloud server was set up using an Intel i7-5820K desktop processor equipped with an NVIDIA GeForce GTX 980 graphics card. The detailed system-level trace collection on a mobile platform required superuser rights on the mobile device. Further, testing on the high-speed 5G connection required the support of mmWave high-frequency band by the mobile phones. Currently, there are very limited phone models that fit both these criteria. Hence, Google Pixel 5 was chosen as the client mobile platform. In order to set up communication with the cloud desktop server over the internet, the Moonlight Internet Hosting tool [14] can be used. However, depending on the network configuration, ISP, firewalls, etc., port-forwarding may not be possible for the server in some cases, which happened in our scenario. Hence, we used ZeroTier VPN [76] to setup the internet communication between our desktop server and client device.

Software Setup: Our server was configured for a detailed pipeline stage analysis. First, the gaming workloads were installed on our server and shortcuts were added to NVIDIA GeForce Experience [56], exposing them to the client UI.

Frame Rendering Time: For measuring the rendering time at the server for each frame, FrameView software [3] was used, which can capture frame data from most Graphics APIs such as Vulkan, OpenGL, DirectX 9, 10, 11, and 12.

Frame Encoding Time: The NVIDIA GeForce Experience uses the NVENC hardware encoding cores on the NVIDIA GPUs for encoding the rendered frames. As NVIDIA GeForce Experience is proprietary, we encoded the game frames using the open-source ffmpeg [24] software instrumented to utilize the hardware encoding NVENC cores on the NVIDIA GPUs and measured the frame times. During the gameplay, the rendered game frames were captured from the frame buffer in the GPU memory using the NVIDIA ShadowPlay software [58], which performs hardware-accelerated frame capturing from the GeForce GPUs. The captured frames were then fed to the ffmpeg tool that uses the NVENC encoding cores to generate the encoded frames at desired encoding settings such as encoding quality, bit-rate, and encoding speed.

Frame Decoding Time: The Moonlight client interface source code was annotated to capture the latency stats as well as the received game frame sizes on the client device.

Mobile Platform Power Measurement: Until a few years ago, mobile devices had removable batteries and performing hardware power measurement using the Monsoon Power monitor [15, 29, 70, 75, 78] by replacing the phone battery with the power supply probes from the Monsoon power monitor

was a feasible task. However, most of the current smartphones are manufactured with the phone back cover tightly integrated into the phone's main body, thus making it difficult to disassemble the phone and replace the battery without potentially damaging the phone's internal circuitry in the process. Further exacerbating the situation, 5G mmWave network services are still in the deployment phase. As a result, mmWave service availability is limited to a very few locations. Our 5G experiments were conducted outdoors and the client device was held stationary with clear Line of Sight (LoS) to the 5G base-station. Using the Monsoon Power Monitor under such conditions to measure the device power out in the wild is a very challenging task as it would require carrying a big and heavy power bank to power the Monsoon Power Monitor which in turn would power the mobile phone, and at the same time run the gaming workloads and collect statistics on the phone. Due to these challenges, we chose to use a software power measurement approach to collect the power readings from the voltage and current values exposed by the Android Kernel at the path */sys/class/power_supply/battery/* on the mobile device. To collect the power readings, the termux application [26] (a terminal emulator for Android) was used to run the scripts on the device while running the game streaming applications on them.

For profiling the games, we repeated the gameplay of each game for the same time duration over multiple iterations starting at the same game state (save point) in each game for each of its gameplays, while keeping the user input events as similar as possible for each game and reported the averaged values. This was done to ensure the reproducibility of the results. However, owing to the non-deterministic nature of the game environment due to subsystems such as the physics simulation (such as particles, wind, waves, ragdoll, etc.), enemy AI and dynamic nature of other human players (in multiplayer games), capturing user input events and replaying through a macro software didn't yield exactly repeatable results for each round. Hence, we chose to do this manually for all the gameplays as described above and sidestepped the additional overhead of the macro tool. Furthermore, as capturing metrics across different stages of the pipeline over the single run could potentially cause interference with other metrics, we isolated the profiling of different stages to avoid any interference between our logging mechanism and the application performance. Also, in order to ensure that there was no additional factors affecting our readings, before each run, we made sure that there was enough cool down period to avoid thermal throttling and monitored the network throughput (≈1Gbps), signal strength (> -85dBm), and battery temperature (< 77°F when the mobile device is idle) for a window of 10 seconds.

4 CHARACTERIZING CLOUD GAMING THROUGH PARAMETER SWEEPING

As discussed in Sec. 2, a large number of parameters affect the QoS and energy behaviors of the cloud gaming pipeline. These parameters span across servers on the cloud (e.g., rendering graphic setting, encoding quality), network types (e.g., WiFi, 4G, 5G), as well as the client on the edge (e.g., battery percentage, network conditions, etc.). In this section, we perform a thorough parameter sweep to investigate the inefficiencies in the current cloud gaming pipeline, including frame rate (in Sec. 4.1.1), image quality (in Sec. 4.1.2), power and energy efficiency with respect to the network type (in Sec. 4.2.1, bitrate (in Sec. 4.2.2) and various games (in Sec. 4.2.3).

4.1 QoS

As mentioned in Sec. 2, frame rate and image quality are the two crucial metrics that help quantify the QoS for the cloud gaming applications. Next, we strive to answer the QoS related question: *Which parameters affect the QoS and how?*

4.1.1 Frame Rate. Note that as video games usually involve fast motions, 60 FPS (frames per second) frame rate (16.66 ms per frame) is necessary to avoid human-eye perceptible stutters while playing games for a smooth, responsive and enjoyable experience [37]. In order to determine

whether each individual component/stage presented in the game streaming pipeline (shown in Fig. 2) is able to generate frames at a throughput of 60 FPS or not, we start by studying the effect of network types on the frame rate as the network connection partitioning the cloud-end and the client-end plays a critical role in catering and maintaining a good user experience during streaming from the cloud to the client.

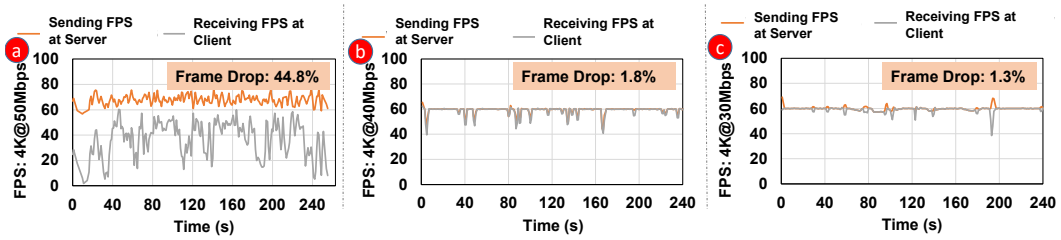

Fig. 5. Frames per second (FPS) study for Destiny 2 at different bitrates on a 5G network.

To study the frame drops experienced, during the Destiny 2 gameplay, due to the high frequency 5G mmWave band network, we plot the video stream frame rate generated from the cloud renderer (orange line) as well as the incoming frame rate received from the network by the client (gray line) in Fig. 5. The user is able to change the bitrate settings (e.g., 50Mbps in ⓐ, 40Mbps in ⓑ and 30Mbps in ⓒ) before beginning gameplay over a 5G network to the cloud server. Such settings are notified to the server to change the *encoding bitrate* accordingly. From Fig. 5, we can make the following observations:

- As shown in Fig. 5ⓐ, we can clearly observe a substantial difference in the frame rates before transmission on 5G network (orange line) and after reception at the client (gray line). Such a large gap is observed due to the increased frame drops potentially owing to the increased network latency as well as the increased packet loss due to the congestion in the 5G network buffers at higher bitrates [23, 28, 34, 72, 79] (50Mbps as shown in Fig. 5ⓐ). This stems from the fact that as the bitrate increases, the encoded video (game) frame size also increases, which adds to the time taken for sending and receiving all packets related to one frame. For example, when played at 50Mbps bitrate, the size of the encoded frame to be transmitted to the client is ≈100 kB whereas the encoded frame size for 40Mbps gameplay is ≈75 kB. Also, at higher bitrates, the amount of buffering required along intermediate network routers within the 5G networks increases leading to buffer overflow (packet loss) at the congested 5G links.
- To justify that by reducing the bitrate we can potentially reduce frame drops while transmitting the frames over network, we also profiled the frame rate when the bitrate is set to 40Mbps, and plotted the FPS timeline in Fig. 5ⓑ. Due to less data transmission, fewer packets are dropped, and thus, the frame drops decrease to only 1.8%. Similarly, on reducing the bitrate to 30Mbps, the frame drops further reduce to 1.3% (as shown in Fig. 5ⓒ). This indicates that playing 4K games at higher bitrates (which is a necessity for high-quality 4K cloud gaming applications [20]) using the 5G networks still suffers significant frame drops.
- Apart from high bitrates, the geographical distance from the server also affects network latency. To address this, various edge cloud servers or content delivery networks (CDN) are used for video streaming applications. Similarly, playing high resolution 4K games using the game streaming applications (cloud gaming services) can also benefit from an edge cloud.
- Another factor affecting the 5G network throughput and latency is the highly fluctuating nature of the 5G signals as discussed in Sec. 1. As mentioned earlier in Sec. 3.3, we conducted our profiling when the 5G signal strength was strong and the throughput was stable. On a general

note, we observed fluctuations in the 5G network throughput and variations in signal strength due to tree cover, cloudy skies, and buildings obstructing LoS communication with 5G base-stations. However, we didn't conduct our experiments in such conditions. We leave the scope of understanding the impact of 5G fluctuations on the application performance for our future studies.

