A Large-Scale Network Measurement Study of NVIDIA GeForce NOW Cloud Gaming in the Wild

Minzhao Lyu and Vijay Sivaraman

***Abstract*—Cloud gaming, whereby game graphics is rendered in cloud servers and streamed back to the user, expands the gaming market to billions of users who do not have gaming consoles or high-power graphics PCs. However, cloud gaming puts high pressure on the access network operator to provide stable and high-bandwidth network connectivity (often an order of magnitude more than console gaming) so users can have a good gaming experience. This paper aims to help network operators gain visibility into cloud gaming behaviors and experience, so they can better plan for it and manage it. We focus specifically on NVIDIA's GeForce NOW cloud gaming platform, and make four contributions. First, we comprehensively analyze the network anatomy of the cloud gaming session, including establishment of flows and their volumetric patterns, and illustrate differences across user setups. Second, we develop a method to detect cloud gaming activity and user setup in real-time, as well as quantify user experience in terms of frame rate, resolution, and latency. In our third contribution we deploy our algorithm in a campus network and reveal insights on user setups (PC versus mobile, browser versus app) and experience (frames-per-second and resolution) of gameplay sessions detected over four months. For our final contribution we measure cloud gaming at scale in the wild by deploying our solution in a commercial operator of NVIDIA's GeForce NOW platform, and show how experience relates to user setup, game title, access network type and subnet.**

***Index Terms*—Cloud gaming, Network measurement, Quality-of-service, Quality-of-experience.**

I. Introduction

Cloud gaming, also known as Game-as-a-Service (GaaS), allows users to play graphics-intensive games without having to own expensive hardware such as gaming consoles (*e.g.,* PlayStation and X-Box) or PCs equipped with high-end graphics cards. Instead, game graphics is rendered on powerful cloud-hosted servers, and the resulting video is streamed instantaneously to the end-user's device. Cloud gaming is estimated to grow 58.75% each year to become a $22.53 billion industry by 2030 [3]–[5]. Entities such as NVIDIA, Amazon, Sony, and Microsoft, as well as a slew of smaller companies, have commercial cloud gaming offerings in the market already.

A significant barrier to the growth of cloud gaming is the high bandwidth and stability that it demands from the network. Whereas a typical game (*e.g.,* shooting game like Call-of-Duty or sports game like FIFA) played on a console or PC requires only a few hundred kilobits-per-second (kbps) from the network [6], a cloud game demands two orders of magnitude more, namely tens of Megabits-per-second (Mbps). If such higher bandwidth (coupled with low latency) is not consistently available, gaming experience becomes frustrating due to drop in frames-per-second (fps), poor resolution, high lag and jerky movements. Internet Service Providers (ISPs) that connect cloud gaming players and cloud infrastructures (*e.g.,* GPU clusters) often bear the brunt of this frustration, leading to complaints, churn, and brand damage.

Network operators have historically provisioned their networks for peak loads, agnostic to the application mix. This blunt approach is not cost-effective, because elastic applications (like large downloads) can grab unconstrained amounts of network bandwidth at any instant, degrading performance for sensitive applications like cloud gaming. ISPs are therefore looking for ways to distinguish (and potentially segregate) the latter – as one example, Comcast recently announced [7] it will support low latency forwarding in its cable broadband network for application streams marked by specific content providers (such as Apple and NVIDIA). Network operators consequently need mechanisms to identify bandwidth and latency sensitive content streams such as cloud gaming, continuously monitor user experience on these streams, and if needed invoke network APIs for dynamic Quality-on-Demand (QoD) [8].

To the best of our knowledge, prior works that develop network traffic analysis methods for network operators lack the capability to measure the prevalence of, and experience on, cloud gaming over their network infrastructure with fine visibility into user setups and functionalities of traffic flows, so that they can configure their networks for guaranteed user experience. For example, existing works have analyzed network traffic for web browsing [9], video streaming [10]–[14], virtual reality [15], *etc.*, but not cloud gaming. Existing studies of cloud gaming have considered video processing delays on client devices/cloud servers [16], energy consumption on mobile devices [17], characterizing game streaming RTP flows [18], assessing the impact of wireless/edge network conditions on game streaming flows [19]–[21], which have not developed a traffic analysis method for ISPs who seek insights to better optimize their networks for the emerging application.

In this paper, building upon our previous work [2], we describe our real-time network traffic analysis methods that can be deployed by network operators to gain fine-grained visibility into cloud gaming behaviors and experience, so they can be actively involved in managing service delivery quality.

This is the preprint version of our paper [1] accepted at IEEE/ACM Transactions on Networking. (*Corresponding author: Minzhao Lyu.*)

M. Lyu and V. Sivaraman are with the School of Electrical Engineering and Telecommunications, University of New South Wales, Sydney, NSW 2052, Australia (e-mails: minzhao.lyu@unsw.edu.au, vijay@unsw.edu.au).

Funding for this project is provided by the Australian Government's Cooperative Research Centres Projects (CRC-P) Grant CRCPXIV000099.

A preliminary version [2] of this article was presented at the Passive and Active Measurement (PAM) 2024 conference.

We have chosen to focus on NVIDIA's cloud gaming platform, GeForce NOW (GFN), in this paper for a few reasons: (a) it is currently recognized as the leader in the global cloud gaming market [22]; (b) it supports a rich collection of multiplayer games from all major game publishers like Steam, Ubisoft, and Epic Games; (c) it is accessible broadly, both via the browser and a bespoke app, on desktops (*i.e.,* macOS and Windows) and mobile devices (*i.e.,* android and iOS), thereby providing a rich set of conditions under which it can be studied; and (d) we have partnered with the network operator that exclusively hosts the GFN cloud gaming servers for the Australia and New Zealand region, allowing our method to be validated and deployed for the leading cloud gaming service provider at scale.

Our *first contribution* (§III) characterizes the network anatomy of cloud gaming services. We identify the establishment of critical service flows, and benchmark their volumetric patterns, associated with gameplay management, user input, and audio/video streaming, while highlighting the differences between the user playing on a browser versus the app. We also identify attributes in the traffic that aid in user experience estimation, such as sequence numbers for measuring game lag and marker packets or stochastic patterns in packet payload sizes for detecting frame boundaries.

Our *second contribution* (§IV) develops a practical network traffic analysis method to detect cloud gaming sessions and measure gameplay experience in real-time. We use a stateful mapping mechanism that tracks service domains accessed by active flows so as to detect the start of a cloud gaming session, as well as identify the user setup (*i.e.,* operating system and software agent type). We then categorize the gameplay flows as gaming management, user input, and video streaming, based on volumetric attributes. Network operators can therefore prioritize certain gameplay flows for guaranteed user experience such as by mapping user inputs into low-latency network slices (*e.g.,* URLLC) while assigning gaming video to high-bandwidth slices. Finally, we derive user experience measures, including game lag (latency between the gamer device and cloud server), video frame rate, and video graphic resolution.

For our *third contribution* (§V), we implement our measurement system in a University campus network[1] with on-premise student housing, which mimics an ISP serving residential users. We evaluate the accuracy of our method via ground-truth gameplay in the lab on-campus. We then collect data in the wild, and present aggregated insights obtained over the course of four month spanning 673 hours of playtime. We found about 16% of game sessions was via browsers, which were mostly with high frame rate but low resolution. The PC app accounted for 93% of playtime, and was associated with all UHD resolution, though interestingly it tended to reduce frame rate from 60 fps to 30 fps more often, most likely so it could preserve the higher video resolution. The mobile app accounted for 1% of game sessions, and was with the worst performance in both frame rate and resolution. Knowing the mix of user setups will help ISPs tune their network functions to uplift user experience in a cost-efficient manner.

In our *fourth contribution* (§VI), we deploy our system with our partnered network operator, which exclusively hosts NVIDIA's GeForce NOW cloud gaming servers in Australia and New Zealand. Our real-time cloud gaming experience measurement is augmented by partial and anonymized GeForce NOW game session metadata, including game title, client subnet and access type. The measurement outputs are informing our partner operator to tune their network capacity and develop appropriate bundles and plans. Aligned with our ethics approval[2], we discuss our national-scale deployment insights over a one-month period, highlighting differences from our campus deployment. For example, we found similar patterns in game session duration and bandwidth across user setups; unified game lag, and varying graphic resolutions and streaming frame rates per network access type and client subnet. These findings provide precise reference for network optimization, such as ensuring consistent bandwidth levels for under-performing segments.

II. Cloud Gaming Background & Workflow

In this section, we provide an overview of current development in cloud gaming and highlight the challenges that arise for network operators (§II-A). We then discuss typical operational process of gameplay on a cloud gaming platform (§II-B).

A. Development of Cloud Gaming

In order to provide gamers with the opportunity to play graphic-intensive games on their everyday PCs without the need to invest in high-end devices, leading companies in graphic processing (*e.g.,* NVIDIA), cloud services (*e.g.,* Amazon), and gaming operations (*e.g.,* Sony) have embarked on the development of the "cloud gaming" business model, also referred to as Game-as-a-Service. This model moves on-device gaming processing to the powerful cloud compute clusters. Under this model, gamers can subscribe to the service and gain access to a dedicated cloud platform that supports a wide range of graphics-demanding games. The gameplay scenes are then rendered in real-time on the cloud servers and streamed to the user's device.

The popular cloud gaming platforms (NVIDIA's GeForce NOW [23], Microsoft's XBox Cloud Gaming [24], Sony's PlayStation PLUS [25] and Amazon's Luna [26]) are accessible on major OSes and support both console applications and browsers, except that XBox Cloud Gaming does not support macOS and is only available via its console application.

However, such operational mode shifts the requirement of running high-end games from expensive graphic processing

[1]We have obtained ethics clearance (UNSW Human Research Ethics Advisory Panel approval number HC211007) which allows us to analyze campus network traffic for application usage and behaviors. All user identities remain anonymous – no attempt is made to extract or reveal any personal user information, and all results presented are aggregated across the campus.

[2]We have obtained ethics clearance (UNSW Human Research Ethics Advisory Panel approval reference number iRECS5933), allowing us to report the measured cloud gaming characteristics associated with game session metadata for the Australia and New Zealand region in an aggregated manner. Note that all user identifies are anonymous and we made no attempt to collect or reveal any personal information.

Fig. 1. A typical cloud gaming process.

and computing hardware on user device to the network's capability of streaming high-resolution live video, along with other essential gaming requirements such as low latency and jitter. This can result in significant data consumption, often exceeding several tens of Megabits per second. In comparison to traditional online games that primarily exchange lightweight flags consisting of user inputs and server responses, which typically require only a few kilobits per second, cloud gaming places an unprecedented burden on carrier networks. If not properly managed, the increased network demands can lead to customer frustration and even prompt users to switch network providers. Recognizing the emerging challenges for network operators, we undertake a study that focuses on the network traffic characteristics of a representative cloud gaming platform, GeForce NOW with the aim to detect cloud gameplay and measure gamer experience. We have also validated our methods for the other three platforms, but will keep our discussions on these platforms succinct.

B. Operational Process of Cloud Game Sessions

Cloud gaming platforms serve as intermediaries that connect user devices and individual game servers. These platforms receive input from gamers including mouse movement, keyboard input, and upstream audio, and perform graphic and gaming computations on their behalf. Real-time gameplay scenes are streamed back to users via video and audio services. Therefore, a typical cloud gaming process involves two types of sessions that are in charge of platform administration and actual gameplay, respectively.

