A Network Analysis on Cloud Gaming: Stadia, GeForce Now and PSNow

Andrea Di Domenico, Gianluca Perna,
Martino Trevisan, Luca Vassio, Danilo Giordano
Politecnico di Torino first.last.polito.it

Abstract

Cloud gaming is a class of services that promises to revolutionize the videogame market. It allows the user to play a videogame with basic equipment while using a remote server for the actual execution. The multimedia content is streamed through the network from the server to the user. Hence, this service requires low latency and a large bandwidth to work properly with low response time and high-definition video. Three among the leading tech companies, (Google, Sony and NVIDIA) entered this market with their own products, and others, like Microsoft and Amazon, are planning to launch their own platforms in the near future. However, these companies released so far little information about their cloud gaming operation and how they utilize the network. In this work, we study cloud gaming services from the network point of view. We collect more than 200 packet traces under different application settings and network conditions from a broadband network up to poor mobile network conditions, for 3 cloud gaming services, namely Stadia from Google, GeForce Now from NVIDIA and PS Now from Sony. We analyze the employed protocols and the workload they impose on the network. We find that GeForce Now and Stadia use the RTP protocol to stream the multimedia content, with the latter relying on the standard WebRTC APIs. They result bandwidth-hungry and consume up to 45 Mbit/s, depending on the network and video quality. PS Now instead uses only undocumented protocols and never exceeds 13 Mbit/s. 4G mobile networks can often sustain these loads, while traditional 3G connections struggle. The systems quickly react to deteriorated network conditions, and packet losses up to 5% do not cause a reduction in resolution.

1 Introduction

Year by year new services are born as the Internet connection becomes faster and more reliable for end users [1]. The HTTP protocol has been the foundation of the World Wide Web, allowing the transfer of static content through the network. In the late 1990s and early 2000s, other protocols has become popular while supporting a broader spectrum of services. Notable examples are P2P for sharing files and Voice-over-IP (VoIP) for phone calls, made popular by Skype. In the 2010s, social networks and high-definition video streaming have become the new heavy-hitters in terms of involved users and traffic volume, making the Internet a pillar of our working and leisure activities. Such heterogeneous classes of services have different requirements from the network point of view. For example, VoIP requires low latency, while video streaming needs large bandwidth. Their popularity shows that the Internet has succeeded in providing a reliable means for these applications to run and in satisfying the users' expectations.

During the last couple of years, some of the largest tech companies like Google and Sony announced new platforms for the so-called *cloud gaming*. In few words, the user plays a videogame using her equipment, while the actual execution takes place on the cloud and the multimedia content is streamed through the network from the server to the user. This implies that the user terminal no longer needs to be equipped with powerful hardware (like GPUs) to play with recent demanding games, but only requires a fast and reliable communication with the cloud. This new platforms promise to revolutionize the videogame industry, which has steadily increased in the last decades from basically a not existing market in the 1970s to an estimated USD 159.3 billion sales in 2020 and 2.7 billion players worldwide [2]. Videogame is set to become the most lucrative sector in the whole entertainment market by the end of 2020, surpassing television. Given these facts, the success of cloud gaming service would be of huge importance for the companies promoting it and for the Internet network handling the resulting traffic.

Google, NVIDIA and Sony announced three different proprietary platforms for cloud gaming, namely, Stadia, GeForce Now and PS Now. At the time of writing of this paper (January 2021), these services are available to users as a full product. Microsoft also proposed its platform named xCloud, which is currently under testing. Amazon also announced Luna, currently under testing as well. The companies disclosed limited amount of information about the infrastructure of such services and on how they utilize the network in terms of employed protocols and bandwidth usage.

In this work, we target the three aforementioned cloud gaming services and offer a preliminary characterization of the network workload they generate. We collect 225 packet traces under different application settings and network conditions, and we show which protocols they adopt for streaming multimedia content, the volume of traffic they generate and the servers and domains they contact.[1] We show that Stadia and GeForce Now adopt the standard RTP protocol for streaming [3], while PS Now uses an undocumented protocol or encapsulation mechanism. Stadia and GeForce Now transmit video up to 45 Mbit/s, while PS Now does not exceed 13 Mbit/s. For comparison, a Netflix 4K video consumes approximately 15 Mbit/s,[2] making cloud gaming potentially a new heavy hitter of the future Internet. With these characteristics, mobile networks can sustain high-definition gaming sessions only in case of good-quality $4G$ connections, as we quantify using real-world speedtest measurements. Their client applications contact cloud servers located in the autonomous systems (ASes) of the respective corporations. They are no farther than 20 ms in terms of Round Trip Time from our testing location in northern Italy, a figure low enough for playing a videogame with no noticeable lag. To the best of our knowledge, we are the first to perform a study on these cloud gaming services, namely Stadia, GeForce Now and PS Now, showing their characteristics and peculiarities in terms of network usage.