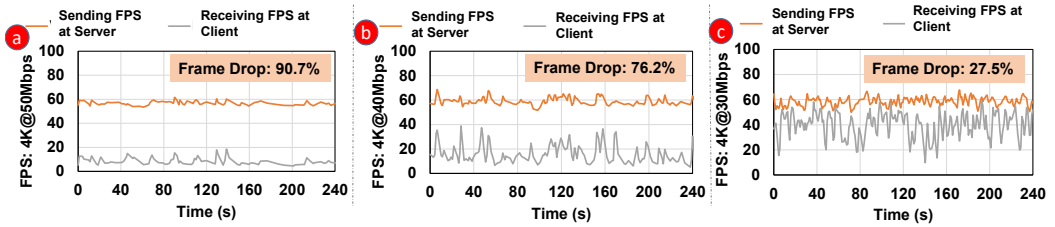

Fig. 6. Frames per second (FPS) study for Destiny 2 at different bitrates on WiFi network.

Our above study has revealed the extent to which the 5G network can support 4K game streaming with various bitrates. Next, we further investigate the effect of bitrate on frame rate when using WiFi, and plot the results in Fig. 6. The WiFi network shows characteristics similar to the 5G network with respect to different bitrates. More specifically, we make the following observations:

- As shown in Fig. 6ⓐ, when using WiFi to stream the 4K game frames at a 50Mbps bitrate, the frame drop rate can be as high as 90.7% (sender: 60 FPS; receiver: only 10 FPS). The frame drop rate is around 2× compared to when using 5G. This is due to the noticeably lower average bandwidth of WiFi compared to 5G.
- When the bitrate is reduced to 40Mbps, as shown in Fig. 6ⓑ, the frame drop rate decreases to 76.2%. Such an improvement of 14.5% on the frame throughput comes from the lower bitrate, but is still far from guaranteeing a good quality of experience for the user.
- By further decreasing the bitrate to 30Mbps, as shown in Fig. 6ⓒ, the frame drops now reduce to around 27.5%. This is still worse than the 40Mbps performance supported by 5G (see Fig. 5ⓑ), which indicates that the WiFi's capacity cannot even support 4K games at a 30Mbps bitrate.

The results in Fig. 5 and Fig. 6 indicate that the WiFi network suffers more frame drops compared to 5G, due to its insufficient bandwidth i.e., 250 Mbps in WiFi compared to more than 1 Gbps in 5G (measured using the Opensignal speedtest app [59]). Our tests demonstrate how the network technology used could potentially be a bottleneck for the QoS frame rate.

Meanwhile, the next question we want to ask is: *assuming an ideal/infinite network connection, can the user still observe a drop in the frame rate?* To answer this, we next investigate the execution latency of the two major components, namely, frame rendering and encoding, on the cloud side, shown in Fig. 4. Note that the rendering time reported includes the rendering requests' queuing time in the render queue along with the actual rendering time. Towards this, we plot both the encoding time and rendering time for seven games, with various graphics settings (low/minimum and high/maximum) as well as encoding qualities (low, medium and high) in Fig. 7. We present the games clustered as per their graphics fidelity with the high graphics-intensive games towards the right end and the low graphics-intensive games towards the left end in this and subsequent figures in the paper. From the Fig. 7, we make the following observations:

- The rendering time (denoted with cross marks) in several graphic-intensive games such as Destiny 2, Skyrim, AoE-III, exceeds the 16.66 ms deadline (denoted as the red line) , thus generating frames at a reduced rate (< 60 FPS), which in turn results in the client observing a reduced frame rate. This is because these games require detailed textures and/or 3D object rendering.

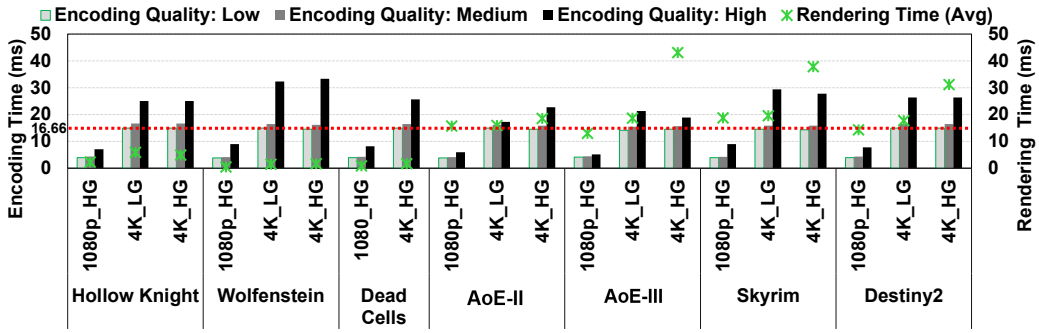

Fig. 7. Frame rendering and encoding time for seven gaming workloads. XX_YY where XX represents frame resolution [1080p, 4K] and YY represents graphics settings [HG, LG]. HG denotes high/maximum graphics settings and LG denotes low/minimum graphics settings allowed by the respective game engines.

Fig. 8. Discernible scene texture details for the same scene in Age of Empire-III at different graphics settings.

For example, as shown in Fig. 8, for the same scene/frame in Age of Empires-III 4K resolution, compared to low graphics settings, the high graphics version has more perceptible details such as the grass and ground textures. On the other hand, games such as Hollow Knight, Dead Cells, Wolfenstein, AoE-II, etc., in which less texture information and graphics effects for 2D objects is rendered, take less rendering time.

- Although AoE-II is a 2D rendered game, it takes more time for rendering owing to an inefficient game engine, which results in rendering requests spending significant time (≈10 ms) in the render queue [44]. Due to the same reason, AoE-III takes much more time for the rendering than similar graphics-intensive such as Skyrim and Destiny 2.
- In terms of encoding latency, at a high level, the encoding latency primarily correlates with the encoding quality levels across games, which translates to different compressed frame sizes. Across all 7 games, encoding at high quality takes significantly higher time than that at low quality. Especially in the case of 4K resolution video games, the encoding latency at high quality exceeds the 16.66 ms threshold, thus resulting in a reduced frame rate (< 60 FPS). Further, for the games with less detailed textures such as Hollow Knight, Dead Cells, Wolfenstein, etc., even encoding at low quality approaches the 16.66 ms deadline (which is significantly higher than their corresponding rendering time), thus suggesting that the encoding stage is one of the major QoS bottlenecks in the game streaming pipeline.
- Concentrating on each game, by comparing the last two sets of bars (4K_LG and 4K_HG), we observe similar encoding times regardless of graphic settings (i.e., high or low). This is because the encoding time is mostly determined by the frame resolution, rather than the texture details in one particular frame.

We note that while studying the frame rate for each of the components/stages in our game streaming pipeline (Fig. 4), we made sure the input frame rate to the corresponding component is at

least 60 FPS. For example, as mentioned earlier in Sec. 3.3, as there is no way to profile the encoding time taken by the proprietary NVIDIA GeForce Experience to encode each frame, we record the frames generated after the rendering stage and use the open-source FFMPEG [24] with the same hardware encoder (NVENC in the NVIDIA GPUs) used by the NVIDIA GeForce Experience to profile the encoding time. Similarly, to determine the network performance studied earlier, we ensure that the input frame rate to the network stage is 60 FPS, which is automatically handled by the smart motion interpolation feature supported by NVIDIA GeForce Experience to produce the (N+1.5)th frame between the (N+1)th frame and (N+2)th frame [55] when the rendered frame rate is less than 60 FPS. This ensured that the outgoing frame rate from the server to the network was 60 FPS, while measuring the received frame rate after the network transmission.