We now walk through an example play of a popular first-person shooting game (*i.e.,* Counter-Strike: Global Offensive or its abbreviation CS:GO) on GeForce NOW platform. As shown in the leftmost window of Fig. 1, after logging into the cloud gaming platform via either console application or browser, the gamer is directed to the "**platform session**", during which the gamer can browse available games and choose the one to play. In addition, some of the graphical settings such as resolution and frame rate are also options to be set by the gamer during the platform session. Once a gameplay is about to start, as shown by the second leftmost window in Fig. 1, the optimal cloud server for this gameplay is selected via a set of network measurements for key performance metrics, such as latency and throughput between the gamer device and candidate clusters.

After initial setup, the "**gameplay session**" begins, such as the CS:GO shooting gameplay visually depicted in Fig. 1. As will be soon discussed in §III, the gameplay session places significant demand on network bandwidth and latency, which directly impact the performance for cloud gaming. After exiting a gameplay session, a gamer returns to the platform session (the rightmost window of Fig. 1) to select/start next gameplay or finish the cloud game session.

III. Characterizing Cloud Gaming Traffic

We now delve into the network traffic characteristics of gameplay on the GeForce NOW platform obtained from our labeled traffic traces captured in our lab environment (§III-A). We begin by discussing the anatomy of service flows (§III-B) in both platform and gameplay sessions, and then focus on the profile of flows that deliver critical functions in gameplay sessions (§III-C). We also generalize our obtained insights to three other platforms including XBox Cloud Gaming, PlayStation Plus and Amazon Luna in §III-D.

A. Dataset

The GeForce NOW cloud gaming platform is available via both console applications and browsers on PCs (*i.e.,* macOS and Windows) and mobile devices (*i.e.,* iOS and android). Therefore, we capture packet trace files (PCAP) during cloud gaming sessions of a real-time shooting game (*i.e.,* CS:GO) and a massive multiplayer role-play game (*i.e.,* Path of Exile) on the above user setups, which can be categorized as either **desktop console application**, **mobile console application**, or **browser**. In what follows, we primarily focus on the insights from gameplays on the console application installed in a macOS desktop, while the differences in other setups (*i.e.,* Chrome browser in macOS, console application in Windows, Chrome browser in Windows, console application in android, and Safari browser in iOS) are also discussed throughout the section. To validate our obtained insights, we further collected traffic traces of 27 cloud game sessions for each type of user setups. The 27 cloud gaming sessions for each setup type cover three commonly available graphic resolutions from ultra high-definition (UHD), full high-definition (FHD), to high definition (HD), with either 30fps, 60fps or 120fps video frame rates.

B. Anatomy of Service Flows

As visually shown in Fig. 2, we first look at the anatomy of network communications between a gamer device and the cloud gaming platform as obtained from our analytical results. There are three types of flows that collectively serve a cloud gaming session, namely for **platform administration**, **platform management**, and **gameplay** that are illustrated as blue, yellow, and green arrows in Fig. 2, respectively.

Specifically, once a user opens the cloud gaming console application or browser, *i.e.,* entering a platform session, a series of administration flows (*i.e.,* 1 in Fig. 2) are initialized for

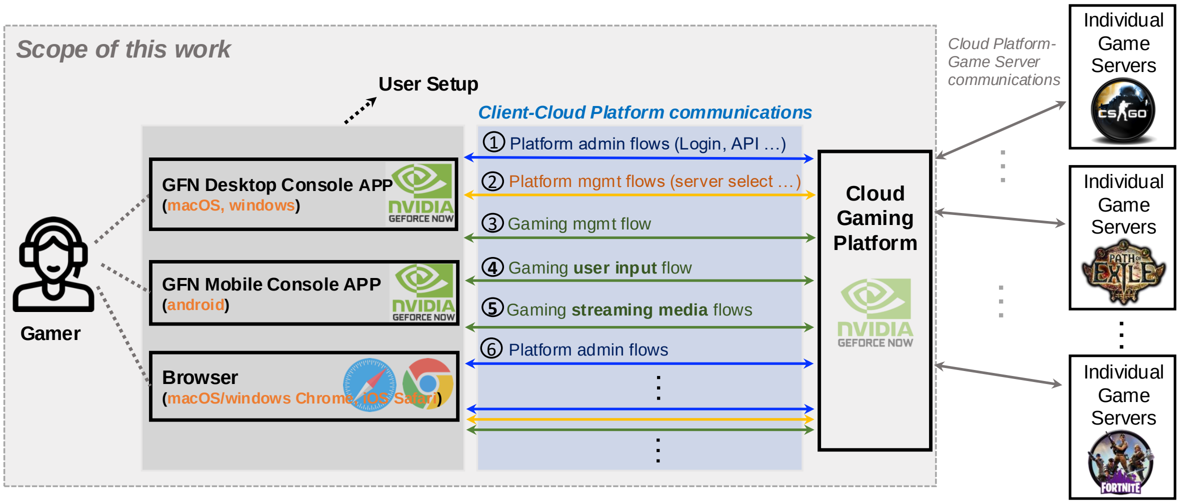


Fig. 2. Anatomy of cloud gaming communications via GeForce NOW (GFN) platform.

administrative support, such as content management system (CMS), API utilities, login portals, and account management. After the user selects a game to play, a number of platform management flows (*i.e.,* ② in Fig. 2) are started for network diagnosis and cloud server selection. Those flows are mapped to the second screenshot in Fig. 1 and hold critical roles in the successful initiation of the subsequent gameplay. During actual gameplay sessions, three types of flows are initialised, each serving a specific purpose. Firstly, there are gaming management flows (*i.e.,* ③) responsible for controlling the gameplay session (*e.g.,* measuring run-time latency) and exchanging metadata (*e.g.,* coordinating interactions between the client and the platform). In addition, there are specific gameplay flows dedicated to user input (from mouse, keyboard and microphone) and streaming media contents (*i.e.,* downstream video and audio), annotated as ④ and ⑤ respectively After a gameplay session, the platform administration flows are restarted as the user is returned to the platform session.

We now give a representative example of the above-discussed service flow usage as a time-series plot in Fig. 3. In this example, we played two games on the macOS PC console application. Our user activity could be separated into six stages. We logged into the cloud game platform and selected our first game (*i.e.,* CS:GO) to play in stage 1; played the CS:GO game in stage 2; returned to the platform in stage 3; entered the login and character selection page of our second game (*i.e.,* Path of Exile) in stage 4; entered the world of Path of Exile in stage 5, and finished our cloud gaming play in stage 6. We note that each of the six activity stages are either platform or gameplay session as defined in Fig. 1, thus, in Fig. 3, they are annotated by blue or red labels for the two session types, respectively.

In Fig. 3, we could see the timespans of all relevant flows that are with identical color indicating their flow types as consistent in Fig 2. The median throughput of each flow is indicated by the line thickness in Fig. 3. A representative collection of flows are labeled (as y-axis ticks) in the format of simplified service prefix (extracted from SNI or DNS records) and identifiable port numbers. The labels for other flows are not included for readability. Now we discuss the details per flow type.

1) Platform Administration Flows: Platform administration flows are shown as blue lines in Fig. 3. They are all sent to the service port $TCP|443$. After decoding packet headers, we confirm that they are all HTTPS flows toward the cloud gaming provider domains (*i.e., nvidia.com*, *geforcenow.com*, and *geforce.com*) for administrative services as indicated in their subdomain prefixes, such as content management system (CMS) [27] (*cms*) and frontend APIs (*gx-target-experiments-frontend-api*). Depending on their service purposes, some of the flows (*e.g., login* and *userstore*) are only active during the platform session, whereas others (*e.g., cms* and *events*) remain active during the subsequent gameplay sessions. As for their volumetric profiles, all of them are having small (or even negligible) amount of volume usage, *i.e.,* less than several Megabytes.

As discussed in the Appendix §A of the conference version [2], comparing the platform administration flows across different user setups, we have observed that the sessions via Chrome browser has most of the service flows seen in our discussed example, except for *cms* and *als* which are related to high-performant graphics. Besides, unlike PC setups, cloud game sessions via android mobile console application have limited usage of platform administrative flows that seem to only cover essential services such as *login*, *event*, and *userstore*.

2) Platform Management Flows: As depicted by the yellow horizontal bars in Fig 3, the platform management flows exhibit a relatively lower quantity compared to the platform administration flows. They are all HTTPS flows sent to service port $TCP|443$ of the GeForce NOW cloud server clusters, which are all associated with the cloud cluster domain *nvidiagrid.net*.

Unlike the above-discussed administration flows that provide support to the user administration on the platform, such as login, events, and user store, the platform management flows serve critical tasks relating to the delivery of subsequent gameplay session. In stage 1 of Fig. 3, we observe the first management (yellow) flow directed towards the subdomain prefix *gfnpc.api.entitlement-prod*, which grants the client ac-

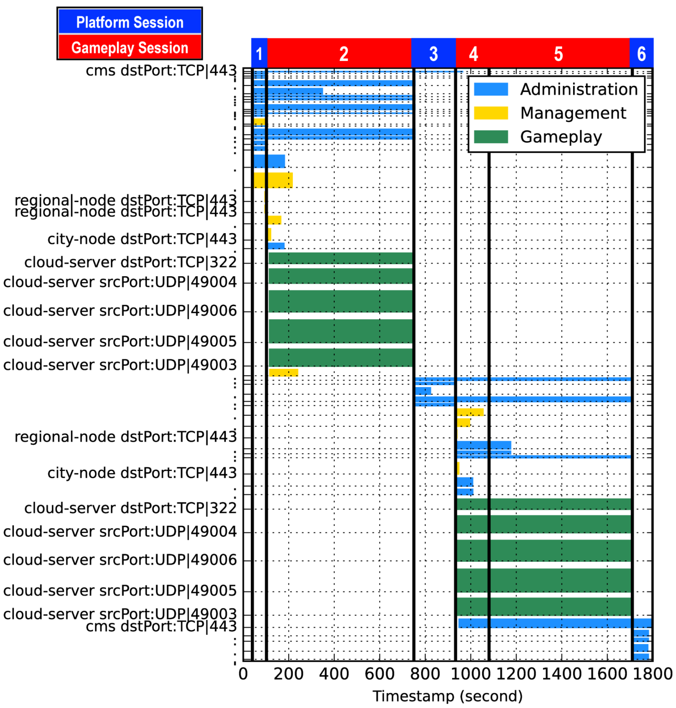


Fig. 3. Flow profiles of example gameplay via GeForce NOW desktop console application. Service prefixes and port numbers of representative flows are shown by their respective y-ticks. Throughputs of flows are indicated by their bar thicknesses (normalised by logarithmic functions).

cess to the GeForce NOW production system. This is followed by multiple flows toward subdomain prefixes in the format of *server_[location]_[partner]*, which serve the purpose of cloud server selection by measuring network performance metrics between the user and various available vantage points. As indicated by the service prefixes in Fig. 3 from top to down, the selection process progresses from the regional node to the city node and ultimately to each individual cloud server.

Given the indispensable roles of platform management flows, *i.e.,* system access and server selection, there is no significant difference that can be observed across different user setups, except that four flows toward the service prefix *img* of the production system domain *nvidiagrid.net* are seen in platform sessions on android mobile but not on PC setups.