The remainder of the paper is organized as follows. Section 2 presents the related work. Section 3 describes our experimental setup while Section 2 illustrates the findings we obtain analyzing the packet traces. Finally, Section 5 concludes the paper.

2 Related work

Several research papers already studied cloud gaming under different perspectives [4]. Authors in [5] focused on the study of the cloud gaming platforms and optimization techniques, while several works provided general frameworks and guidelines for deploying cloud gaming services from the technical [6, 7] and business [8] points of view. Other studies focused on the factors affecting the subjective QoE of users [9, 10] or proposed novel techniques to improve it [11, 12]. Some works proposed approaches for optimizing multimedia streaming [13, 14, 15] or GPU usage [16] in the specific context of cloud gaming. Interestingly, in 2012 the authors of [17] concluded that the network infrastructure was unable to meet the strict latency requirements necessary for acceptable gameplay for many end-users.

Industrial pioneers of cloud gaming such as Onlive, StreamMyGame and Gaikai have already been object of study [18, 19, 20, 21]. However, research papers targeting the services recently launched by leading tech companies such as Google and Sony have not yet been published. Only Stadia has been studied by Carrascosa *et al.* [22], who investigated its network utilization; however, their work did not compare with other platforms nor explore mobile scenarios. In this paper, we close this gap and provide a first characterization of the generated traffic in terms of employed protocols and network workload.

3 Measurement collection

In this section, we describe our experimental testbed and the dataset we collect. We focus on the three aforementioned cloud gaming services, namely Stadia, GeForce Now and PS Now, on which we created standard accounts that we use for our experiments. We deploy a testbed using three gaming devices: A PC running Windows with a 4K screen, an Android smartphone, and the dedicated Stadia dongle.

All devices are connected to the Internet through a Linux gateway equipped with two 1 Gbit/s Ethernet network interfaces and a 300 Mbit/s WiFi 802.11ac wireless card. The Windows PC is connected to the gateway on the first Ethernet card, while the Android smartphone and the Stadia dongle are connected via WiFi. The gateway uses the second Ethernet interface as upstream link to the Internet, which is provided by a 1 Gbit/s Fiber-To-The-Home subscription located in Turin, Italy. Figure 1 sketches our testbed.

Stadia runs via the Chrome browser on the Windows PC and with its own mobile application on the Android phone (we used version 2.13). Moreover, we perform additional experiments by using the dedicated Chromecast Ultra dongle which allows the user to play connecting it to a screen. GeForce Now runs from a specific application in both cases (version 1.0.9 for PC and 5.27.28247374 for Android), while PS Now only works from the PC application (version 11.0.2 was used).

We play the three services making gaming sessions approximately 5-10 minutes long and capturing all the

[1] For brevity, we use throughout the paper the term *domain* referring to the Fully Qualified Domain Name.

[2] `https://help.netflix.com/en/node/87`, accessed January 2021.

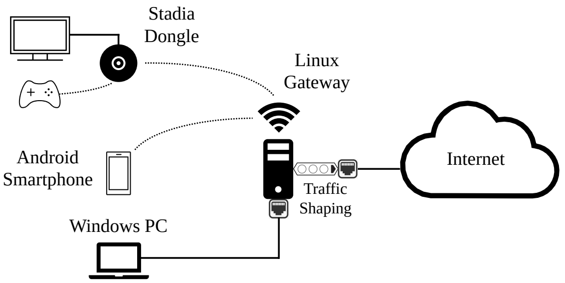


Figure 1: Testbed used for the experimental campaigns.

traffic that the devices exchange with the Internet. We seek for reliable results by playing a wide spectrum of videogames on all platforms, from first-person shooters to racing and adventure – e.g., Sniper Elite 4, Destiny, Grid and Tomb Rider.

With this testbed, we perform five different experimental campaigns, summarized in Table 1. Firstly, we run different gaming sessions for each platform using the Windows PC, the smartphone and the dongle (when possible). Secondly, we run different gaming sessions by using the Windows PC and by manually configuring the applications to stream video with different *quality levels*. This option is available on Stadia and GeForce Now. Thirdly, only for GeForce Now, we instrument the Windows application to use one of the 14 available data centers by tuning the *server location* application setting. Next, we artificially impair the network in terms of *bandwidth* and *latency* and *packet loss* to study the behavior of the applications under different network conditions. To this end, we run the `tc-netem` tool on the Linux gateway to progressively decrease the available bandwidth from 100 to 5 Mbit/s, impose additional latency from 10 to 300 ms, or 1-10% packet loss. For Stadia and GeForce Now, we replicate all the experiments using both the PC and the smartphone. Moreover, we perform all experiments also with the Stadia dongle. Finally, we take Stadia as a case study to understand the behavior with different mobiles networks. To this end, we perform different gaming session with the PC and emulating on the Linux gateway different mobile networks using ERRANT [23]. ERRANT is a state-of-the-art network emulator which imposes realistic network conditions based on a large-scale measurement campaign under operational mobile networks. ERRANT can reproduce the variability of conditions intrinsically rooted in mobile networks due to different operators, RATs (i.e., 3G or 4G), signal quality (e.g., bad quality due to weak signal). ERRANT comes with 32 network profiles describing the typical network conditions observed in different European operators, under 3G and 4G. We also use the ERRANT speedtest training dataset to study the possibility of using Stadia on different conditions under mobile networks.