Based on the cloud gaming pipeline shown in Fig. 4, we next wish to study the two components on the client side – decoding and display – to investigate whether any bottleneck(s) exist from the edge-end. Note that the display time is determined by HSYNC (also known as line pulse or column pulse) and/or VSYNC (also known as frame pulse or row pulse), which are the horizontal or vertical sync signal. VSYNC is periodically generated by the display processor hardware every 11.1 ms with a 90 Hz refresh rate. Thus, we argue that the display time does not sit on the critical path (which is 16.66 ms for 60 FPS). On the other hand, the frames that are eventually shown on the user's display are mainly determined by the decoding latency. Therefore, we primarily focus on the decoder component decoding the compressed frames with various bitrates, and report the results in Fig. 9. To profile the decoding latency at the client, we measure the time taken between the reception of the input compressed frame at the decoder and the deposition of the decoded frame in the output buffer by the decoder. From Fig. 9, we make the following observations:

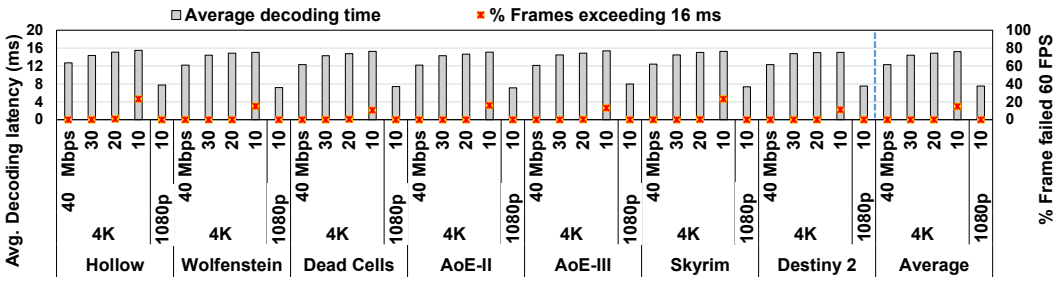

Fig. 9. Frame decoding time of seven gaming workloads at different bitrates and resolutions.

- Comparing the last two bars in each game, we find that, on average, with the same bitrate (10Mbps in this case), the decoder takes more time to decode a frame with a larger frame resolution (15.23 ms for 4K, and 7.49 ms for 1080p). This is because the larger the frame resolution is, the greater the number of pixel macro blocks the decoder needs to visit, thus the more execution time it requires.
- When comparing the first four bars within each game (4K resolution with various bitrates), we observe that the decoding time increases as the bitrate decreases. This is because the decoder has to perform more operations when it has less information (low bitrate scenarios) in the encoded frame to decode and generate the final frame than when it has more information (high bitrate scenarios).
- Across different games, by comparing the 4K resolution decoding time, we observe that the decoding times for corresponding bitrates are almost the same. This indicates that the decoding time primarily depends on the frame resolution and the encoding bitrate.

- From the percentage of frames that failed the 60 FPS requirement shown on the right y-axis, we observe that, on average, around 15% of the 4K frames with 10Mbps bitrate miss the deadline compared to no frame drops when playing 1080p resolution with 10Mbps bitrate. This exposes one potential bottleneck on the decoder stage, which may jeopardize the user experience on the client side.

4.1.2 Image Quality. As discussed in Sec. 2, both the frame rate (FPS) and the image quality are metrics that define the user's QoS, and we have discussed how the frame rate varies with different configurations of encoding, rendering, network connection, and decoding. Next, we focus on the image quality, and investigate the Peak signal-to-noise ratio (PSNR), which is one of the most important metrics for quantifying the image quality [4, 22], with different graphic settings, encoding qualities/speeds and resolutions across various games. Note that the acceptable PSNR for a cloud gaming is usually 40dB [10, 30] and a higher signal-to-noise ratio is better.

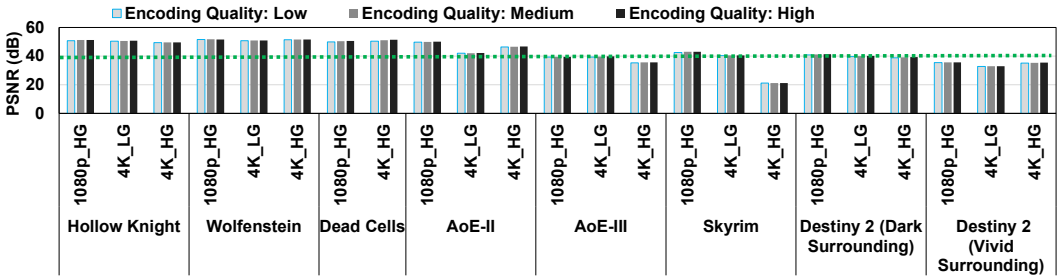

Fig. 10. PSNR of 7 gaming workloads for different encoding quality at 20 Mbps bitrate.

Fig. 10 plots the PSNR values we captured from seven games (the last two sets of bars are collected from the same game, i.e., Destiny 2, but with different surrounding scenes). We sweep the parameters from the rendering (i.e., graphics setting is low (minimum) or high (maximum)), encoding (i.e., the encoding quality is low, medium, or high) in different game's pipeline. From this figure, we can make the following observations:

- We observe that three out of seven games do not reach this quality requirement, i.e., AoE-III, Skyrim and Destiny 2 with 4K resolution, while most of the games meet the requirement with 1080p resolution. Such PSNR drop is brought by the encoding stage on the cloud side, which compresses/approximates the number of bits required for representing the current frame, which increases as the frame resolution increases. When the bitrate is set to be relatively low (i.e., 20Mbps in this case), the encoder has to compress the frame very aggressively for 4K frames compared to 1080p frames. Such a high compression ratio, especially in graphics-intensive games such as Destiny 2 with vivid surroundings, leads to an unacceptably low PSNR quality.
- By zooming into each game and comparing the PSNR with different encoding qualities (low, medium and high), we do not observe much PSNR difference. This is mainly because the encoding algorithms used in high-speed low latency specialized encoding cores on the NVIDIA GPUs (NVENC) have been highly optimized to support fast compression for most of the real-time applications. This indicates that the speed/quality of the encoding does not quite affect the image quality, and thus offers a potential optimization for improving the throughput and latency of the encoding stage in the entire pipeline, without losing much image quality.

To prove that the image quality is mainly affected by the bitrate, we further profiled the PSNR with respect to a high bitrate (50Mbps), for the above seven games, as shown in Fig. 11. By comparing this with 20Mbps case in Fig. 10, we observe a 9.36% PSNR improvement on average with this high bitrate. More specifically, with 50Mbps bitrate, now almost of the game scenarios successfully meet

the 40dB quality requirement, except for two cases (Skyrim and Destiny 2 with Vivid Surrounding) with negligible quality gap – less than 2.5%. Further, we can notice PSNR for Skyrim with 4K resolution and high graphics settings (4K_HG) with 20Mbps bitrate improves by 82% for 50Mbps bitrate. This is due to the encoding at 50 Mbps bitrate has less compression loss than 20Mbps, thus leading to an increase in PSNR (image quality). These observations indicate that the bitrate is the most critical parameter for improving the PSNR image quality.

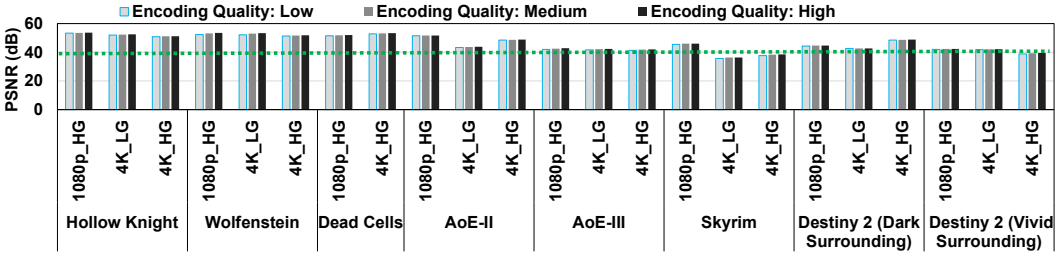

Fig. 11. PSNR of 7 gaming workloads for different encoding quality at 50 Mbps bitrate.