3) Gameplay Session Flows: Once a suitable cloud server is successfully selected by the platform management flows, the actual gameplay session is started. As depicted by the green lines in stage 2 of Fig. 3, five gameplay session flows are initiated for the CS:GO gameplay. This observation remains consistent for all gameplays via console applications on both mobile and PC devices. Similarly, in stage 4 and 5 of Fig. 3, the same combination of five gameplay session flows can be observed for the Path of Exile gameplay.

Specifically, the first gameplay session flow is always directed to destination service port $TCP|322$ on the cloud server. This is followed by four Real-Time Transport Protocol (RTP) flows originating from client ports $UDP|49003$, $UDP|49004$, $UDP|49005$, and $UDP|49006$ toward dynamically selected service ports on the cloud server. According to NVIDIA Support [28], the four client ports are assigned for downstream audio, upstream audio, downstream video, and user input, respectively. We believe that the dynamic selection of service ports is performed by the first TCP flow for gaming management purpose, as illustrated in Fig. 2.

Notably, the flows originating from $UDP|49005$, responsible for downstream video, consistently consumed larger amounts of bandwidth than all other gameplay session flows. In the 10-minute CS:GO gameplay (stage 2), this resulted in downstream video data transfers of 1422MB as calculated from the packet capture file, while in the Path of Exile gameplay (stages 4 and 5), it reached 1961MB. As for the flows for downstream audio, upstream audio, and upstream user input, they consumed several MB, several tens of MB, and several tens of MB data transfers, respectively.

Similar insights are observed for gameplay session flows across console applications on both mobile and desktop devices. However, in browser-based gameplay, their gaming management flows is delivered using WebRTC protocol toward the service port $TCP|49100$ [29]. Additionally, gameplays on browsers utilize a single UDP flow to carry both downstream media contents and upstream user input, in contrast to the separate flows seen in console applications.

4) Key Takeaways: As a recap, platform administration flows and platform management flows are responsible for services between the user and the cloud gaming platforms before the start of an actual gameplay, such as login process, gaming catalog, and cloud server selection. Therefore, these two flow types do not directly impact the user-perceived game streaming quality during actual gameplay sessions. As for the gameplay session flows that stream video, audio, and user inputs, the sufficiency of network capacity allocated to those flows imposes direct impact on the user-perceived gaming quality and deserves prioritization over the other two types of flows. In what follow, we will focus on the gameplay session flows to analyze their volumetric profiles and packet statistics.

C. Profile of Gameplay Session Flows

We now focus on gameplay session flows (represented by the green lines in Fig. 3) that are critical to the cloud gaming experience. For flows that are active during a gameplay session, we analyze their volumetric profiles (§III-C1); benchmark their bandwidth consumptions across different levels of graphic quality including frame rate and resolution (§III-C2); and explore how video frame rate could be identified by packet size patterns of flows carrying downstream video data (§III-C3).

1) Flow Volumetric Profile: For our example gameplay discussed in Fig. 3, the inbound and outbound packet rates of the five gameplay session flows, including gameplay management, upstream user input, upstream audio, downstream audio, and downstream video, are presented in Fig. 4.

First, the **gameplay management flow** exhibits a low rate of one pair of packets every two seconds. It can serve as a reliable indicator for measuring network latencies between the user and the cloud platform by tracking the sequence and acknowledgment numbers in TCP packet headers.

Second, the upstream **user input flow** exhibits a packet rate ranging from 11 to 267 pps, depending on the user's activity

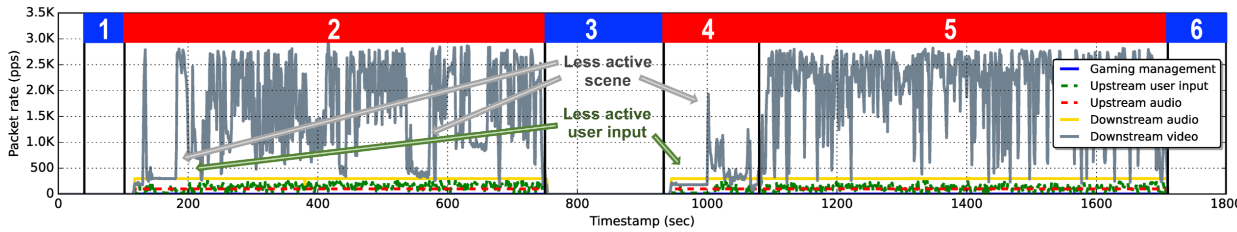


Fig. 4. Volumetric profiles of gameplay session flows (the green lines in Fig. 3).

level. As depicted by the green lines in Fig. 4, the baseline packet rate (11pps) occurs when the user is actively moving the mouse or typing on the keyboard, such as during the matching phase of a shooting game (beginning of stage 2 in Fig. 4) or the opening cinematic of a role-play game (beginning of stage 4). It is worth noting that the outbound and inbound packet rates of this flow are often equal. Regarding the throughput of the user input flow, the outbound direction exhibits minimum and maximum values of 1k and 82kbps, respectively, which are approximately ten times larger than those observed in the inbound direction.

Third, the two flows for **upstream and downstream audio** both have constant packet rates and throughputs in their respective directions. Specifically, the upstream audio flow exhibits a packet rate of 100pps with a throughput of 15kbps, while the downstream audio flow has a packet rate of 300pps with a throughput of 37kbps. These values remain consistent regardless of the status of the input/output voice, such as being muted, at low volume, or at high volume, which we thoroughly tested during our experiments. In their opposite directions (*e.g.,* inbound direction for upstream audio flow), both of them constantly have 2pps packet rate with 156bps throughput.

The most bandwidth consuming flow type, *i.e.,* **downstream video flows**, exhibit a packet rate that ranges from approximately 300pps during less active scenes (as shown in Fig. 4) to 3000pps. The corresponding bandwidth consumptions for these packet rates are 3Mbps and 34Mbps, respectively. It is important to note that during active gameplay periods, such as stage 5 in Fig. 4), the bandwidth consumption mostly remains at the upper bound level. On the other hand, the low packet rate is observed during inactive periods, which often coincide with the inactivity of the user input flows. However, there are exceptions during static scenes with frequent user inputs, such as the login scene of a gameplay.

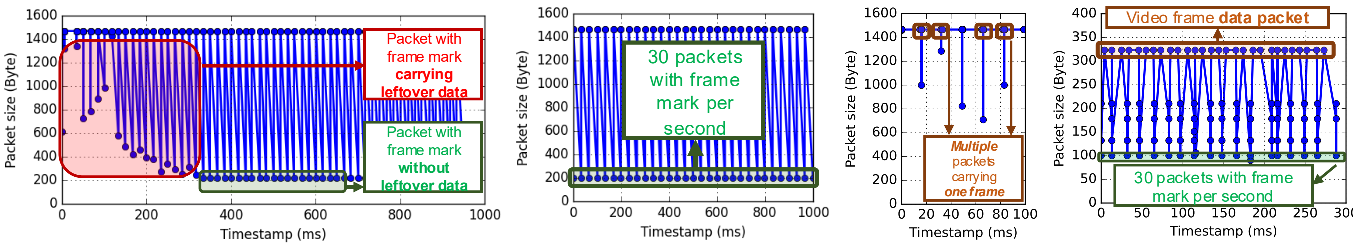
Similar patterns in packet rates and bandwidth consumption are observed for gameplay session flows across both desktop and mobile console applications. In the case of browser sessions with only one streaming flow, its volumetric pattern is primarily dominated by the downstream video content.

We note that the numerical results presented earlier were specific to the video configuration with 60fps for the frame rate and a resolution of 1920x1080 (FHD). However, users have the flexibility to choose from a wide range of frame rates and resolutions, either statically or dynamically adjusted based on the network conditions during gameplay. In the following analysis, we will discuss the bandwidth consumption of downstream video flows for different graphic configurations.

2) Bandwidths of Downstream Video Flows across Streaming Configurations: Frame rate and graphic resolution are configurable parameters for a cloud gameplay. To analyze the bandwidth consumption for downstream video flows, we manually selected various available frame rates (30, 60 or 120fps) and graphic resolutions (ranging from 3840x2160 to 1024x768). For browser sessions where only one gameplay flow is present, we also measured the bandwidth consumption as it is primarily influenced by downstream video content. As discussed for Fig. 4, the packet rate and throughput for downstream video flows stay at a peak range during active gameplay scenarios. The representative peak bandwidth consumption of those downstream video flows under different graphic configurations are reported in §3.3 of the preliminary version [2]. In general, lower frame rates and coarse-grained graphic resolutions always result in lower bandwidth consumption, both in active and less active scenarios. It is worth noting that by examining the peak bandwidth consumption of video flows and the current frame rate (which could be inferred from packet size patterns and will be discussed soon), it is possible for an ISP to infer the current graphic resolution of an active cloud gameplay by its user.

3) Packet Size Patterns in Downstream Video Flows across Frame Rates: Identifying the number of packets with a frame marker set in the RTP header of downstream video flows is a straightforward method to benchmark the current frame rate. In RTP flows, the frame marker indicates the completion of the currently transmitted video frame [30]. By counting the packets with the frame marker in the downstream video flow, we can accurately determine the frame rate of a cloud gameplay. However, in a high-speed network, this method requires decoding RTP packets that could introduce non-negligible overheads (*e.g.,* via multiple sequential packet parsers each decoding one packet layer). Besides, frame marks may not always be set correctly or being encrypted, makes RTP headers not decodable. For a lightweight and robust method, we have observed certain patterns in the packet sizes of downstream video flows to determine the frame rate without decoding the Ethernet/IP/UDP/RTP headers. This approach ensures resilience even when frame marks are not available. Now we look at several example downstream video flows shown in Fig. 5.

In general, a video frame is carried by two types of RTP packets. The first type carries all or the majority of video frame data and has a fixed large packet size (*i.e.,* 1466 bytes in console applications). The second type consists of smaller-sized packets (*e.g.,* 216 bytes) that carry frame markers, indicating the completion of a frame transmission. During gameplay sessions, the number of data packets required to



(a) MacOS app - 60FPS. (b) Windows app - 30FPS. (c) Multiple packets. (d) Windows browser - 30FPS.

Fig. 5. Packet payload sizes in representative downstream video flows (each packet is represented as a blue dot) as time-series plots.

transmit a video frame is dynamically adjusted based on the amount of video data in the frame. This can range from one packet to multiple packets, as visually depicted in Fig. 5(c). Importantly, there is always one small-sized marker packet indicating the end of each frame, which is highlighted by the green box in Fig. 5(a). These frame marker packet may have larger sizes (still smaller than the size of data packets) to accommodate any remaining video data from the previous packets. This is indicated by the red box in Fig. 5(a).

We observed a consistent pattern in the downstream video flows, where the number of packet groups (comprising several data packets followed by one marker packet) aligns perfectly with the current frame rate. An example of this pattern can be seen in Fig. 5(b), where we observe 30 packet groups within one second for a frame rate of 30fps. By analyzing the size patterns of these packet groups, we can accurately identify the frame rate being received by a cloud gamer in real-time.