In total, we collect 225 packet traces, summing to 390 GB of data. We share with the research community a sample of these traces from the three services.[3]

Then, we analyze the traffic traces using the Wireshark packet analyzer.[4] We also use Tstat [25], a passive meter, to obtain flow-level logs summarizing the observed TCP/UDP flows. Finally, we use the Chrome debugging console to study Stadia and the disk log files for GeForce Now.

4 Results

We now illustrate the findings we obtain from the analysis of the collected network traces. We first show which network protocols each service employs for streaming and signaling (e.g., user's commands) and analyze in details the different approaches used for audio and video transmission. We then provide quantitative figures on the volume of traffic the services generate at different video quality levels and study the impact of mobile network scenarios. Finally, we study the contacted servers in terms of Autonomous Systems (ASs), RTT distance, and discuss how the infrastructures are organized.

4.1 Employed protocols

In this section, we describe the protocols that are used by the three cloud gaming providers to stream multimedia content and transmit control data for, e.g., session setup and users' commands. Table 2 provides an overview of the protocols we observe, as well as the employed codecs.

Stadia: The service from Google uses the most standard protocol mix as it relies on WebRTC [26]. In few words, WebRTC is a set of standard application programming interfaces (APIs) that allow real-time communication from browsers and mobile applications. It establishes sessions using the DTLS protocol for key exchange. The multimedia connection between client and server is set up using ICE, which in turn relies on the STUN and TURN protocols for NAT traversal. We find that Stadia uses WebRTC with no substantial modifications, both from the browser and mobile application. The traffic captures using the dedicated dongle device (Chromecast) confirm that the observed traffic is consistently compatible with WebRTC. When the multimedia session begins, the client starts a normal WebRTC *peer connection* towards the server, creating a UDP flow in

[3] Sample packet traces available at [24].

[4] `https://www.wireshark.org/`.

Table 1: Overview of the measurement campaign.

Application	PC	Smartphone	Dongle	Quality levels	Server location	Traffic shaping	Mobile Networks	Total tests
Stadia	✓	✓	✓	3		✓	✓	94
Geforce Now	✓	✓		2	✓	✓		71
PS Now	✓					✓		60

which DTLS, STUN and RTP are multiplexed according to the RFC 7893 [27]. RTP is used for multimedia streaming, while DTLS carries the user's input.

GeForce Now: It adopts a different approach. The server is first contacted using the TLS (over TCP) protocol to set up the session. Interestingly, the Client Hello messages contain the Server Name Indication extension, which allows us to infer the server hostname (see Section 4.6 for details). Then, the client opens multiple UDP channels directly, without relying on the common session establishment protocols (ICE, STUN and TURN). Only the first packet from the client contains an undocumented hello message. Each inbound flow then carries a normal RTP stream. The client sends the user commands on a dedicated UDP flow using an undocumented protocol. All flows use fixed ports on the client-side, in the range 49003-49006, while they vary on the server-side.

PS Now: This service adopts a completely custom approach, with no standard in-clear header. The client opens a UDP connection towards the server, without relying on any documented protocol, and, as such, we can only analyze the raw payload of packets. Still, a complex manual work allowed us to catch at least the high-level encapsulation schema, that we briefly describe. The first byte of the packet is used to multiplex multiple data channels. The channel 0 is used for signaling and user's commands, while 2 and 3 are used to multimedia streaming from the server. This is confirmed by the plausible packet size and inter-arrival distributions and allows us to infer which kind of multimedia content is carried on each channel, as we illustrate in Section 4.3.

Take away: *Stadia relies on WebRTC for streaming and session setup. GeForce Now uses RTP with no standard session establishment protocol. PS Now employs a fully-custom approach.*

4.2 Network testing

All three services have built-in functionalities to probe the network between the client and (multiple) gaming server machines to determine if the conditions are sufficient for a stable gaming session. In few words, the client applications perform a speed test-like measurement to estimate the network delay and bandwidth. We notice that the network testing is not performed consistently on each session startup, but the applications tend to re-probe the network only after a variation of the client IP address.

Stadia performs a speed test based on RTP packets carried over a session established using the standard WebRTC APIs like it does for the multimedia streams. The server (not necessarily the same used for the subsequent gaming session) sends to the client 5-6 MB of data, resulting in a UDP session 5-60 seconds long, depending on the network conditions. The RTP packets are large-sized, around 1200 bytes on average, but we cannot inspect their payload since it is encrypted.