4.2 Power and Energy

As discussed in Sec. 2, apart from the QoS perspective, another critical concern is regarding the power/energy-efficiency on the client/edge side, due to the fact that most of these mobile devices are backed with a standalone battery. Targeting the power/energy-efficiency, in this section, we further study how the power and energy consumption behave along with different network types, bitrates, as well as game characteristics.

4.2.1 *Power/Energy With Respect To Network Type.*

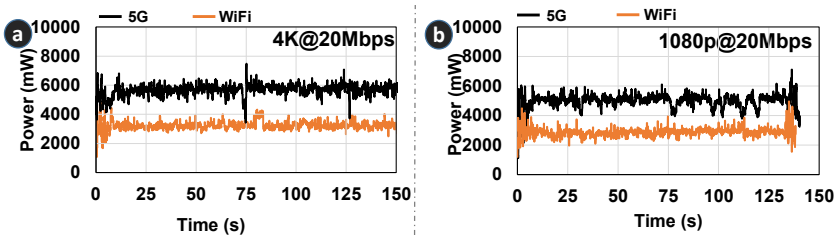

Fig. 12. Destiny 2: Power dissipation in 5G vs. WiFi.

To investigate how various network types affect the power consumption on the client devices, we plot the total power consumption along with time in Fig. 12ⓐ, when playing the Destiny 2 game with 4K resolution, and 20Mbps bitrate settings. The power traces are collected by probing the voltage and current values in the Android kernel as mentioned in Sec. 3. Compared to when using WiFi, the phone consumes 44% more power on average when using a 5G network (3.25 Watts for WiFi, 5.65 Watts for 5G). Similarly, comparing the power consumption when playing the Destiny 2 game at 1080p resolution (lower resolution) and 20Mbps bitrate (as shown in Fig. 12ⓑ) , the phone consumes about 5 Watts and 2.9 Watts on average for 5G and WiFi, respectively. This indicates that although 5G brings many advantages such as higher bandwidth and shorter round-trip latency, it consumes higher power compared to WiFi. These design trade-offs need to be carefully considered to make the optimal decisions for a specific optimization goal, e.g., performance-oriented, tail-latency-oriented, or power-oriented.

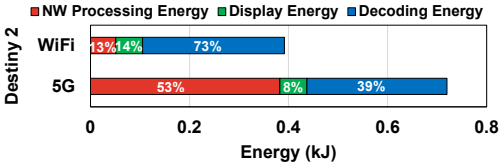

Fig. 13. Destiny 2: Energy consumption WiFi vs. 5G.

Recall from the cloud game streaming pipeline in Fig. 4, there are three major components on the client side – network processor, frame decoder, and display unit. We next breakdown the total energy consumption of game streaming on the client end into three parts: i) network processing energy including the consumptions from modem, RF, CPU interrupts for handling the communication, as well as the memory usage; ii) display energy including the consumptions from the display processor and screen; and iii) decoding energy including the consumptions from the frame decoder, CPU cycles and memory activities involved for the decoding. From Fig. 13 where Destiny 2 is played at 4K resolution (high graphics settings i.e., 4K_HG configuration mentioned earlier in Sec. 4.1.1) with 20Mbps bitrate, we make the following observations:

- The overall energy consumption to play the Destiny 2 game using the 5G network is around 46% more than using the WiFi. Detail reasons are explained later in this section.
- Specifically, the energy breakdown using the WiFi connection indicates that the decoding dominates the energy consumption, which accounts to around 73% of the total energy. While the network processing and the display consume only 13% and 14% of the total energy, respectively. This suggests that decoder is a good optimization target for improving the energy-efficiency.
- On the other hand, we observe from the 5G energy breakdown that the energy bottlenecks shift to both the network processing (i.e., 53%) and the decoding (i.e., 39%), while the display only consumes about 8% of total energy. Especially, the network processing energy when using 5G is significantly higher (about 7.5×) than that when using WiFi. This suggests the 5G network processing as the major optimization target in the client device for the 5G-enabled use cases.
- Comparing the absolute energy breakdown between 5G and WiFi, we observe that both the decoding and the display consume similar amounts of energy across different networks. The only source of the gap comes from the different power consumption by the 5G and WiFi networks, as also discussed above in Fig. 12(a).

To understand why 5G on the mobile device consumes more power, next, we take a look at the architecture of a typical 5G network subsystem in a phone, as shown in Fig. 14. At a high-level, the 5G network packet processing is done across the following components: antenna, RF Front End (RFFE), RF Transceiver, Modem, and CPU. The function of the each of the components can be

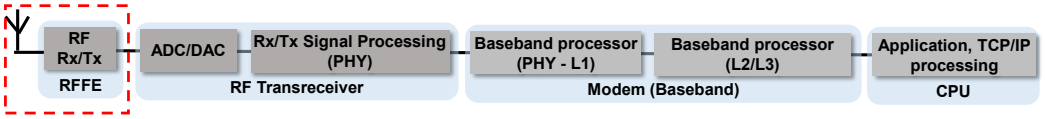

Fig. 14. Mobile platform 5G architecture.

explained using the lifetime of a received network packet. First, the antenna receives the signal and RFFE module does analog signal processing to downconvert the received signal frequency to an intermediate frequency. Next, at the RF Transceiver, the received analog signal is converted to digital domain, followed by baseband signal processing across PHY, L1 and L2 layers at the Modem and finally, the TCP/IP layer processing of the received packet is done by CPU. The Gbps range broadband data speed in 5G comes from the use of mmWave frequency bands which operate at around 28-52 GHz. Further, since the signals are transmitted at higher frequency (GHz), they have shorter wavelength and travel less distance and suffer from increased path loss compared to 4G that operates at MHz range. As a result, the 5G mmWave antenna on the phones needs to be powered on at a higher power than the 4G antenna. Additionally, in order to receive and transmit

packets to the base-station, a dedicated channel is assigned by the base-station to the phone which is relinquished after a certain period of inactivity which is usually around ≈ 10s of seconds. During the period of inactivity that is usually referred to as the "network tail", although there is no packet transmission/reception, the antenna and RFFE module still continue to remain powered at the high power state waiting for the arrival of new packets from the network, before going into the low power state after the timeout of the inactivity timer. The energy expended during these periods of the network tail can constitute a significant portion of the overall network energy consumption. This can be further explained using the overall power dissipation profile of a video streaming application as an example. Recall that in Sec. 2, we described the video streaming pipeline where the user receives the pre-encoded video frames. The most common approach for saving edge device network energy is batch downloading or buffering the future video frames into playback buffers. In Fig. 15, the video streaming application uses the playback buffer with a duration of minimum buffered frames as 15 seconds and a maximum buffered frames duration as 60 seconds. The application pauses downloading the new frames when the buffer reaches 60 seconds and resumes downloading when the buffer reaches 15 seconds. As can be seen in Fig. 15, after the end of each download period (when the buffer reaches 60 seconds), the device still continues to remain at the high power state before switching to the low power state after the end of the tail period. As discussed above, the high power state in the network tail period is mostly due to the 5G mmWave antenna that are still in high power state waiting for new packets. Thus, if the buffer duration configurations are not optimal, it could lead to more number of network tails, thus leading to high network tail energy. For the buffer setting in the Fig. 15, the network tail energy accounts for ≈35% of the total energy consumption during video playback. This leads to the observation that *the most power-hungry component in the 5G network subsystem is the antenna module* which, when in high power state even during periods of no packet transmission/reception, dissipates high power, thus making the 5G network subsystem much more power-hungry.

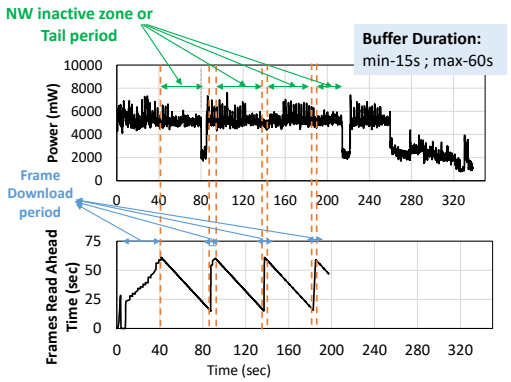

Fig. 15. Case Study: 5G power consumption for video streaming on a mobile device.