In contrast to gameplay via console applications, the scenario for browsers is slightly different as there is only one RTP flow responsible for carrying downstream video, audio, and user input data. As illustrated in Fig. 5(d), each vertically aligned group of packets consists of several data packets corresponding to a video frame, followed by a frame marker packet, and three additional packets for downstream audio, upstream audio, and user input. Despite this difference, the consistent pattern observed in each packet group via browsers still aligns with the delivery of video frames.

4) Key Takeaways: The gameplay session flows that stream user input, downstream video and up/downstream audio exhibit flow-level volumetric profiles relating to the streaming quality (*i.e.,* graphic resolution and frame rate) and packet-level patterns directly determined by the frame rate. As will be soon discussed in §IV-B, by collectively analyzing the packet-level and flow-level volumetric attributes of the RTP flows, we can effectively measure the game streaming quality in an encryption-agnostic manner.

D. Generalizing Insights into Other Cloud Gaming Platforms

Similar insights were obtained in our analysis of other three cloud gaming platforms (*i.e.,* XBox Cloud Gaming, PlayStation Plus and Luna), as the underlying mechanisms for cloud gaming services are largely identical across providers. Specifically, each cloud gameplay session begins with a group of platform administrative flows for users to log in and navigate the available game catalog, platform management flows for selecting the cloud servers that will host the subsequent

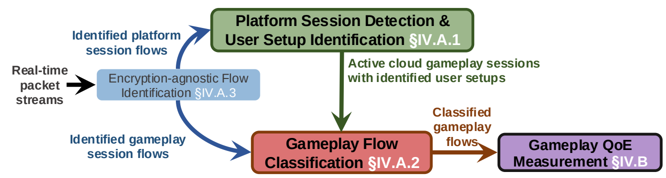


Fig. 6. Overview of our cloud gameplay detection and experience measurement methodology in §IV.

gameplay, and gameplay flows for synchronizing user input and streaming media. The primary variation among providers lies in their service domain names and prefixes. For example, XBox uses *xboxlive.com* for platform administration and *gssv-play-prod.xboxlive.com* for optimized game streaming; PlayStation Plus uses *playstation.net* and *gaikai.com*; and Luna uses *amazon.com* and *a2z.com*. Notably, only GFN sessions via its console application uses five concurrent flows for game streaming, whereas other providers, including their native applications and consoles, currently use a single flow.

IV. Gameplay Detection and Experience Measurement Methodology

In this section, building upon the insights gained from §III, we present the development of our network traffic analysis framework that detects cloud gaming sessions, identifies user setups, and continuously measures the performance metrics of each cloud gameplay session, as illustrated in Fig. 6. We then discuss the generalizability of our method to other cloud gaming platforms (§IV-D).

A. Detecting Cloud Game Sessions

As previously shown in Fig. 1 and 4, a cloud gameplay typically goes through two types of sessions, namely platform session (for platform administration, game browsing, and cloud server selection) and gameplay session. From our analysis, platform sessions exhibit identical usage of service flows in terms of user setups, which are important for a network operator to better understand their customer segments and potential causes of performance degradation. The flows in gameplay sessions directly determine users' cloud game experience.

Note that our method uses the identification of flows directed towards specific service domains based on the Server Name Indication (SNI) field in the TLS headers, which may

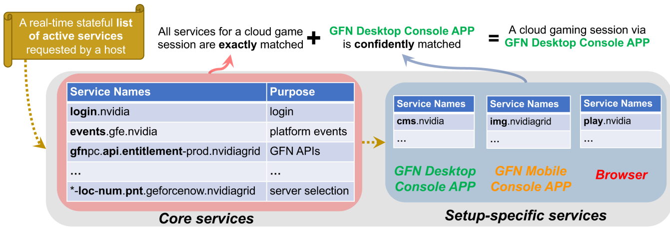


Fig. 7. Process for the detection of platform sessions and identification of user setups.

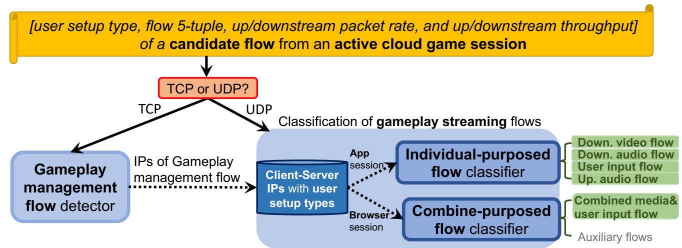


Fig. 8. Process for the classification of gameplay session flows, wherein the gameplay management flow detector is deterministic rule-based model and individual/combine-purposed flow classifiers are statistical models.

not be available due to the potential adoption of encrypted-SNI (ESNI). Therefore, we have also devised an encryption-agnostic approach (in §IV-A3) to overcome this challenge.

We now present our method to detect both platform and gameplay sessions.

1) Platform Sessions and User Setup Identification: User enters platform sessions either in the initial phase after launching to the cloud game platform (*e.g.,* stage 1 in Fig. 3) or in a subsequent phase after finishing each gameplay session (*e.g.,* stage 3 and 6 in Fig. 3). As discussed in §III-B, when a platform session begins, a series of HTTPS flows are initiated towards different service prefixes of the cloud gaming domains. These HTTPS flows contain service names in the server name indication (SNI) [31] fields of their TLS handshakes, which precede the encrypted application data communication. Therefore, we detect platform sessions by monitoring flows toward each service domain, extracted from their respective SNI records.

After analyzing our traffic traces across all user setups, we categorize the flows into two types including core services and setup-specific services. Core service flows are always initiated during the starting phase of platform sessions, regardless of the user setup type. There are also service flows specific to certain user setups. For instance, flows directed toward *login.nvidia* are always active, while flows toward *play.nvidia* only occur in platform sessions via browsers. Additionally, we have observed that the majority of core service flows follow a sequential order, starting from user login and progressing to server selection. In contrast, the occurrence of setup-specific service flows often exhibits randomness in their sequence.

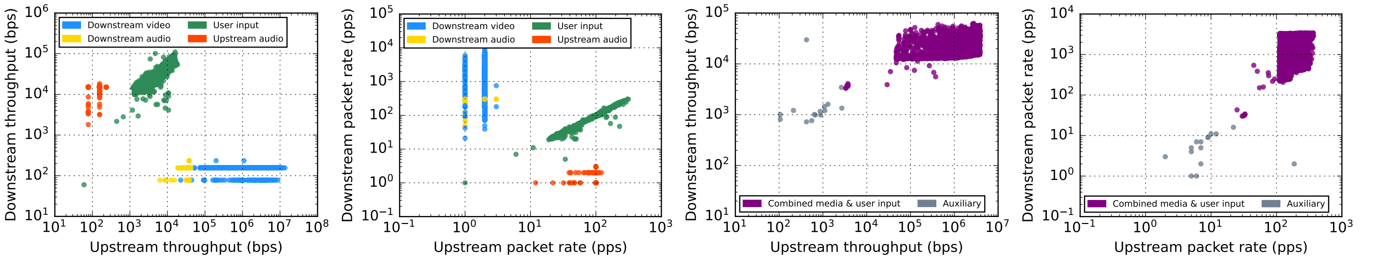
As shown in Fig. 7, we have developed a codebook correlation of domain names to detect the start of a cloud gaming play (through core services) and identify the user setup type (*i.e.,* desktop console application, mobile console application, or browser). A wildcard match of domains in the corresponding table will trigger a successful detection. For example, as demonstrated by the top portion of Fig. 7, the exact match of flows in core service table triggers the successful detection of a cloud gaming session. Additionally, a confident match in the desktop console application table (compared to the other tables) determines that the cloud game session is being played on a desktop console. For simplicity, we have provided only a snippet of our matching tables for GFN platform.

2) Gameplay Sessions and Gameplay Flow Classification: As a recap from §III-C, after starting a gameplay on the cloud platform, one TLS-encrypted TCP gaming management flow and several (one for browser and four for console application) UDP flows are started for media and user input streaming. Cloud gaming sessions from console applications have their TCP gaming management flows directed to service port numbers $TCP|322$, while browsers use port $TCP|49100$. Irrespective of the user setup types, all gaming management flows have their **service names** (extracted from SNI fields) following a consistent pattern of *a-b-c-d.pnt.nvidiagrid.net*. It is important to note that the *a-b-c-d* in this pattern represents the IP address *a.b.c.d* of a cloud server assigned to a gameplay session, which is also the server IP for the subsequent gameplay UDP flows.

We devise our method to identify gameplay session flows based on their service names and five-tuples for active users detected in the platform sessions (§IV-A1). Subsequently, these gameplay session flows are classified for their purposes, as defined in §III-C, by considering the user setup types, flow five-tuples, and volumetric profiles (*i.e.,* packet rate and throughput).

The classification process is shown in Fig. 8. This process takes standard input attributes of candidate flows from active users as scoped down by their service name patterns shown as the yellow banner in Fig. 8.

The first module of the classification process, named "gameplay management flow detector", is to **classify gameplay management flows** using deterministic flow rules on metadata fields. For GeForce NOW, a candidate TCP flow (identified by its service name) sent to port $TCP|322$ or $TCP|49100$ from an active user are labeled as console application or browser, respectively. For the other three cloud gaming platforms, the rule matches candidate TCP flow to $TCP|443$ on the selected cloud server. As shown by the dashed arrow leaving the detector, the five-tuple (including client and server IP addresses) of a detected gameplay management flow and its associated user setup type will be stored in a runtime database to be used by the classifier for gameplay streaming flows that are initiated shortly afterward. In our implementation, the runtime database is placed in-memory with each flow five-tuple indexed by hashing. After tuning for our regional deployment, each entry is assigned a five-second idle retention period, which provides a sufficient window to track active flows while maintaining stable memory usage below tens of Megabytes. With this configuration, the entry lookup latency remains constantly below 0.03s during peak hours, resulting in negligible impact on the detection accuracy of subsequent



(a) Throughput (individual-purposed). (b) Packet rate (individual-purposed). (c) Throughput (combine-purposed). (d) Packet rate (combine-purposed).

Fig. 9. Distributions of volumetric attributes for individual-purposed and combine-purposed gameplay flows.

game streaming flows, which we empirically observed to arrive within 0.1s to 0.5s.

The second step, shown in the right blue region of Fig. 8, is to **classify gameplay streaming flows** of a cloud gameplay using machine learning classifiers on volumetric attributes. In our implementation, well-tuned random forest models are deployed, which achieves satisfactorily good accuracy (*i.e.,* above 99%) for the small number (*i.e.,* 4 in this task) of input attributes. We acknowledge that there are other options for the models such as lightweight neural networks, especially when the number of attributes becomes large, which are engineering alternatives for this study and not explicitly discussed. Depending on the user setup types of a gameplay session, classifiers for either individual-purposed or combine-purposed flows will be used. The **individual-purposed flow classifier** is used for streaming flows during app sessions. One streaming flow delivers a certain type of data including downstream video, downstream audio, upstream audio, or upstream user input. Classification of these flows is based on their volumetric attributes, specifically the packet rate and throughput, in both upstream and downstream directions. While we acknowledge that prior works in network flow classification have also used other attributes derived from packet streams such as inter-arrival time and entropy in packet sizes, we select the two volumetric attributes in both directions to classify gameplay streaming flows for two reasons. First, they are lightweight to compute during runtime as the count of packets and summation of their sizes per one-second interval. Second, the selected volumetric attributes can effectively separate the different types of streaming flows. As shown in Fig. 9(a) and 9(b), they display a clustered distribution for the four types of streaming flows, which can be captured by a well-trained statistical model with nearly 100% accuracy in our deployment.