GeForce Now uses an schema similar to the one used in the popular tool Iperf[5], in which the client sets up a network test over a UDP channel on the server port 5001 (the same port used by Iperf). The first few packets carry JSON-encoded structures to set up the test. In case the test includes a latency measure, the last flow packets indicate the measured RTT samples. In case the test is only for bandwidth, we observe a stream of large-sized UDP packets lasting 5-10 seconds. Again, the testing server is different from the one used for the subsequent gaming session. The inspection of the JSON messages allows us to understand that the client probes the latency towards multiple alternative measurement servers.

PS Now adopts again a fully-custom approach. At the beginning of each gaming session, the client performs a few-seconds long bandwidth test using a custom or fully-encrypted protocol running over UDP. We cannot infer any information from the packets, for which we only observe that they all have size 1468 bytes. The test is performed towards a server different from the one used for the proper gaming streaming session. We note similar additional streams consisting of few packets toward a handful of other servers, that we conjecture are used to probe the latency towards more endpoints.

[5] `https://iperf.fr/`, accessed January 2021.

Privacy concerns: While analyzing the GeForce Now network testing mechanism, we notice that the client-side control packets used to set up the test expose the user to a severe privacy concern. The user ID is sent in clear into the UDP packet, allowing an eavesdropper to uniquely identify a user even if she changes her IP address or is roaming on another network. We compared the user ID to the user account number that we obtain on the profile management page on the NVIDIA website and they match, confirming that the identifier is uniquely associated with the account. Following the best practices for these cases, we signaled the issue to NVIDIA before making our paper public, which plans to resolve it on one of the next updates.

Take away: *Stadia uses an RTP stream for testing the network, while GeForce Now relies on an Iperf-like mechanism. Again, PS Now employs a fully-custom approach. We found a severe privacy leakage in the current GeForce Now implementation that allows an eavesdropper to obtain the user identifier.*

4.3 Multimedia streaming

We now analyze in details how the three services stream the multimedia content (audio and video) from the gaming server to the client. In the case of Stadia and GeForce Now, we will provide figures extrapolated inspecting the RTP headers, while for PS Now we can separate the different streams by looking at the first byte of the UDP payload as mentioned in the previous section. In the last row of Table 2 we report the employed video codecs as we extract from the browser/application log files. The widespread H.264 codec is used by both GeForce Now and Stadia, which can also employ the newer VP9 if the device supports it. For PS Now, we could not obtain any information about the codecs.

Stadia relies on the WebRTC APIs, and, as such, the multimedia streaming follows its general principles. A single UDP flow carries multiple RTP streams identified by different Source Stream Identifiers (SSRC). A stream is dedicated to the video, while another one to the audio track. We also find a third flow used for video retransmission, as we confirm using the Chrome debug console.[6] During most of the session, it is almost inactive. At certain moments the flow becomes suddenly active, carrying large packets containing video content. Moreover, this behavior co-occurs with packet losses and bitrate adaptations on the video stream, as we expect for a video retransmission feature.

GeForce Now again relies on RTP for multimedia streaming, as described in the previous section. Differently from Stadia, it uses separate UDP flows for the different multimedia tracks, whose client-side port numbers can be used to distinguish the content as NVIDIA publicly declares.[7] On port 49005, a UDP flow carries a single RTP stream for the inbound video. The audio is contained in a UDP flow on port 49003, in which we find two RTP streams active at the same time.

Regarding PS Now, we cannot find any header belonging to publicly documented protocol. However, the inspection of several packet captures allows us to infer the encapsulation schema used by the application. A single UDP flow carries all multimedia streams. To multiplex the streams, the first byte of UDP payload indicates the channel number, followed by a 16-bit long sequence number. Channel 2 carries the video stream, as we can conclude looking at the packet size and inter-arrival time of packets. Channel 3 carries the audio track as the packets are small and fixed-sized (250 B) and arrive at a constant pace of one every 20 ms. We also find channel 0, especially at the beginning of the flow, which we conjecture is used for signaling. Finally, channel 12 seldom becomes active, especially in correspondence of large packet losses. As such, we conjecture that it is used for video retransmission or some form of forward error correction (FEC), similarly to the Stadia approach.

Take away: *For Stadia, a single UDP flow carries separate streams for audio, video and video retransmissions. GeForce Now uses multiple UDP flows on different client-side ports. In PS Now, a single UDP flow appears to carry an audio, a video and a retransmission/FEC stream.*

4.4 Network workload

We now focus on the workload imposed on the network by users playing on cloud gaming services. We start our analysis with Figure 2, in which we show the evolution of a gaming session of around 10 minutes for each service. We made the corresponding packet traces available to the community.[3] The picture reports the bitrate of the inbound traffic, due almost exclusively to the video multimedia stream. We first notice that Stadia (Figure 3a) has a constant bitrate, while for GeForce Now and PS Now (Figures 3b and Figure 3c respectively) it is considerably more variable. Especially for GeForce Now, we can root this in the different use of the video codecs, as we describe later in this section. Indeed, the application logs confirm that the variations in the bitrate are not caused by resolution adjustments or codec substitution. Looking at Stadia, it is clear the role of the video retransmission stream (green dashed line), which

[6] The associated RTP stream is found to have mimeType `video/rtx`.