The reason why the 5G antenna module is very power-hungry is because of the massive arrays of antenna panels used on the mobile devices which employ techniques such as MIMO (multiple input multiple output) and beamforming [63], which are necessary for successfully capturing and retrieving the 5G mmWave signals that are extremely lossy in nature due to their shorter wavelength (high frequency) as discussed in Sec. 1. Powering up these antenna arrays requires a significant amount of energy [33]. However, not all the antennae in the panels are used all the time. Hence, innovations in designing an energy-efficient 5G mmWave antenna system [60] with smart and dynamically re-configurable antenna module could help in minimizing the energy consumption of 5G network modules on the mobile devices.

4.2.2 Power/Energy With Respect To Bitrates.

Not only the type of network connection (WiFi or 5G) can shape the mobile devices's energy behavior, but also the bitrate and resolution settings change the workload on the network subsystem as well as the decoder. To further study how these two parameters shape the energy behavior at the edge, we plot the energy consumption when playing the Destiny 2 game, connecting with 5G, with

various resolutions and bitrates in Fig. 16. From this figure, the following takeaways can be observed:

- With a fixed frame resolution such as 4K shown in the first three bars, with the bitrates increasing from 10Mbps to 20Mbps then to 40Mbps, the total energy consumption increases by 5% and 10%. By increasing the bitrate by twice, both the network and the decoder components on the client are affected: i) the amount of data transferred through the network is doubled which translates to more network energy consumption; and ii) meanwhile the decoding efforts are reduced to some extent due to the additional information (lower compression ratio) is available already for the decoding, which indicates that the decoder actually consumes less energy. Combining these two opposite trends, eventually the total energy still increases by 5%. This indicates that the energy saved by the decoding stage is overshadowed by the additional energy consumed by the network. A similar trend also occurs with other frame resolutions, which provides a generic insight of how the bitrate shapes the total energy with a positive correlation between them.

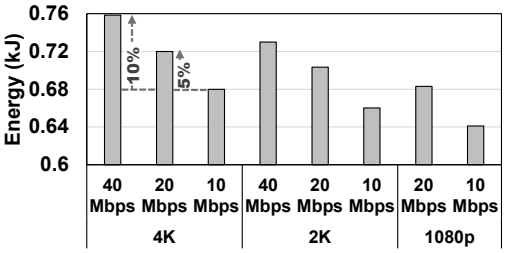

Fig. 16. Destiny 2: Energy comparison w.r.t. bitrates and resolutions on 5G.

- On the other hand, if we use the same bitrate (e.g., the third bar showing the 10Mbps in 4K and 2K (1440p) resolutions) and compare the energy consumptions with different resolutions, we find that the higher resolution frames consume more energy. Such additional energy consumption mainly comes from the decoding stage, which needs to fill/decode the pixel values for a larger output frame, compared to decoding for a lower resolution.

4.2.3 Power/Energy With Respect To Games.

We have discussed how the network settings (in Sec. 4.2.1), the bitrates as well as the resolutions (in Sec. 4.2.2) affect the energy consumption at the edge/client devices. Another dimension of investigation is the game-specific characteristics. Recall Table 1 where we summarized several salient features of the game workloads used in this paper, including scene details, scene variation, and camera motions. We study the average power consumption across the games when played at 4K resolution with 20Mbps bitrate using the 5G and WiFi network connection in Fig. 17.

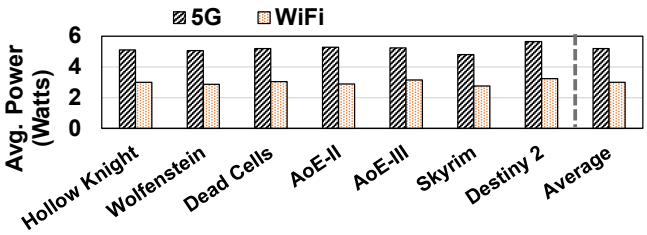

Fig. 17. Average power consumption across 7 games.

On average, across all 7 games, the overall average power consumption using 5G is about 1.7× higher than when using WiFi. Also, playing different games with different levels of graphic details consumes different power. Specifically, Destiny 2 drains more power than the other games mainly because as indicated in Table 1, Destiny 2 is the most graphic-intensive game.

To further understand the difference in energy consumption across the different games, (similar to the energy breakdown consumed by three components – network processing, decoding and display in Fig. 13) we also plot the breakdown for energy consumption of different games played at 4K resolution with 20Mbps bitrate using the 5G and WiFi network in Fig. 18 and make the following observations:

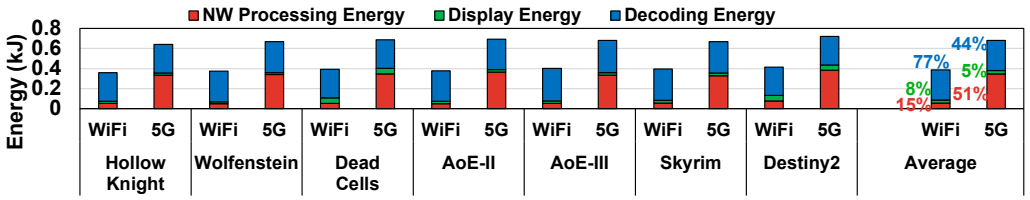

Fig. 18. Energy consumption across the different games when using 5G and WiFi.

- Overall, the 5G network processing energy dominates the total energy breakdown across all the games when using 5G, while the decoding energy dominates when using WiFi. The decoding energy across all the games is similar (accounting to ≈1.5 kJ of energy for about 10 minutes of gameplay).
- Comparing the network processing energy across all the games, for both the network types (5G and WiFi), the network processing energy (red bars) of the Destiny 2 game can be seen to be the highest among the other games. This is because Destiny 2 is a shooter game with very detailed scene textures and faster player movements. This results in high scene variations leading to substantial differences between consecutive frames, which in turn results in generation of larger number of motion vectors per frame. Consequently, receiving and processing the increased volume of data (motion vectors) at the client network interface consumes more energy.
- The display energy consumption varies significantly depending on the game being played. For example, for both the 5G and WiFi network types during 10 minutes of gameplay for each of the games, Destiny 2 and Dead Cells drain 0.28 kJ and 0.27 kJ energy, respectively for the display, while the Hollow Knight and Return to Castle Wolfenstein only consume 0.1 kJ, 0.09 kJ, respectively. This is because the gameplay environments of Destiny 2 and Dead Cells were much more brighter and vivid than Hollow Knight and Return to Castle Wolfenstein, which have more darker background (illuminating brighter pixels for the OLED mobile platform displays consume more energy than lighting the dark ones).

5 DISCUSSION

In this section, based on our thorough experiments in Sec. 4, we discuss how various configurable parameters, such as frame rendering quality, encoding bitrate, frame resolution, and network connection can be modulated to improve QoS as well as energy efficiency during gameplay. We also discuss briefly that the effect of the parameters on QoS and energy consumption can vary for different gaming workloads.

Client-Aware QoS and Energy Optimization: Based on our observations, we propose an edge device condition-aware scheme under which the client transmits the user preference, battery percentage, network conditions and frame drop (due to decoding deadline miss) statistics to the server during a gameplay as shown in Fig. 19. The server utilizes these information to dynamically modify the game visual settings, encoding bitrate and frame resolution allowing for a better user experience based on the the user preference – best gameplay experience (at the cost of faster battery drain), or prolonged gameplay experience (at the cost of reduced QoE). Transmission of these lightweight information can be done as frequently as

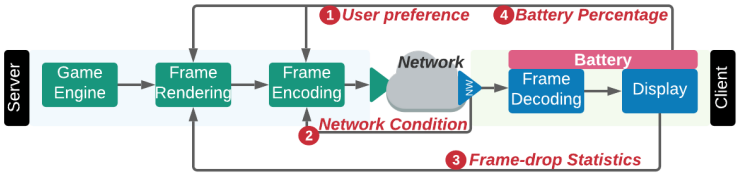

Fig. 19. QoS and energy-aware optimizations.

the rate of user input transmission from the client to the server. The polling of user input already occurs at a rate commensurate with the display/touchscreen refresh rate. Listed below are the strategies to be followed based on the collected statistics.