In the case of browser sessions, a candidate UDP flow can either be a combined-purposed streaming flow that carries both media and user input data or an auxiliary flow, such as one used by the STUN WebRTC service. The **combine-purposed flow classifier** is used to identify the streaming flow by its volumetric attributes. As shown in Fig. 9(c) and 9(d), auxiliary service flows have packet rates and throughout consistently smaller than those of the combined streaming flows. This difference allows our well-trained statistical model to capture them with 100% accuracy.

In §IV-B, we will discuss the continuous monitoring of the streaming flows and gaming management flows to infer user experience per cloud gameplay session.

3) Encryption-Agnostic Service Flow Identification: The recent proposals of SNI encryption techniques have sparked discussions in the industry and could potentially be adopted in the near future. If that happens, current network analysis relying on SNI signatures will become ineffective. To tackle this anticipated issue, we develop an encryption-agnostic technique (the blue module in Fig. 6) as an enhancement to our existing flow detection using service names extracted from the SNI field. This enhancement allows us to identify flows with their service names associated with platform and gameplay sessions by analyzing packet payload sizes, without the need to inspect TLS headers.

According to prior research, flows with specific functionalities, such as console gaming [6], VoIP/video/file transfer/chat/browsing [32], [33], and encrypted web [34], can be detected by the distinctive distribution or sequence of packet sizes they exhibit. Building on these findings, to detect platform and gameplay management flows without inspecting SNI, we leverage the sequence of payload sizes in the first few packets of a TCP flow, which contain predefined service requests after TCP three-way handshakes. As shown in §4.1 of the preliminary version [2], there are distinguishable patterns in the sequence of packet payload sizes for gameplay management flows. With the precise specification of these signatures, we have achieved 100% accuracy in detecting cloud gaming sessions during our lab evaluation. It is important to note that this technique is designed for platform HTTPS flows and gameplay management TCP flows. The detection of gameplay UDP flows that uses flow metadata and volumetric attributes as discussed in §IV-A2, does not rely on service name signatures and is already encryption-agnostic, meaning it is not affected by future SNI encryption.

B. Measuring Game Streaming Quality

The streaming quality of a cloud gameplay session is primarily determined by three factors: the synchronization speed of the user's mouse/keyboard input with the cloud platform (measured by game lag in §IV-B1), the smoothness of the streamed gaming scene (measured by streaming frame rate in §IV-B2), and the clarity of the gaming graphics (indicated by graphic resolution in §IV-B3).

In this section, we propose metrics to monitor these three key performance indicators derived from real-time volumetric

Algorithm 1 An algorithm for measuring the streaming frame rate (*i.e.,* frame count of an interval) of cloud gameplay using packet size patterns of downstream video flows.

Input: *packets* in game streaming flow; measurement interval ΔT; payload size margin $\Delta size$
Output: measured *frame_count*

1: $frame\_count \leftarrow 0$
2: $t_{start} \leftarrow packets[0].arrival\_timestamp$
3: $size_{max} \leftarrow packets[0].payload\_size$
4: $flag_{max} \leftarrow FALSE$
5: **for** p **in** packets **do**
6:     **if** $p.arrival\_timestamp > t_{start} + \Delta T$ **then**
7:         **print** $frame\_count/\Delta T$
8:         $frame\_count \leftarrow 0$
9:     **end if**
10:     **if** $p.payload\_size > size_{max}$ **then**
11:         $size_{max} \leftarrow p.payload\_size$
12:         **continue**
13:     **end if**
14:     **if** $p.payload\_size < size_{max} - \Delta size$ **then**
15:         **if** $flag_{max}$ is $TRUE$ **then**
16:             $frame\_count \leftarrow frame\_count + 1$
17:             $flag_{max} \leftarrow FALSE$
18:         **end if**
19:         **continue**
20:     **end if**
21:     $flag_{max} \leftarrow TRUE$
22: **end for**

statistics of gameplay session flows. Our metrics are computed from transport-layer headers and packet sizes, therefore, are agnostic to the encryption of application-layer headers and payloads.

1) Game Lag: The first metric is game lag (*i.e.,* client-platform latency), which represents the response time from the moment a gamer inputs commands with their keyboard or mouse till those commands are executed by the cloud server. As discussed in §III-B3, there is a single gaming management flow over TCP that remains active throughout the entire gameplay session. This flow is detected by our methodology proposed in §IV-A2.

The gaming management flow exhibits a constant packet rate of one pair of TCP packets every two seconds, where each pair consists of an upstream and downstream packet with matching **sequence** and **acknowledge numbers** in their TCP headers, as explained in §III-C.

To measure the real-time latency experienced by the cloud gamer, we continuously monitor the arrival timestamps of each packet pair (identified by their sequence and acknowledge numbers) within the gaming management flow. Specifically, we track the timestamps t_{up} and t_{down} and calculate the latency Δt between t_{down} and t_{up}.

2) Game Streaming Frame Rate: The second metric we consider for user experience is the video frame rate being streamed to the cloud gamer. As discussed in §III-C3, a higher frame rate, such as 60fps, imposes stricter network requirements in terms of higher bandwidth and lower packet loss. In return, it offers the user a smoother gaming video experience.

To track this performance metric, we leverage the periodic patterns of packet payload sizes observed in downstream video flows for both console applications and browsers during each gameplay session. These periodic patterns serve as a **direct** measure of the video frame rate, as explained in §III-C3 and illustrated in Fig. 5. Considering that GeForce NOW offers frame rate options of 30fps, 60fps, and 120fps, we expect the observed count of periodical patterns per second to closely align with one of these three values.

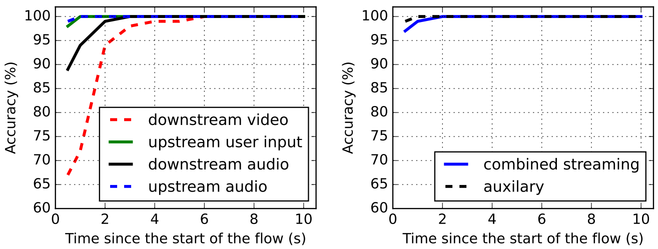
The pseudocode block in Algorithm 1 shows our approach for measuring frame rate from downstream video flows. The method takes as input the downstream packets in a gameplay video flow. It also requires an interval ΔT (set to 1 second in our implementation) that determines the frequency of measurements, and a payload size margin $\Delta size$ (fine-tuned using ground-truth sessions to 1 byte) that allows for variations in the payload size of full-size video packets in the flow.

The algorithm initializes four assisting variables from line #1 to line #4, which captures the frame count ($frame\_count$), starting timestamp (t_{start}), the maximum packet payload size ($size_{max}$), and a flag indicating the presence of packet with the maximum payload size ($flag_{max}$), all for the current time interval. Within the loop that processes the packet streams (line #5), the measured frame rate per interval ΔT is reported and reset from line #6 to line #9. Additionally, the algorithm determines the full payload size of video packets in the monitored flow from line #10 to line #13, which appears in the beginning few packets of each frame as discussed in §III-C3. From line #14 to line #21, the algorithm captures the periodical pattern observed in the packet payload sizes of a video flow once the maximum packet payload size has been detected (*i.e.,* $flag_{max}$ is $TRUE$). This pattern includes sequences of full-sized video packets followed by smaller-sized ones carrying frame markers and/or remaining data.

3) Gaming Graphic Resolution: Our third performance metric is graphic resolution, which represents the visual quality of the graphics being streamed to the cloud gamer by the cloud platform. The graphic resolution, along with the video frame rate, determine the bandwidth consumption of a downstream video flow, as discussed in §III-C2. As just discussed, we can directly determine the current level of **video frame rate**. Therefore, to diagnose the real-time graphic resolution, we measure the current **bandwidth consumption** of the video flow and refer to the mapping of the peak bandwidth consumption under different frame rate and graphic resolution. This mapping enables us to deduce the current graphic resolution based on the given throughput and frame rate.

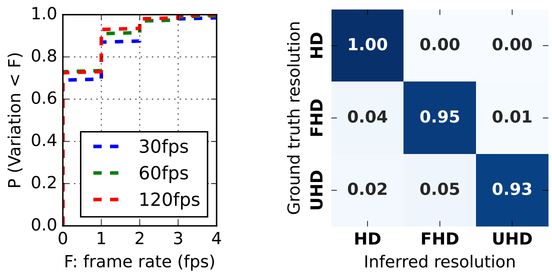
C. Evaluation with Ground-Truth Cloud Gaming Sessions

As will be discussed in §V, we have implemented a fully-functional prototype of our cloud gaming detection and experience measurement framework in our university campus. We begin by evaluating the accuracy of our system by playing cloud games from our lab on campus and comparing it to ground truth. We had volunteers play a total of 621 sessions of GeForce NOW gaming from our lab, and the ground truth they recorded was compared against the outputs reported from our system monitoring campus-wide traffic for the specific IP address of the lab devices. The sessions were designed



(a) Individual-purposed. (b) Combine-purposed.

Fig. 10. Classification accuracy of streaming flows that serve cloud game sessions in the manner of (a) individual-purposed or (b) combine-purposed. The results are reported based on the classifications using the first 0.5 to 10 seconds of volumetric attributes for each flow type.



(a) Frame rate. (b) Graphic resolution.

Fig. 11. Evaluation results for streaming quality inference accuracy, including (a) the variation between measured and ground-truth frame rates, and (b) the accuracy of inferred graphic resolution bands compared to ground-truth labels.

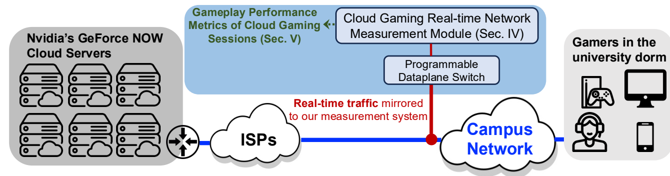
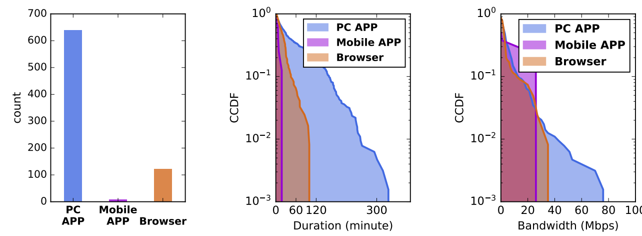


Fig. 12. The measurement system deployed at our university campus network.



(a) #Session. (b) Session duration. (c) Bandwidth.

Fig. 13. GeForce NOW cloud gaming usage patterns from our five-month campus network deployment.

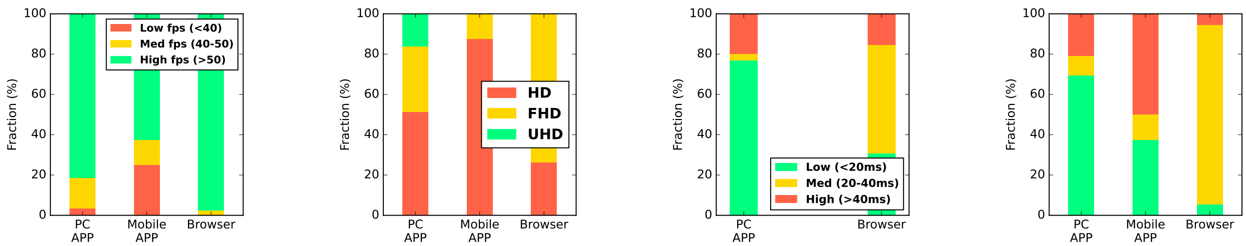
to encompass various streaming configurations, including user setups (PC app, mobile app, and browser), frame rates (120fps, 60fps and 30fps), and video resolution bands (FHD, HD, and SD).