[7] `https://nvidia.custhelp.com/app/answers/detail/a_id/4504/~/how-can-i-reduce-lag-or-improve-streaming-quality-when-using-geforce-now`, accessed January 2021.

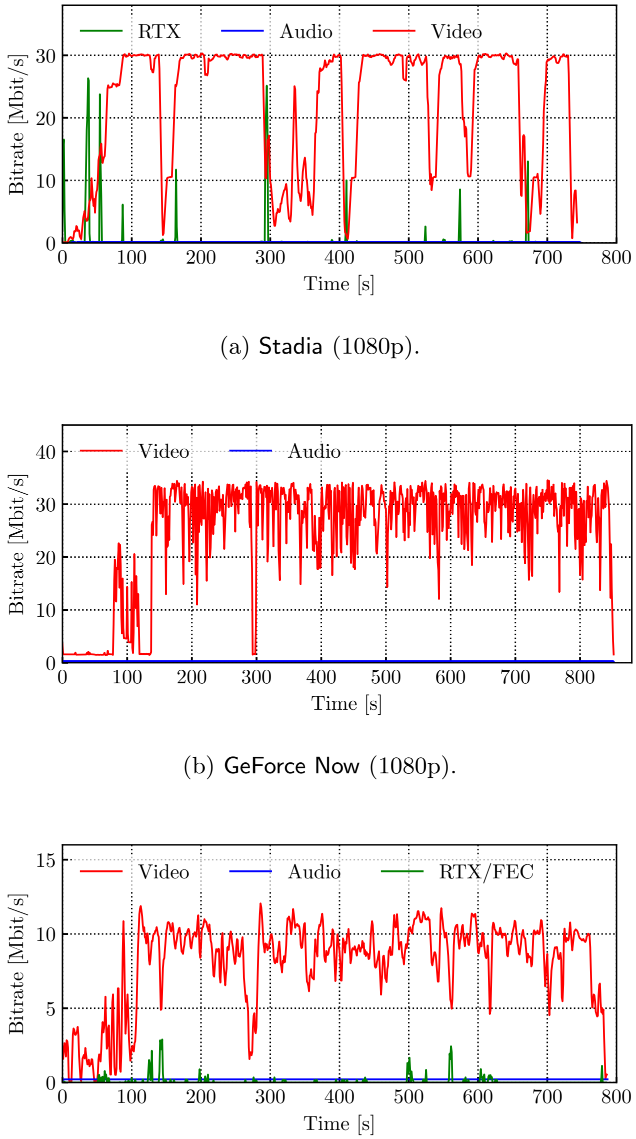
Table 2: Protocol usage for different gaming session components.

	Stadia	GeForce Now	PS Now
Streaming	RTP	RTP	Custom (UDP)
Player's input	DTLS	Custom (UDP)	Custom (UDP)
Session setup	DTLS, STUN	TLS	Custom (UDP)
Network Testing	RTP	Iperf-like	Custom (UDP)
Video Codec	H.264, VP9	H.264	-

becomes active in correspondence of impairments in the main video stream (red solid line). We notice a very similar behavior in PS Now, which allows us to conjecture the presence of an analogous mechanism.

We summarize the network workload showing in Figure 3 the empirical cumulative distribution function of the video bitrate that we observe. For Stadia and GeForce Now, we report separate distributions for different video resolution thanks to Chrome debug console and application logs, respectively. For PS Now, we only report the overall distribution as we cannot extract any statistics from the client app. As mentioned before, Stadia exhibits a rather constant bitrate (Figure 3a). The service allows three streaming resolutions, namely 720p, 1080p and 4K (2160p), whose average bitrate is 11, 29 and 44 Mbit/s respectively. This is consistent with what is declared in documentation.[8] Stadia employs both H.264 and VP9 video codecs, with 4K streaming using uniquely VP9.