- For the best gameplay preference, when the network condition is poor, we first reduce the bitrate and observe the decoding time for these reduced bitrate frames. If the decoder misses its deadline, we reduce the frame resolution both at client and server. When the network condition improves, we increase the frame resolution and the bitrate and further monitor the decoding time for these frames. If they miss the decoding deadline, we increase the bitrate.
- For the prolonged gameplay preference, when the network condition is poor and/or the client device battery level is low, we reduce both the frame resolution and the bitrate. When the network condition improves and the client device battery level is high enough, only then we increase the frame resolution and the bitrate.
- If we observe frame generation deadline misses by the frame renderer at the server end, we reduce the in-game graphical settings.

When the frame resolution is reduced, there is a reduction in network packet processing time and decoding time at the client, leading to a reduction in battery depletion rate. On the server end, frame resolution reduction leads to an increase in frame rendering fps. On the other hand, increasing the bitrate yields high quality video game frames, but at the cost of increased battery depletion rate due to an increase in network packet processing energy at the client.

6 RELATED WORKS

Quality of Service for Cloud Gaming: Quality of Service (QoS) has been studied by different prior works to help improve the user's quality of experience (QoE) in cloud gaming applications [16, 25]. To understand how good is the QoS of current cloud gaming systems, Kuan-Ta Chen et al. proposed a suite of measurement techniques for QoS evaluations, across adaptable frame rate, graphic quality, server processing delays, and network bandwidth [16]. More recently, Lindstrom et al. studied the impact of the QoS factors most affecting the players' QoE and in-game performance by manipulating the players' network conditions and collecting low-level QoS metrics [25]. The methodology utilized in these prior works mainly focused on statistic correlation analysis on the QoS metrics, leaving the end-to-end cloud gaming pipeline as a black-box. Instead, in this work, we breakdown the end-to-end pipeline into cloud, network and client components, and sweep all the configurable parameters that affect the QoS as well as the energy efficiency to provide a systematical analysis on the performance and energy-efficiency of the cloud gaming.

5G Network: With its promised multi-Gbps speed, sub-10ms low latency and massive connectivity, 5G networks are being rapidly deployed. 5G networks have the potential of improving the performance and QoE for various applications. Motivated by this, there exist prior studies that examined different 5G carriers, deployment schemes, radio bands, protocols, and mobility patterns [54, 62, 72]. Dongzhu Xu et al. demystified operational 5G network by focusing on four major aspects: physical layer signal quality and performance, end-to-end throughput and latency, QoE of 5G's applications, and energy consumption on mobile phones [72]. Their experimental results show that the 5G link approaches Gbps throughput, with the cost of 2-3× power consumption over 4G. On the other hand, Narayanan et al. investigated the performance, power, and QoE implications of 5G, and revealed key characteristics of commercial 5G in terms of throughput, latency, handover behaviors, radio state transitions, and power consumption, compared to 4G/LTE networks [54]. Furthermore, these revealed characteristics provided more insights into how to better utilize 5G by balancing the critical trade-off between performance and energy consumption. However, there have been very few works that study how 5G benefits the communication stress between the client and the

cloud in the context of cloud gaming stream applications [62]. In this paper, we investigated the trade-offs between WiFi and 5G by investigating 5G's pros and cons in terms of power/energy consumption, and the potential improvements on frame throughput.

Video Streaming Optimizations: The past years have also seen many research works optimizing the video streaming applications' performance and energy consumption. For example, adaptive bitrate streaming (ABR) is a technique for dynamically selecting the compression level and video quality of a video stream, based on the current network bandwidth condition [18, 32, 36, 38, 45, 68, 74]. The receiver buffer resizing is another dimensional technique for adjusting to the difference between the transmission rate and the playback rate [6, 29, 43, 66]. On the client/edge side, Zhang et al. proposed and experimentally evaluated three complementary techniques, race-to-sleep, content caching, and display caching [77], with the collective goal of minimizing the energy consumption of the video processing flows on mobile devices. Recent advances in neural super-resolution have also been utilized to enhance video quality by leveraging client-side computation [73]. To make such computation more light-weight and less power-hungry, NEMO [73] proposed a system that selectively applies neural super-resolution to only a small number of pixel points. The evaluation results indicate around 32.1% user QoE improvement. We would like to emphasize that, although the above mentioned techniques work well for video streaming applications where the videos to be streamed are pre-encoded offline and stored at the server, most of them cannot be easily deployed in the applications such as cloud gaming, video calling/conferencing, etc. due to their real-time and interactive nature. Furthermore, it is important to note that in contrast to the video calling/conferencing applications, the additional latency incurred to transmit the user inputs from the client to the server (for processing the future frames at the server) at the runtime in the cloud gaming applications makes the deployment of the above adaptive playback techniques further challenging.

7 CONCLUSION

This paper presents a comprehensive performance and energy consumption analysis of game streaming applications across a typical cloud gaming pipeline, consisting of the server end, network and client/mobile platform. Our analysis reveals that i) On the cloud/server end, the rendering stage and the encoding stage consume significant amount of time for the graphics-laden games and miss the target 16.66 ms frame deadline for several applications, thus suggesting to minimize the latency of these two stages through novel hardware and software optimizations. ii) WiFi and 4G networks are capable of supporting 1080p game streaming to a large extent without significant frame drop. However, the QoS/frame drop increases significantly for 4K videos. High throughput/bandwidth offered by 5G network can support 4K video game streaming. However, playing high-quality 4K video games at higher bitrates suffers significant frame drops potentially owing to the increased network latency and congestion in network buffers in the 5G networks. The lossy nature of the 5G signals (due to their high operating frequency) can also exacerbate the situation. Thus, a coordinated design to support variable bitrates between a server and an edge device for real-time and interactive use cases such as cloud gaming applications as well as optimizations in the 5G networks' backend to support high-bitrate game streaming is essential to utilize 5G more effectively. iii) On the mobile side, across all the games, the 5G network module is the most energy consuming component (about 51% on average, which is about 36% higher compared to WiFi) followed by the decoding stage, thus stressing on the need for more energy-efficient 5G modules and decoder hardware. This analysis is essential to understand the interplay of different stages of the gaming pipeline and identify bottleneck stages for improving the QoS and energy efficiency of the emerging GaaS paradigm.

8 ACKNOWLEDGEMENTS

We thank our shepherd, Martin Arlitt, for his insightful comments and suggestions towards improving the paper content and presentation. We also thank the anonymous reviewers for their helpful feedback. This research is supported in part by NSF grants #1629915, #2116962, #1763681, #1912495, and #1526750.