Our system correctly reported the detection of all the cloud gaming sessions immediately upon commencement of actual gameplay, and the user setup (PC app, mobile app and browser play) was also identified with 100% accuracy. For the classification accuracy of game streaming flows carrying various content types, as shown in Fig. 10, an overall accuracy of about 99% is achieved with the volumetric attributes averaged from the first four seconds since the start of a session, and become nearly 100% for the first five seconds. This indicates that the volumetric difference of streaming flows carrying various content types become deterministically clear after 5 seconds. Therefore, in our engineering implementation, we use the first five seconds of volumetric attributes for classification.

For the metrics describing game streaming quality, the frame rate and graphic resolution are inferred by our algorithmic method from stochastic payload size patterns and throughput of each candidate flow. Our evaluation results show that both metrics are measured accurately compared to the ground-truth values collected on the client side, where traffic traces were recorded and analyzed. Specifically, the CDF plot in Fig. 11(a) shows the variation of measured frame rate using our method versus the ground-truth frame rate from the decoded frame markers. Less than 2 to 3 fps deviations for frame rates per second are observed for sessions set to three different frame rate bands including 30, 60, and 120 fps. For the classification accuracy of graphic resolution provided in Fig. 11(b), we see an overall accuracy of above 95%. All HD sessions are correctly inferred, while a small fraction of FHD and UHD sessions are mislabeled as lower resolution bands. Upon investigating the session-level volumetric profiles, we found that the misclassified instances all correspond to less intensive games with relatively static in-game scenes, such as card games like Hearthstone. This observation suggests a potential research direction to incorporate cloud game context into streaming quality measurement, which is beyond the scope of this paper.

D. Discussion on Generalizability and System Optimization

For the **generalizability**, we have validated with ground-truth dataset that the developed methods on gaming session detection, user setup identification, and gameplay performance measurement can also be generalized to other three popular cloud gaming platform with platform-specific signatures obtained from ground-truth traffic traces, including service domains, packet payload sizes, RTP port number, and flow volumetric criteria. Also, the measurement methods for game lag and video frame rate do not require signature thus are directly applicable. As also evidenced in other research works [20], [35], cloud gaming platforms on the market today leverage a common mechanism. Due to potential variations of cloud gaming traffic patterns over time, the changes of considered metrics (such as protocol type, volumetric statistics, and flow service domains) may diminish the effectiveness of a trained classification model. Consequently, retraining the model becomes necessary for optimal performance. Additionally, if significant modifications occur in the network anatomy of cloud gaming sessions (as briefly captured in Fig. 8), such as the changing between individual-purposed and combine-



(a) Steaming frame rate. (b) Graphic resolution. (c) Game lag (wired LAN). (d) Game lag (wireless LAN).

Fig. 14. GeForce NOW cloud gaming experience insights from our five-month campus deployment.

purposed gameplay streaming flows on a certain user setup type, adjustments to our model placement will be required.

We also acknowledge that the **system** can be **optimized** for telemetry performance. For instance, the in-memory flow volumetric telemetry can be optimized with compacted sketches [36], lightweight bitmaps [37], or space-efficient bloom filters [38] for better searching efficiency and memory utilization especially when handling large traffic volume. In this paper, our design focuses on the traffic analysis process to accurately detect cloud gaming sessions, classify gameplay flows, and measure streaming quality. As a result, system-level optimizations are left for future work.

V. Measurement Insights into a University Campus Mimicking Residential ISP Network

We have implemented a fully-functional prototype of our cloud gaming detection and experience measurement methodology in a large University campus with tens of thousands of students, including several hundred who reside in the dorms. As visually shown in Fig. 12, our system (highlighted in the blue region) takes a mirrored feed of all traffic to/from the campus, obtained via passive optical taps on the fibres connecting the campus to the Internet. The mirrored traffic is aggregated by a Tofino programmable switch before being forwarded to two 10Gbps network interfaces on a commodity blade server (configured with an 8-core Intel Xeon E5-2620 CPU and 64 GB DDR4 RAM) running our measurement module. We now present some insights obtained from the university campus deployment, as measured by our system over the five months from November 2023 to February 2024, on how user settings in GeForce NOW cloud gaming impact network bandwidth demand and end-user experience. For ISPs hosting residential users, such visibility can help them better understand their cloud gaming customer profiles, troubleshoot experience problems, and optimize network functions to better support cloud gaming flows using network slices, priority queues, and network APIs.

A. User Settings and Bandwidth Demand

During the five-month period when campus Internet usage is primarily by research students, staffs and dormitory residents, our system tracked a total of 769 cloud game sessions. These sessions accounted for 673 hours of playtime and consumed 8.02 TB of data.

As shown in Fig. 13(a), the majority of gameplay sessions (83%) and gameplay time (93%) were from the PC app, followed by browsers (16% of sessions and 7.1% of total playtime), and mobile app (1% of sessions and 0.2% of playtime). This is of interest to ISPs because the bandwidth demanded by the gaming streams vary across these platforms – Fig. 13(c) shows the bandwidth distribution (measured over 1-second intervals) on the three platforms. The browser almost never exceeds 30 Mbps, while the PC and mobile app have significant duration with bandwidth demand in the range of 30-75 Mbps. While the underlying reason for this becomes clear when we examine video frame rates and resolutions next, visibility into app/browser mix may better equip ISPs in planning and provisioning their network capacity as cloud gaming grows.

Interestingly, compared to the usage pattern in May 2023, as reported in our preliminary work [2], which corresponds to a busy month during a teaching semester, we observed significantly fewer sessions played on mobile devices during our current measurement period (from November 2023 to February 2024, spanning the summer holiday for coursework students). A similar observation can also be made for browser sessions. As will be discussed next, we can clearly see that cloud gaming players mostly play on PC app which can provide the best user experience.

B. User Settings and Gaming Experience

We saw in §IV-C that frame rate can vary significantly, so for ease of depiction we group it into three bands: Low (<40 fps), Medium (40-50 fps) and High (>50 fps). Video resolution has already been banded into UHD, FHD, and HD. For the three user platforms (PC app, mobile app, and browser), we depict the percentage of time that the cloud game video operates at the three frame rate bands in Fig. 14(a), and the percentage of time the video renders in the three resolutions in Fig. 14(b). It is very interesting to note that browser gaming almost always has high fps but never goes to FHD resolution, whereas the PC app has FHD resolution for 42% of the time but drops fps to medium or low about 10% of the time. The mobile app provides the best mix of fps and resolution overall, due to the advantage that it only has to work on a smaller screen. ISPs might use such visibility into the trade-offs on frame rates and resolution of cloud gaming sessions to better troubleshoot and support their customers on specific platforms.

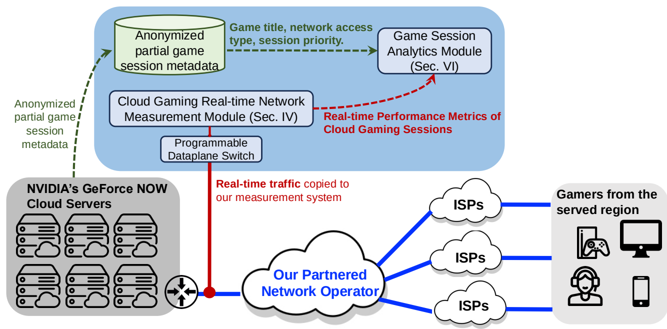


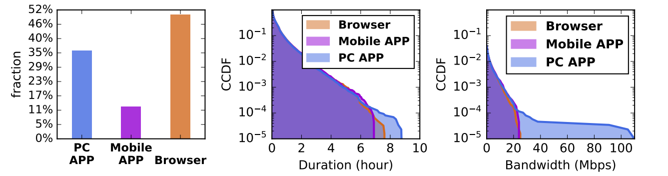
Fig. 15. Our Measurement system deployed by our partner network operator hosting NVIDIA's GeForce NOW cloud gaming servers.

Another interesting observation made by our system is in regards to the game lag across the platforms. To keep the comparison fair and not be influenced by wireless characteristics, we first limited it to wired end-hosts (which are identifiable by IP address being on a different subnet to the WiFi network). Fig. 14(c) shows that lag (averaged over 1-second intervals) stays under 20ms (the recommended threshold provided by NVIDIA) 75% of the time when played via the app, compared to 25% when played via the browser. We believe this is because the app is better optimized for gaming than the browser – indeed, we had shown in §III-C1 that the app maintains separate flows for user input and media, whereas the browser mixes all traffic into the one flow, which can lead to degraded jitter performance. ISPs may use such information to encourage customers who receive degraded experience to move from browser-based to app-based cloud gaming user setups.

As for the wireless WiFi network, we can make a similar observation that app sessions, including both PC app and mobile app, provide lower game lag compared to browser sessions. Also, PC app has 70% sessions low-lag, outperforming mobile app with only 38% low-lag sessions. Last, both PC app and browser sessions on the wireless network (as shown in Fig. 14(d)) perform worse in game lag compared to sessions on the wired network (as shown in Fig. 14(c)).

VI. Insights into a Network Operator Hosting NVIDIA's GeForce NOW Cloud Servers

We deploy our system for cloud game experience measurement in a commercial operator hosting the GeForce NOW cloud gaming servers in our region (*i.e.,* Australia and New Zealand). Measurements from our system are combined with metadata shared by the operator, allowing us to evaluate cloud gameplay performance across various user setups, game titles, network connectivity types, and client subnets. We outline the deployment architecture in §VI-A, provide baseline aggregate statistics in §VI-B, and highlight insights into the factors impacting game experience in §VI-C. The study in this section is complaint with ethics approval iRECS5933 obtained from the UNSW Human Research Ethics Advisory Panel.



(a) #Session. (b) Session duration. (c) Session bandwidth.

Fig. 16. Cloud gaming usage pattern collected in the wild from commercial partner hosting GeForce NOW cloud servers for the region.

A. Deployment System Architecture Overview

Fig. 15 shows the architecture of our measurement system, deployed by our partner commercial operator hosting the NVIDIA GeForce NOW cloud gaming servers. The game servers (left) exchange traffic with gamers (right) across any of a number of ISPs.

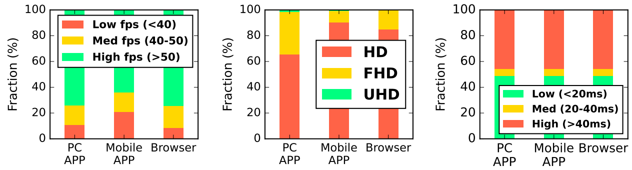
Our partner facilitated a copy of all the gaming traffic to our system which analyzes it in real-time. Specifically, the traffic copying process first uses a passive optical tap that mirrors all packets transiting over a fiber cable with a common industrial setting of 20/80 split for signal strength, incurring negligible packet transmission overhead. The Tofino 2 programmable switch that receives mirrored traffic performs lightweight packet format checking and filtering before dispatching upstream and downstream packets to the two network interfaces on the measurement server.