Different is the picture for GeForce Now, shown in Figure 3b. The service allows several video resolutions, both in 16:9 and 16:10 aspect ratio. Here, we report the bitrate of the lowest (720p) and the highest (1080p) resolutions available for the 16:9 aspect ratio. Moreover, we show the 540p resolution which is only adopted automatically in case of bad network conditions, that we trigger imposing a bandwidth limit on the testing machine. The figure shows that the bitrate has large variability, especially for 720p and 1080p. In median, 720p (1080p) consumes 15 (20) Mbit/s, which is consistent to what is declared on the system requirements of the service.[9] However, the interquartile ranges (IQRs) are in the order of 15 Mbit/s, much more than the 2-3 Mbit/s IQRs observed Stadia. The bitrate reaches peaks of more than 30 and 40 Mbit/s for 720p and 1080p, respectively. Without access to the unencrypted raw video stream, we conjecture that GeForce Now makes a highly-dynamic use of the H.264 compression parameters to adapt to different video characteristics (static/dynamic scenes) and network conditions. Indeed, our experiments with limited bandwidth show that, for example, GeForce Now



(a) Stadia (1080p).

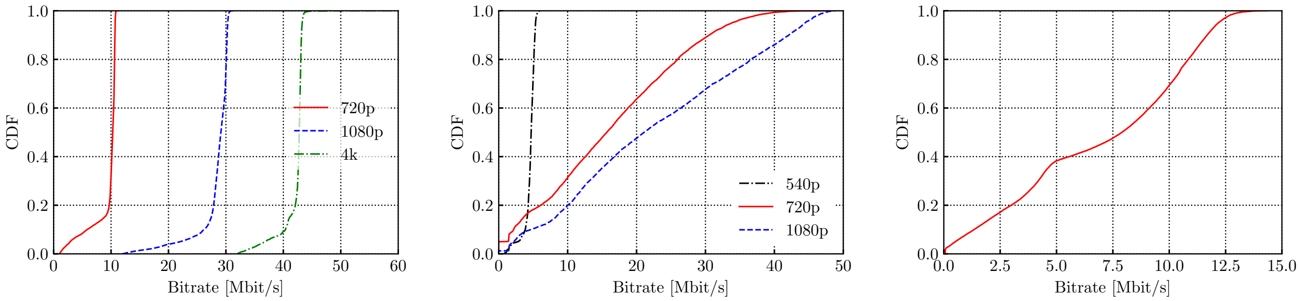
(b) GeForce Now (1080p).

(c) PS Now.

Figure 2: Examples of temporal evolution of bitrate for three gaming sessions.

[8] https://support.google.com/stadia/answer/9607891, accessed January 2021.

[9] https://www.nvidia.com/it-it/geforce-now/system-reqs/, accessed January 2021.



(a) Stadia.

(b) GeForce Now.

(c) PS Now

Figure 3: Cumulative distribution of the video bitrate, for different quality video levels.

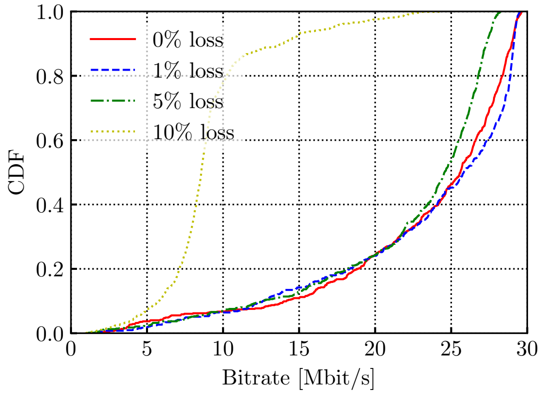


Figure 4: Bitrate distribution with different artificial packet loss.

can sustain a 1080p video stream also with less than 15 Mbit/s available bandwidth without dropping the frame rate, likely adjusting the H.264 compression parameters.

Finally, Figure 3c shows the bitrate distribution for PS Now. Given the lack of application settings or debug information, we only show the overall distribution of the bitrate. The online documentation recommends a minimum bandwidth of 5 Mbit/s and states that video streaming has a 720p resolution. However, we observe the bitrate reaching up to 13 Mbit/s, with a consistent variability. Interestingly, when we impose a 10 Mbit/s or lower bandwidth limitation on the network, the bitrate adapts consequently (see the peak in the distribution at 5 Mbit/s). However, we cannot link it with a resolution lower than 720p.

We now study the impact of packet losses on cloud gaming, focusing on Stadia as a case study. As described in Section 3, we run experiments in which we enforce artificial packet loss via the `tc-netem` tool, ranging from 1 to 10. We then study how the application reacts to these scenarios, focusing on the sending video bitrate and quality level. In Figure 4, we show the distribution of bitrate for experiments with 1, 5 and 10% packet loss, and report the bitrate of sessions with no packet loss as a reference (red solid line). We notice that a packet loss of 1 and 5% (blue and green dashed lines, respectively) does not cause significant bitrate variations. Indeed, the bitrate reaches 30 Mbit/s and we notice the game is running at 1080p most of the time, with some 720p periods. We notice a constant activity on the *retransmission* stream, used to recover lost packets. Different is the case with 10% packet loss (yellow dashed line). In this case, Stadia keeps the video resolution at 720p most of the time, and, as such, the bitrate is limited to 10 Mbit/s. We notice again a high activity on the *retransmission* stream, which, however, is not deemed sufficient to go for higher video resolutions. In a few cases, 1080p is achieved, but for short 5-10 second periods. In all the the cases, the games were still playable and enjoyable, as reported by the volunteers.