REFERENCES

[1] NVIDIA Corporation. 2021. GeForce Now. "https://www.nvidia.com/en-us/geforce-now/".
[2] Amazon.com, Inc. 2020. Amazon Luna. "https://www.amazon.com/luna/landing-page".
[3] Andrew Burnes. 2020. FrameView Performance and Power Benchmarking App: Free Download Available Now. "https://www.nvidia.com/en-us/geforce/news/nvidia-frameview-power-and-performance-benchmarking-app-download/".
[4] Amit Ahlawat Anju. 2016. Performance Analysis of Image Compression Technique. *Image* (2016), 107–111.
[5] Anritsu. 2020. Measuring Path Loss of 5G FR2 Transmissions Through Common Materials Found in the Signal Path. "https://dl.cdn-anritsu.com/en-us/test-measurement/files/Application-Notes/Application-Note/11410-01189A.pdf".
[6] Niranjan Balasubramanian, Aruna Balasubramanian, and Arun Venkataramani. 2009. Energy Consumption in Mobile Phones: A Measurement Study and Implications for Network Applications. In *Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement*. Association for Computing Machinery, New York, NY, USA, 280–293.
[7] Bethesda Softworks LLC. 2021. Skyrim. "https://elderscrolls.bethesda.net/en/skyrim".
[8] Bethesda Softworks LLC. 2022. Return to Castle Wolfenstein. "https://store.steampowered.com/app/9010/Return_to_Castle_Wolfenstein/".
[9] Bungie, Inc. 2021. Destiny 2. "https://www.bungie.net/en/pub/aboutdestiny".
[10] Wei Cai, Zhen Hong, Xiaofei Wang, Henry C. B. Chan, and Victor C. M. Leung. 2015. Quality-of-Experience Optimization for a Cloud Gaming System With Ad Hoc Cloudlet Assistance. *IEEE Transactions on Circuits and Systems for Video Technology* (2015), 2092–2104.
[11] Wei Cai, Ryan Shea, Chun-Ying Huang, Kuan-Ta Chen, Jiangchuan Liu, Victor C. M. Leung, and Cheng-Hsin Hsu. 2016. A Survey on Cloud Gaming: Future of Computer Games. *IEEE Access* (2016), 7605–7620.
[12] Cameron Gutman, Diego Waxemberg, Aaron Neyer, Andrew Hennessy . 2013. Moonlight Android. "https://github.com/moonlight-stream/moonlight-android".
[13] Cameron Gutman, Diego Waxemberg, Aaron Neyer, Michelle Bergeron, Andrew Hennessy, Aidan Campbell. 2013. Moonlight. "https://moonlight-stream.org/".
[14] Cameron Gutman, Diego Waxemberg, Aaron Neyer, Michelle Bergeron, Andrew Hennessy, Aidan Campbell. 2021. Moonlight Internet Hosting Tool. "https://github.com/moonlight-stream/Internet-Hosting-Tool".
[15] Aaron Carroll and Gernot Heiser. 2010. An Analysis of Power Consumption in a Smartphone. In *Proceedings of the 2010 USENIX Conference on USENIX Annual Technical Conference*. USENIX Association, USA, 21.
[16] Kuan-Ta Chen, Yu-Chun Chang, Hwai-Jung Hsu, De-Yu Chen, Chun-Ying Huang, and Cheng-Hsin Hsu. 2014. On the Quality of Service of Cloud Gaming Systems. *IEEE Transactions on Multimedia* (2014), 480–495.
[17] Nachiappan Chidambaram Nachiappan, Praveen Yedlapalli, Niranjan Soundararajan, Mahmut Taylan Kandemir, Anand Sivasubramaniam, and Chita R. Das. 2014. GemDroid: A Framework to Evaluate Mobile Platforms. In *The 2014 ACM International Conference on Measurement and Modeling of Computer Systems*. Association for Computing Machinery, New York, NY, USA, 355–366.
[18] Philip A Chou and Zhourong Miao. 2006. Rate-distortion Optimized Streaming of Packetized Media. *IEEE Transactions on Multimedia* (2006), 390–404.
[19] Dean Takahashi. 2021. Newzoo: Cloud gaming will reach 23.7M paying users and generate $1.6B in 2021. "https://venturebeat.com/2021/08/26/newzoo-cloud-gaming-will-reach-23-7m-paying-users-and-generate-1-6b-in-2021/".
[20] Dror Gill. 2019. How To Cut Cloud Gaming Bitrates In Half So That Twice As Many Users Can Play. "https://blog.beamr.com/2019/11/19/how-to-cut-cloud-gaming-bitrates-in-half-so-that-twice-as-many-users-can-play/".
[21] Epic Games, Inc. 2021. Fortnite. "https://www.epicgames.com/fortnite/en-US/home".
[22] Fernando A. Fardo, Victor H. Conforto, Francisco C. de Oliveira, and Paulo S. Rodrigues. 2016. A Formal Evaluation of PSNR as Quality Measuremen Parameter for Image Segmentation Algorithms. *CoRR* (2016), arXiv:1605.07116.
[23] Nick Feamster and Hari Balakrishnan. 2002. Packet Loss Recovery For Streaming Video. In *12th International Packet Video Workshop*. PA: Pittsburgh, 9–16.
[24] FFmpeg team. 2000. FFmpeg. "https://www.ffmpeg.org/".
[25] Sebastian Flinck Lindström, Markus Wetterberg, and Niklas Carlsson. 2020. Cloud Gaming: A QoE Study of Fast-paced Single-player and Multiplayer Gaming. In *2020 IEEE/ACM 13th International Conference on Utility and Cloud Computing (UCC)*. 34–45.

[26] Fredrik Fornwall. 2021. Termux. "https://termux.com/".
[27] Google. 2019. Stadia - One place for all the ways we play. "https://stadia.google.com/".
[28] Habtegebreil Haile, Karl-Johan Grinnemo, Simone Ferlin, Per Hurtig, and Anna Brunstrom. 2021. End-to-end Congestion Control Approaches for High Throughput and Low Delay in 4G/5G Cellular Networks. *Computer Networks* (2021), 107692.
[29] Wenjie Hu and Guohong Cao. 2015. Energy-aware video streaming on smartphones. In *2015 IEEE Conference on Computer Communications (INFOCOM)*. Institute of Electrical and Electronics Engineers Inc., United States, 1185–1193.
[30] Chun-Ying Huang, Kuan-Ta Chen, De-Yu Chen, Hwai-Jung Hsu, and Cheng-Hsin Hsu. 2014. GamingAnywhere: The First Open Source Cloud Gaming System. *ACM Trans. Multimedia Comput. Commun. Appl.* (2014).
[31] Chun-Ying Huang, Cheng-Hsin Hsu, Yu-Chun Chang, and Kuan-Ta Chen. 2013. GamingAnywhere: An Open Cloud Gaming System. In *Proceedings of the 4th ACM Multimedia Systems Conference*. Association for Computing Machinery, New York, NY, USA, 36–47.
[32] Te-Yuan Huang, Ramesh Johari, Nick McKeown, Matthew Trunnell, and Mark Watson. 2014. A Buffer-Based Approach to Rate Adaptation: Evidence from a Large Video Streaming Service. In *Proceedings of the 2014 ACM Conference on SIGCOMM*. Association for Computing Machinery, New York, NY, USA, 187–198.
[33] Yiming Huo, Xiaodai Dong, and Wei Xu. 2017. 5G Cellular User Equipment: From Theory to Practical Hardware Design. *IEEE Access* (2017), 13992–14010.
[34] Mikel Irazabal Bengoa. 2021. Enhanced Quality of Service Mechanisms for 5G Networks. (2021).
[35] Bart Jansen, Timothy Goodwin, Varun Gupta, Fernando Kuipers, and Gil Zussman. 2018. Performance Evaluation of WebRTC-Based Video Conferencing. *SIGMETRICS Perform. Eval. Rev.* (2018), 56–68.
[36] Junchen Jiang, Vyas Sekar, and Hui Zhang. 2012. Improving Fairness, Efficiency, and Stability in HTTP-Based Adaptive Video Streaming with FESTIVE. In *Proceedings of the 8th International Conference on Emerging Networking Experiments and Technologies*. Association for Computing Machinery, New York, NY, USA, 97–108.
[37] Julz. 2021. What's the Best Frame Rate for Gaming ? "https://www.build-gaming-computers.com/best-frame-rate-for-pc-gaming.html".
[38] Mark Kalman, Eckehard Steinbach, and Bernd Girod. 2002. Rate-distortion Optimized Video Streaming with Adaptive Playout. In *Proceedings. International Conference on Image Processing*. IEEE, III–III.
[39] Kuba Kaszyk, Harry Wagstaff, Tom Spink, Björn Franke, Mike O'Boyle, Bruno Bodin, and Henrik Uhrenholt. 2019. Full-System Simulation of Mobile CPU/GPU Platforms. In *2019 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*. IEEE Computer Society, Los Alamitos, CA, USA, 68–78.
[40] Kozakoff,Dennis J. and Corallo, Carlo A. and Petra, D. and Roovers, Wilhelmus Cornelus Wal . 2016. 5G Cellular Electromagnetic Window Considerations. "https://www.dsm.com/content/dam/dsm/dyneema/en_GB/Downloads/Researchpapers/5G_Cellular_article.pdf".
[41] KRAFTON, Inc. 2021. PUBG Mobile. "https://www.pubgmobile.com/en-US/home.shtml".
[42] KRAFTON, Inc. 2022. PUBG: Battlegrounds. "https://store.steampowered.com/app/578080/PUBG_BATTLEGROUNDS/".
[43] Xin Li, Mian Dong, Zhan Ma, and Felix C.A. Fernandes. 2012. GreenTube: Power Optimization for Mobile Videostreaming via Dynamic Cache Management. In *Proceedings of the 20th ACM International Conference on Multimedia*. Association for Computing Machinery, New York, NY, USA, 279–288.
[44] Licidy. 2019. Known Issue: Performance Issues with Low FPS. "https://forums.ageofempires.com/t/known-issue-performance-issues-with-low-fps/60111".
[45] Hongzi Mao, Ravi Netravali, and Mohammad Alizadeh. 2017. Neural Adaptive Video Streaming with Pensieve. In *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*. Association for Computing Machinery, New York, NY, USA, 197–210.
[46] Marshall Honorof. 2021. Can't find a PS5, Xbox Series X or GPU? Embrace it. "https://www.tomsguide.com/news/ps5-xbox-series-x-gpu-semiconductor-shortage".
[47] Microsoft. 2021. Age of Empires-II: Definitive Edition. "https://www.ageofempires.com/games/aoeiide/".
[48] Microsoft. 2021. Age of Empires-III: Definitive Edition. "https://www.ageofempires.com/games/aoeiiide".
[49] Microsoft. 2021. Xbox Cloud Gaming (Beta) with Xbox Game Pass. "https://www.xbox.com/en-US/xbox-game-pass/cloud-gaming".
[50] Mojang Studios. 2011. Minecraft. "https://www.minecraft.net/en-us".
[51] Mordor Intelligence. 2021. Cloud Gaming Market - Growth, Trends, COVID-19 Impact, and Forecasts (2021 - 2026). "https://www.mordorintelligence.com/industry-reports/cloud-gaming-market".
[52] Motion Twin. 2021. Dead CellS. "https://dead-cells.com/".
[53] Nachiket Mhatre. 2021. Here's What Caused The Ongoing Global Chip Shortage & Why It Will Only Get Worse . "https://onsitego.com/blog/global-chip-shortage-explained-causes-future/".