The measurement accuracy has been validated using ground-truth sessions played by our research team and our industry partner, showing no noticeable performance degradation compared to our lab evaluation. This is within our expectation as the inference criteria is specific to the unique characteristics of cloud game streaming flows, such as the deterministic difference in the volumetric profiles of flows for upstream user input versus downstream video and the periodical patterns in packet payload sizes determined by the streaming frame rate, which do not change at different vantage points or client networks.

In addition, our partner also shared anonymized metadata for each game session, comprising the game title, network access types (*i.e.,* wired, WiFi or cellular mobile), client subnet, and some other administrative information. The cloud gaming experience measures are fused with the metadata to produce the insights as discussed next.

B. Usage Statistics and Bandwidth Demands

Data was collected and analyzed over a one month (31 days) period from 8th July 2024 to 7th August 2024, comprising hundreds of thousands of game sessions spanning tens of thousands of hours of gameplay (we cannot report exact numbers due to commercial confidentiality restrictions). The game platform, duration, and bandwidth distribution are shown in Fig. 16. It is interesting to note the differences this data from the wild exhibits compared to the data from the University campus shown in Fig. 13. First, unlike the campus where most games were played on a PC (Fig. 13(a)), a significantly higher fraction of games in the wild are played via the mobile



(a) Steaming frame rate. (b) Graphic resolution. (c) Game lag.

Fig. 17. Cloud gaming experience insights across **user setups** from our one-month deployment at our partnered network operator hosting GeForce NOW cloud servers.

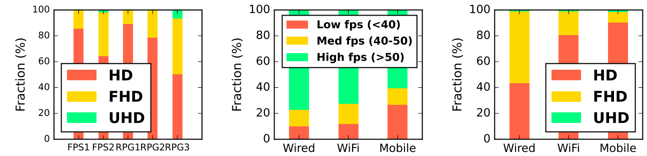
app or browser (Fig. 16(a)). Further, the session durations are very similar across the three platforms (Fig. 16(b)) and their bandwidth demands are very similar as well (Fig. 13(c)). Interestingly, over 99.99% of cloud gaming sessions in the wild consume peak bandwidth of less than 20Mbps, unlike in the University campus where a minority of PC app sessions exceeded 40Mbps (Fig. 13(c)).

C. Cloud Gaming Experience and Impact Factors

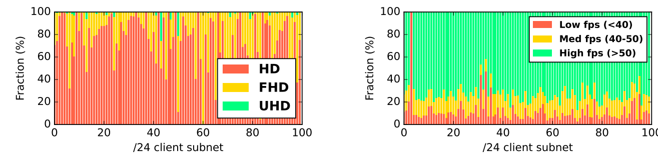
By coupling session metadata information from the GeForce NOW system with measurements from our system on gaming experience, we are able to provide interesting insights on how factors like user setup, game title, network access technology, and client ISP subnet impact experience.

1) User Setup: In Fig. 17 we depict how the three key experience metrics (*i.e.,* streaming frame rate, graphic resolution, and game lag) vary by user setup. The frame rate (Fig. 17(a)) is best for browser sessions, while being marginally lower for the PC app followed by the mobile app – this is no unlike what we had observed in the campus (Fig. 14). The graphic resolution (Fig. 17(b)) is worst for the mobile app, similar to what we had noted in the campus (Fig. 14(b)). Interestingly, the game lag (Fig. 17(c)) is distributed identically across all three platforms (50% in the low lag range, 6% in the medium lag range, and 44% in the high lag range), unlike the significant differences we saw across these platforms in the campus (Fig. 14(d)). We suspect this is because the lag is more influenced by the acess ISP and the routing paths they use, and less so by the user setup. Indeed further results shown below will also demonstrate that game lag exhibits no observable variation across all other impact factors considered in this study.

2) Game Titles: Over 500 game titles are frequently played by Australian and New Zealand cloud gamers, with the five most popular ones including two first person shooting (FPS) games and three role playing games (RPG) accounting for over 60% of the total playtime in our one-month dataset. To comply with our partner's commercial confidentiality requirements, we anonymize the game titles and report only aggregated session experience (Fig. 18) as measured by our system. Interestingly, no significant difference is observed in streaming frame rate and game lag across all 500 popular game titles; however, as shown in Fig. 18(a), there is variability across game titles (*e.g.,* RPG3 and FPS2) in terms of graphic resolution. We are uncertain of the underlying cause for this, which could be the nature of the game or a skew in the user platform on which each game title is played.



(a) **Game title** on resolution. (b) **Access type** on frame rate. (c) **Access type** on resolution.



(d) **Client subnet** on resolution. (e) **Client subnet** on frame rate.

Fig. 18. Cloud gaming experience impacted by **game title, network access type and client network** from our one-month deployment at our partnered network operator hosting GeForce NOW cloud servers.

3) Network Access Type: We compared gaming experience across the type of access network, namely wired, WiFi, and mobile (4G/5G). Even though there was no discernible difference in the game lag across these network types, we doubt appreciable difference in frame rates (Fig. 18(b)) and graphics resolutions (Fig. 18(c)). Not surprisingly, sessions played on a wired Ethernet connection experience the best frame rate and graphic resolution performance, while mobile networks exhibit low fps twice as often and lower resolution for 90% of the game sessions. This suggests that there is a potential for mobile network operators to create premium gaming services leveraging slicing technology and associated API driven Quality-on-Demand capabilities that are maturing in the industry [8].

4) Client ISP Subnet: We compare experience for clients connected from different subnets to understand if experience varies by the gamer's geographical location and access ISP. We consider the top-100 "/24" subnets, each of which had statistically significant sample size (at least 1000 game sessions). The graphics resolution (Fig. 18(d)) and frame rate (Fig. 18(e)) reveal that certain subnets exhibit worse gaming experience than others (*i.e.,* with a higher proportion of red in the respective bars in both figures). For instance, subnet #4 has all game sessions limited to HD resolution accompanied by a frame rate under 40 fps. A deeper investigation reveals that this subnet is not dominated by a particular user setup, game title, or network access type, but originates from an ISP that is coming over a trans-oceanic link. We have also observed that the subnets belonging to an ISP for education providers have noticeably higher fraction of sessions with game streaming frame rates exceeding 50fps compared to other subnets, while the distributions of graphic resolutions remain largely similar. This is possibly because universities and schools in the region are provisioned with sufficient network resources, and the game sessions are often played on portable laptops. We acknowledge that further associating the measured game streaming experience with subnet purposes such as residential, commercial and mobile could lead to more insightful discussions. However, this is beyond the scope of our study due to lack of reliable ground-truth labels for subnet classification.

VII. Related Work

Cloud gaming has been the subject of prior studies that have explored various aspects [39], including cloud architecture [40]–[42], computing resource provisioning [43]–[45], gameplay video encoding [46], [47]. In chronological order, we discuss representative examples of prior works that have focused on enhancing gamer experience for different stakeholders, including cloud platform operators and mobile developers. It is important to note that our work specifically focuses on the cloud gaming experience for ISPs. The authors of [48] discussed several elements in the cloud platform that can have signifiant impact on user experiences if not well optimized. Through a measurement study across Europe, the authors of [49] suggested optimal sever selection and control methods to achieve minimized lag for cloud gamers using mobile platforms. In [50], the authors quantified user perception of graphic delays in different game genres (*e.g.,* card game, shooting, adventure, and racing) each with a unique sensitivity. *DECAF* [16] analysed the user-perceived experiences such as visual response received by user across different game genres. The authors highlight that networking issues like round trip delay, limited bandwidth, and packet losses could significantly degraded the user-perceived experience, which is quantitatively measured by our work using gameplay performance metrics. S. Bhuyan *et al.* [17] characterized the user-perceived performance (*e.g.,* frame rendering/decoding and hardware energy consumption) specifically for cloud gaming plays on mobile platforms over wireless networks.

As for network traffic analysis for cloud gaming services, M. Carrascosa *et al.* [21] identified generic traffic statistics (*e.g.,* utilized protocols, distribution of packet size, and packet inter-arrival time) of cloud gaming sessions on Google Stadia platform and their reactions toward sudden changes of network capacity. X. Xu *et al.* [51] measured the change in bandwidth of cloud gaming flows when competing with TCP Cubic and BBR flows. Existing works such as [18], [52] developed NFV/SDN-based traffic processing systems to extract generic network attributes (*e.g.,* bitrate) of UDP flows to detect those carrying downstream video content of cloud games without considering other critical flows. X. Marchal *et al.* [21], [53] studied the network traffic patterns of cloud gaming platforms under constrained (cellular) network conditions. The works described in [19], [20] analyzed performance anomalies (*e.g.,* channel degradation) of cloud gaming sessions served by 4G networks with time-series network KPIs; and the work in [35] analyzed the adaptability of cloud gaming platforms under various levels of active quality setups and different network conditions. In this paper, as revised and extended from our preliminary work [2], our objective is to address the lack of actionable visibility for network operators. Compared to other works that analyze cloud gaming network traffic to investigate performance impact on RTP flows caused by suboptimal network conditions, we characterize network traffic patterns covering all service flows used in all stages of a cloud gaming session, carrying different gameplay functionalities, and across user setups. The comprehensive and unique insights obtained in our work are leveraged for gameplay detection, user setup identification, and measurement of cloud gaming user experiences.

VIII. Conclusion

We developed a network traffic analysis method that provides network operators visibility into cloud gaming sessions over their networks, specifically those served by NVIDIA GeForce NOW platform. We first analyzed network traffic characteristics of cloud game sessions to establish benchmarks for critical service flows that exhibit unique patterns based on user setups and gameplay experiences. We then design a method to detect cloud game sessions across various user setups by stateful matching of service flows, classify critical gameplay flows using volumetric attributes, and track experience metrics (*i.e.,* game lag, frame rate, and graphic resolution) on those flows. The method is deployed in a large university network and a network operator that operates NVIDIA's GeForce NOW cloud gaming service in Australia and New Zealand, evaluated using ground-truth sessions, and demonstrated for its operational insights from real-world cloud game sessions.

Acknowledgments

We thank Sharat Chandra Madanapalli, Sanyam Jain, Maheesha Perera and Craig Russell from Canopus Networks Pty Ltd and Jeremy Hall, Dylan Ryan and Matthew Kocoski from Pentanet Ltd for their hard work in system engineering and infrastructure support at our partnered network operator exclusively hosting NVIDIA's GeForce NOW cloud gaming servers for Australia and New Zealand region.

References

[1] M. Lyu and V. Sivaraman, "A Large-Scale Network Measurement Study of NVIDIA GeForce NOW Cloud Gaming in the Wild," *IEEE/ACM Transactions on Networking*, 2025.

[2] M. Lyu, S. C. Madanapalli, A. Vishwanath, and V. Sivaraman, "Network Anatomy and Real-Time Measurement of Nvidia GeForce NOW Cloud Gaming," in *Proc. PAM*, Virtual Event, Mar. 2024.

[3] Business, "Microsoft, Activision Blizzard and the Future of Gaming," *The Economist*, Nov 2022.

[4] News, "The Future of Video Games," *The Economist*, Mar 2023.