Take away: *Stadia has three video resolutions with a rather constant bitrate of (median of 11, 29 and 44 Mbit/s). Packet losses up to 5% do not cause a reduction in resolution. GeForce Now streams video at many different resolutions, each having a large variability in the bitrate, reaching up to 45 Mbit/s. PS Now does not exceed 13 Mbit/s and we cannot infer the employed quality levels.*

4.5 Cloud Gaming Under Mobile Networks

We now investigate the case of mobile networks, with the goal of understanding to what extent cloud gaming is feasible and what is the reached quality level. Moreover, we are interested in understanding the impact of variable network conditions, and how applications react to network impairments. Here, we focus on Stadia,

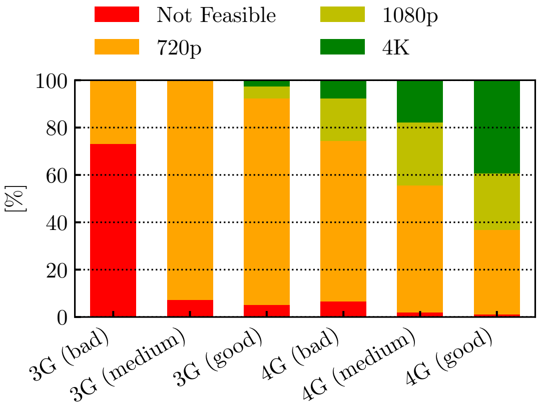


Figure 5: Estimated resolution of Stadia sessions on mobile networks.

which stream video content at fixed bitrates, requiring strict bandwidth constraints.

We first build on a large-scale measurement campaign run on our previous work [23], with the goal of studying to what extent current mobile networks are ready to sustain the load and offer suitable conditions for cloud gaming. The dataset we use includes more than 100k speed test measurements from 20 nodes/locations equipped with SIM cards of 4 operators, located in 2 distinct countries (Norway and Sweden). Each experiment measured latency, downlink, and uplink capacity using multiple TCP connections towards a testing server located in the same country. Moreover, the dataset indicates physical-layer properties such as Radio Access Technology (3G or 4G) and signal strength. The measurement campaign spans the entire 2018, and includes experiments on different hours of the day and days of the week. We use the dataset to understand how a Stadia cloud gaming session would have performed with the measured network characteristics. This is possible thanks to the steady network usage of Stadia, where different video resolutions utilize almost fixed bandwidth. As such, given the network conditions measured on the speedtest measurements, we consider running Stadia *feasible* if there is at least 10 Mbit/s available bandwidth. In case available bandwidth is in $[10-30)$ Mbit/s, we conclude that a user could only reach a video resolution of 720p. When bandwidth is in the range $[30 - 44)$ Mbit/s, a better 1080p video could be transmitted, while more than 44 Mbit/s allow the user to properly receive a $4K$ quality video.

Figure 5 shows the reached video quality levels, offering a breakdown on 3G and 4G, and on different signal qualities as measured by the node radio equipment.[10]

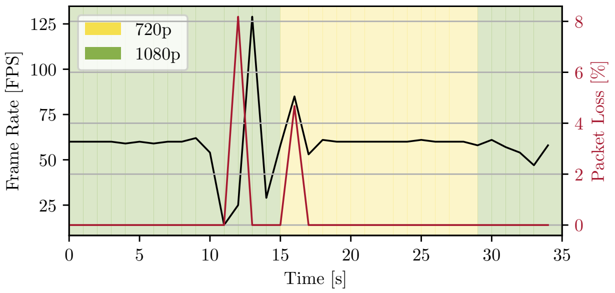


Figure 6: Example of quality-level reduction in Stadia in terms of frame rate and video resolution subsequent to packet losses.

The figure shows how a poor 3G link does not allow using Stadia in most cases, as the available bandwidth is below 10 Mbit/s. The picture changes with medium signal quality, where using Stadia is possible most of the time, with the lowest 720p resolution. An excellent 3G connection allows higher resolution in less than 10% of cases. With 4G, the network generally offers higher bandwidth and 1080p and 4K are possible, especially when the signal quality is good. Indeed, 4G links with high signal strength can sustain Stadia 4K streaming 40% of the cases, and only 38% of the times the client is limited to 720p.

Next, we study the capability of the gaming platforms to offer a good gaming session and cope with different mobile network conditions. To this end, we use the PC to run a gaming session and track different quality metrics about: the frame rate, expressed in Frames Per Second (FPS), the packet loss and the video resolution quality experienced during the gaming sessions. We focus on Stadia as a case study since it is available on a wide range of devices.