[54] Arvind Narayanan, Xumiao Zhang, Ruiyang Zhu, Ahmad Hassan, Shuowei Jin, Xiao Zhu, Xiaoxuan Zhang, Denis Rybkin, Zhengxuan Yang, Zhuoqing Morley Mao, Feng Qian, and Zhi-Li Zhang. 2021. A Variegated Look at 5G in the Wild: Performance, Power, and QoE Implications. In *Proceedings of the 2021 ACM SIGCOMM 2021 Conference (SIGCOMM '21)*. Association for Computing Machinery, New York, NY, USA, 610–625.
[55] NVIDIA Corporation. 2019. An Introduction to the NVIDIA Optical Flow SDK. "https://developer.nvidia.com/blog/an-introduction-to-the-nvidia-optical-flow-sdk/".
[56] NVIDIA Corporation. 2021. GeForce Experience. "https://www.nvidia.com/en-us/geforce/geforce-experience/".
[57] NVIDIA Corporation. 2021. NVIDIA GameStream. "https://www.nvidia.com/en-us/shield/support/shield-tv/gamestream/".
[58] NVIDIA Corporation. 2021. ShadowPlay: Record, Share Game Videos & Screenshots. "https://www.nvidia.com/en-us/geforce/geforce-experience/shadowplay/".
[59] Opensignal. 2021. Opensignal - 5G, 4G, 3G Internet & WiFi Speed Test. "https://play.google.com/store/apps/details?id=com.staircase3.opensignal&hl=en_US&gl=US".
[60] Junho Park, Heechang Seong, Yong Nam Whang, and Wonbin Hong. 2019. Energy-efficient 5G Phased Arrays Incorporating Vertically Polarized Endfire Planar Folded Slot Antenna for mmWave Mobile Terminals. *IEEE Transactions on Antennas and Propagation* (2019), 230–241.
[61] Pelham Smithers, Omri Wallach, Clayton Wodsworth. 2020. The Rise Of Gaming Revenue Visualized. "https://www.visualcapitalist.com/wp-content/uploads/2020/11/history-of-gaming-by-revenue-share-full-size.html".
[62] Oswaldo Sebastian Peñaherrera-Pulla, Carlos Baena, Sergio Fortes, Eduardo Baena, and Raquel Barco. 2021. Measuring Key Quality Indicators in Cloud Gaming: Framework and Assessment Over Wireless Networks. *Sensors* (2021).
[63] Qualcomm Technologies Inc. 2018. First 5G mmWave Antenna Module for Smartphones. "https://www.microwavejournal.com/articles/31448-first-5g-mmwave-antenna-module-for-smartphones".
[64] Eman Ramadan, Arvind Narayanan, Udhaya Kumar Dayalan, Rostand AK Fezeu, Feng Qian, and Zhi-Li Zhang. 2021. Case for 5G-aware video streaming applications. In *Proceedings of the 1st Workshop on 5G Measurements, Modeling, and Use Cases*. Association for Computing Machinery, 27–34.
[65] Sam Desatoff. 2021. Report: Cloud gaming market to reach $6.5 billion by 2024 (Newzoo). "https://gamedaily.biz/article/2143/report-cloud-gaming-market-to-reach-65-billion-by-2024-newzoo".
[66] Aaron Schulman, Vishnu Navda, Ran Ramjee, Neil Spring, Pralhad Deshp, Calvin Grunewald, Kamal Jain, and Venkata N. Padmanabhan. 2010. Bartendr: A Practical Approach to Energy-aware Cellular Data Scheduling. In *Proceedings of the Sixteenth Annual International Conference on Mobile Computing and Networking*. Association for Computing Machinery, New York, NY, USA, 85–96.
[67] Sony Interactive Entertainment LLC. 2021. PlayStation Now. "https://www.playstation.com/en-us/ps-now/".
[68] Kevin Spiteri, Rahul Urgaonkar, and Ramesh K. Sitaraman. 2016. BOLA: Near-optimal bitrate adaptation for online videos. In *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*. IEEE Press, 1–9.
[69] Team Cherry. 2021. Hollow Knight. "https://www.hollowknight.com/".
[70] Ninad Warty, Ramanujan K. Sheshadri, Wei Zheng, and Dimitrios Koutsonikolas. 2012. A First Look at 802.11n Power Consumption in Smartphones. In *Proceedings of the First ACM International Workshop on Practical Issues and Applications in next Generation Wireless Networks*. Association for Computing Machinery, New York, NY, USA, 27–32.
[71] Wikipedia Contributor. 2021. Games as a service. "https://en.wikipedia.org/wiki/Games_as_a_service".
[72] Dongzhu Xu, Anfu Zhou, Xinyu Zhang, Guixian Wang, Xi Liu, Congkai An, Yiming Shi, Liang Liu, and Huadong Ma. 2020. Understanding Operational 5G: A First Measurement Study on Its Coverage, Performance and Energy Consumption. In *Proceedings of the Annual Conference of the ACM Special Interest Group on Data Communication on the Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM '20)*. Association for Computing Machinery, New York, NY, USA, 479–494.
[73] Hyunho Yeo, Chan Ju Chong, Youngmok Jung, Juncheol Ye, and Dongsu Han. 2020. NEMO: Enabling Neural-Enhanced Video Streaming on Commodity Mobile Devices. In *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*. Association for Computing Machinery, New York, NY, USA, 14 pages.
[74] Xiaoqi Yin, Abhishek Jindal, Vyas Sekar, and Bruno Sinopoli. 2015. A Control-Theoretic Approach for Dynamic Adaptive Video Streaming over HTTP. In *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*. Association for Computing Machinery, New York, NY, USA, 325–338.
[75] Chanmin Yoon, Dongwon Kim, Wonwoo Jung, Chulkoo Kang, and Hojung Cha. 2012. AppScope: Application Energy Metering Framework for Android Smartphone Using Kernel Activity Monitoring. In *2012 USENIX Annual Technical Conference (USENIX ATC 12)*. USENIX Association, Boston, MA, 387–400.
[76] ZeroTier, Inc. 2021. ZeroTier. "https://www.zerotier.com/".
[77] Haibo Zhang, Prasanna Venkatesh Rengasamy, Shulin Zhao, Nachiappan Chidambaram Nachiappan, Anand Sivasubramaniam, Mahmut T. Kandemir, Ravi Iyer, and Chita R. Das. 2017. Race-to-Sleep + Content Caching + Display

Caching: A Recipe for Energy-Efficient Video Streaming on Handhelds. In *Proceedings of the 50th Annual IEEE/ACM International Symposium on Microarchitecture*. Association for Computing Machinery, New York, NY, USA, 517–531.

[78] Lide Zhang, Birjodh Tiwana, Zhiyun Qian, Zhaoguang Wang, Robert P. Dick, Zhuoqing Morley Mao, and Lei Yang. 2010. Accurate Online Power Estimation and Automatic Battery Behavior Based Power Model Generation for Smartphones. In *Proceedings of the Eighth IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis*. Association for Computing Machinery, New York, NY, USA, 105–114.

[79] Menglei Zhang, Michele Polese, Marco Mezzavilla, Jing Zhu, Sundeep Rangan, Shivendra Panwar, and Michele Zorzi. 2019. Will TCP Work in mmWave 5G Cellular Networks? *IEEE Communications Magazine* (2019), 65–71.

Received October 2021; revised December 2021; accepted January 2022