[5] Gitnux, "Cloud Gaming Services: A Look at the Latest Statistics," https://blog.gitnux.com/cloud-gaming-services-statistics/#content, 2023, accessed: 2023-06-26.

[6] S. C. Madanapalli, H. H. Gharakheili, and V. Sivaraman, "Know Thy Lag: In-Network Game Detection And Latency Measurement," in *Proc. PAM*, Apr 2022.

[7] J. Livingood, "Comcast Kicks Off Industry's First Low Latency DOCSIS Field Trials," https://corporate.comcast.com/stories/comcast-kicks-off-industrys-first-low-latency-docsis-field-trials, 2023, accessed: 2023-06-26.

[8] CAMARA, "Quality on Demand," https://camaraproject.org/quality-on-demand/, 2023, accessed: 2024-12-19.

[9] N. Wehner, M. Seufert, J. Schuler, S. Wassermann, P. Casas, and T. Hossfeld, "Improving Web QoE Monitoring for Encrypted Network Traffic through Time Series Modeling," *SIGMETRICS Perform. Eval. Rev.*, May 2021.

[10] B. Spang, B. Walsh, T.-Y. Huang, T. Rusnock, J. Lawrence, and N. McKeown, "Buffer Sizing and Video QoE Measurements at Netflix," in *Proc. Workshop on Buffer Sizing*, Jan 2020.

[11] S. Liu, T. Mangla, T. Shaowang, J. Zhao, J. Paparrizos, S. Krishnan, and N. Feamster, "AMIR: Active Multimodal Interaction Recognition from Video and Network Traffic in Connected Environments," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, Mar 2023.

[12] T. Sharma, T. Mangla, A. Gupta, J. Jiang, and N. Feamster, "Estimating WebRTC Video QoE Metrics Without Using Application Headers," in *Proc. ACM IMC*, Montreal QC, Canada, 2023.

[13] Y. Wang, M. Lyu, and V. Sivaraman, "Characterizing User Platforms for Video Streaming in Broadband Networks," in *Proc. ACM IMC*, Madrid, Spain, Nov 2024.

[14] F. Bronzino, P. Schmitt, S. Ayoubi, G. Martins, R. Teixeira, and N. Feamster, "Inferring Streaming Video Quality from Encrypted Traffic: Practical Models and Deployment Experience," *Proc. ACM Meas. Anal. Comput. Syst.*, Dec 2019.

[15] M. Lyu, R. D. Tripathi, and V. Sivaraman, "MetaVRadar: Measuring Metaverse Virtual Reality Network Activity," *Proc. ACM Meas. Anal. Comput. Syst.*, Dec 2023.

[16] H. Iqbal, A. Khalid, and M. Shahzad, "Dissecting Cloud Gaming Performance with DECAF," *Proc. ACM Meas. Anal. Comput. Syst.*, Dec 2021.

[17] S. Bhuyan, S. Zhao, Z. Ying, M. T. Kandemir, and C. R. Das, "End-to-End Characterization of Game Streaming Applications on Mobile Platforms," *Proc. ACM Meas. Anal. Comput. Syst.*, Feb 2022.

[18] J. R. Ky, P. Graff, B. Mathieu, and T. Cholez, "A Hybrid P4/NFV Architecture for Cloud Gaming Traffic Detection with Unsupervised ML," in *Proc. IEEE Symposium on Computers and Communications*, Los Alamitos, CA, USA, Jul 2023.

[19] J. R. Ky, B. Mathieu, A. Lahmadi, and R. Boutaba, "Assessing Unsupervised Machine Learning solutions for Anomaly Detection in Cloud Gaming Sessions," in *Proc. IEEE CNSM*, Thessaloniki, Greece, Oct 2022.

[20] J. R. Ky *et al.*, "ML Models for Detecting QoE Degradation in Low-Latency Applications: A Cloud-Gaming Case Study," *IEEE Transactions on Network and Service Management*, Sep 2023.

[21] M. Carrascosa and B. Bellalta, "Cloud-gaming: Analysis of Google Stadia Traffic," *Computer Communications*, vol. 188, pp. 99–116, Mar 2022.

[22] Markets and Markets, "Cloud Gaming Market by Offiering, Device Type, Solution, Game Type, Region – Global Forecast to 2024," https://bit.ly/3AyEjio, 2019, accessed: 2024-04-27.

[23] Nvidia, "GeForce NOW," https://www.nvidia.com/en-au/geforce-now/, 2023, accessed: 2023-01-12.

[24] Microsoft, "XBox Cloud Gaming (Beta)," https://www.xbox.com/en-us/play, 2023, accessed: 2023-01-12.

[25] Sony Interactive Entertainment, "PlayStation Now," https://www.playstation.com/en-us/ps-now/, 2023, accessed: 2023-04-18.

[26] Amazon, "Luna," https://www.amazon.com/luna/landing-page, 2023, accessed: 2023-01-12.

[27] Kinsta, "What is a Content Management System (CMS)?" https://kinsta.com/knowledgebase/content-management-system/, 2022, accessed: 2023-01-12.

[28] Nvidia Support, "How Can I Reduce Lag or Improve Streaming Quality When Using GeForce NOW?" bit.ly/45TOmfR, 2022, accessed: 2022-12-12.

[29] Nvidia, "WebRTC Browser Client," bit.ly/3LhOPAj, 2022, accessed: 2022-12-14.

[30] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications," RFC 3550, Jul 2003. [Online]. Available: https://datatracker.ietf.org/doc/html/rfc3550

[31] CloudFlare, "What is SNI? How TLS Server Name Indication Works," https://www.cloudflare.com/en-gb/learning/ssl/what-is-sni/, 2022, accessed: 2023-01-12.

[32] S. Roy, T. Shapira, and Y. Shavitt, "Fast and Lean Encrypted Internet Traffic Classification," *Computer Communications*, vol. 186, pp. 166–173, 2022.

[33] R. J. Babaria, M. Lyu, G. Batista, and V. Sivaraman, "FastFlow: Early Yet Robust Network Flow Classification using the Minimal Number of Time-Series Packets," *Proc. ACM Meas. Anal. Comput. Syst.*, Jun 2025.

[34] I. Akbari *et al.*, "A Look Behind the Curtain: Traffic Classification in an Increasingly Encrypted Web," *Proc. ACM Meas. Anal. Comput. Syst.*, Feb 2021.

[35] M. Lyu, Y. Wang, and V. Sivaraman, "Do Cloud Games Adapt to Client Settings and Network Conditions?" in *Proc. IFIP/IEEE Networking Conference*, Thessaloniki, Greece, Jun. 2024.

[36] S. Landau-Feibish, Z. Liu, and J. Rexford, "Compact data structures for network telemetry," *ACM Comput. Surv.*, Mar 2025.

[37] C. Estan, G. Varghese, and M. Fisk, "Bitmap Algorithms for Counting Active Flows on High-Speed Links," *IEEE/ACM Transactions on Networking*, vol. 14, no. 5, 2006.

[38] Z. Zhu, Y. Zhao, and Z. Liu, "In-memory key-value store live migration with NetMigrate," in *Proc. USENIX FAST*, Santa Clara, CA, USA, 2024.

[39] W. Cai, R. Shea, C.-Y. Huang, K.-T. Chen, J. Liu, V. C. M. Leung, and C.-H. Hsu, "A Survey on Cloud Gaming: Future of Computer Games," *IEEE Access*, vol. 4, pp. 7605–7620, 2016.

[40] R. Shea, J. Liu, E. C.-H. Ngai, and Y. Cui, "Cloud Gaming: Architecture and Performance," *IEEE Network*, vol. 27, no. 4, pp. 16–21, 2013.

[41] H. Chen, X. Zhang, Y. Xu, J. Ren, J. Fan, Z. Ma, and W. Zhang, "T-Gaming: A Cost-Efficient Cloud Gaming System at Scale," *IEEE Transactions on Parallel and Distributed Systems*, vol. 30, no. 12, pp. 2849–2865, 2019.

[42] Y. Han, D. Guo, W. Cai, X. Wang, and V. C. M. Leung, "Virtual Machine Placement Optimization in Mobile Cloud Gaming Through QoE-Oriented Resource Competition," *IEEE Transactions on Cloud Computing*, vol. 10, no. 3, pp. 2204–2218, 2022.

[43] Y. Li, C. Shan, R. Chen, X. Tang, W. Cai, S. Tang, X. Liu, G. Wang, X. Gong, and Y. Zhang, "GAugur: Quantifying Performance Interference of Colocated Games for Improving Resource Utilization in Cloud Gaming," in *Proc. ACM HPDC*, Phoenix, AZ, USA, Jun 2019.

[44] X. Zhang, H. Chen, Y. Zhao, Z. Ma, Y. Xu, H. Huang, H. Yin, and D. O. Wu, "Improving Cloud Gaming Experience through Mobile Edge Computing," *IEEE Wireless Communications*, vol. 26, no. 4, pp. 178–183, 2019.

[45] M. Ghobaei-Arani, R. Khorsand, and M. Ramezanpour, "An Autonomous Resource Provisioning Framework for Massively Multiplayer Online Games in Cloud Environment," *Journal of Network and Computer Applications*, vol. 142, pp. 76–97, 2019.

[46] G. K. Illahi, T. V. Gemert, M. Siekkinen, E. Masala, A. Oulasvirta, and A. Ylä-Jääski, "Cloud Gaming with Foveated Video Encoding," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 16, no. 1, Feb 2020.

[47] I. Slivar, M. Suznjevic, and L. Skorin-Kapov, "Game Categorization for Deriving QoE-Driven Video Encoding Configuration Strategies for Cloud Gaming," *ACM Trans. Multimedia Comput. Commun. Appl.*, Jun 2018.

[48] K.-T. Chen, Y.-C. Chang, H.-J. Hsu, D.-Y. Chen, C.-Y. Huang, and C.-H. Hsu, "On the Quality of Service of Cloud Gaming Systems," *IEEE Transactions on Multimedia*, vol. 16, no. 2, pp. 480–495, 2014.

[49] T. Kämäräinen, M. Siekkinen, A. Ylä-Jääski, W. Zhang, and P. Hui, "A Measurement Study on Achieving Imperceptible Latency in Mobile Cloud Gaming," in *Proc. ACM MMSys*, Taipei, Taiwan, Jun 2017.

[50] S. S. Sabet, S. Schmidt, S. Zadtootaghaj, C. Griwodz, and S. Möller, "Delay Sensitivity Classification of Cloud Gaming Content," in *Proc. ACM International Workshop on Immersive Mixed and Virtual Environment Systems*, Istanbul, Turkey, Jun 2020.

[51] X. Xu and M. Claypool, "Measurement of Cloud-Based Game Streaming System Response to Competing TCP Cubic or TCP BBR Flows," in *Proc. ACM IMC*, Nice, France, 2022.

[52] P. Graff, X. Marchal, T. Cholez, B. Mathieu, and O. Festor, "Efficient Identification of Cloud Gaming Traffic at the Edge," in *Proc. IEEE/IFIP Network Operations and Management Symposium*, May 2023.

[53] G. P. K. J. R. Marchal, Xavierl, T. Cholez, S. Tuffin, B. Mathieu, and O. Festor, "An Analysis of Cloud Gaming Platforms Behaviour Under Synthetic Network Constraints and Real Cellular Networks Conditions," *Journal of Network and Systems Management*, Feb 2023.