To study the gaming session in a controlled mobile network environment we emulate, on the Linux gateway, different mobile network conditions by using ERRANT. In details, we perform experiments by using a 4G good network profile as, currently, it is the best quality and most widespread network, and with a 3G good network profile as it is the lowest quality profile showing enough bandwidth to support a gaming session. To reproduce the mobile network variability, we set ERRANT to re-sample the network condition every 10 seconds, i.e., we pick and apply new constraints for the download rate, upload rate, and latency from the selected profile.

Figure 6 reports an example of the frame rate (right x-axis), the packet loss (left x-axis) and the video resolution quality when a sudden change happened during a gaming session while using the 4G good profile. Interest-

[10] The measured Received signal strength indication (RSSI) is mapped to quality levels as recommended at `https://wiki.teltonika.lt/view/Mobile_Signal_Strength_Recommendations`.

Table 3: Gaming servers characterization.

	Servers	Subnets	ASNs	Owner
Stadia	74	22	15169	Google
GeForce Now	37	23	11414, 20347, 50889	NVIDIA
PS Now	36	2	33353	Sony

ingly, for most of the gaming session we experienced stable performance with no packet loss, 60 FPS, and 1080p as resolution. Only when ERRANT picks a download bandwidth below 10 Mbit/s, we experience, for a short time, a reduction in frame rate and resolution down to 20 FPS and 720p, respectively, with a seldom increase of the packet loss. Interestingly, the Stadia platform is able to quickly react and adapt itself by reaching a frame rate of 60 FPS and resolution at 720p, rising again to 1080p as soon as a better bandwidth is available. With a 3G good network profile, instead, we could not run a full gaming session as the network variability introduced by the mobile network caused the game to suddenly stop. This shows how the promising benefits of these solutions currently have limitations that could be overcome with the increasing popularity of fast mobile network technologies – 4G and, in particular, 5G.

Take away: *Stadia cannot run in most of cases with a bad 3G network. With proper signal strength, 3G allows a resolution of 720p. With 4G, it is possible to achieve higher quality levels, and good signal strength allows 4K video resolution in 40% of situations. Controlled experiments reveal how Stadia quickly reacts to deteriorated network conditions.*

4.6 Location of gaming machines

In this section, we provide a preliminary investigation on the cloud gaming infrastructure, in terms of number and location of game servers and employed domains, as observed from our location. This can be useful to identify cloud gaming traffic for, e.g., traffic engineering or content filtering.

We first focus on the remote gaming machines, analyzing the server IP addresses the client applications contact, summarized in Table 3. Indeed, after the initial setup phase, the client exchanges traffic (almost) uniquely with a single server where the gaming is likely executed.[11] Considering Stadia, at each session we contacted a different server – i.e., we performed 74 sessions and reached 74 different server IPs. The lay on 22 subnets /24, all belonging to the Google 15129 AS.[12] Different is the case for GeForce Now and PS Now, for which in roughly 50% of the cases we contacted a server IP we had already observed. GeForce Now servers lay on 23 subnets belonging to three different ASes, all controlled by NVIDIA. Remind that in our experimental campaign, we used all the 14 available NVIDIA data centers by instrumenting the client application. Finally, all 36 server IPs for PS Now belong to only 2 subnets /24 from the 33353 Sony AS. In terms of latency, additional `ping` measures show that Stadia and PS Now servers are 6-8 ms away from our location while Central European GeForce Now in the order of 15-20 ms. However, we cannot link this to the service quality or infrastructure deployment since we perform measurements from a single vantage point. Without traffic shaping, we did not qualitatively observe significant lag between users' commands and game response.

Finally, we analyze the domains that the applications contact during the gaming sessions. We extract them inspecting the DNS queries the client makes prior to open a new connection and extracting the SNI field from the TLS Client Hello messages. In case of Stadia, we only find `stadia.google.com` as domain specific to this service. Indeed, the client application contacts a dozen of other domains, but those are shared with all Google services (e.g., `gstatic.com` and `googleapis.com`), and, as such, not specific of Stadia. Regarding GeForce Now, the application contacts the general `nvidia.com` domain as well as the more specific `nvidiagrid.net`. Finally, PS Now relies on the `playstation .com` and `.net` domains and their subdomains, which are, thus, not specific to the PS Now service. Interestingly, GeForce Now is the only service that uses domains to identify gaming machines, using subdomains of `cloudmatchbeta.nvidiagrid.net`. Indeed, for the other services, the domains we find are associated uniquely to the control servers – used for login, static resources, etc. – while gaming machines are contacted without a prior DNS resolution.

Take away: *We seldom observe repeated IP addresses of the gaming machines for the three services. They are all located in the AS of the respective corporate. GeForce Now is the only one that identifies gaming machines via DNS domain names.*

5 Conclusions and future directions

In this paper, we showed important aspects of the recently launched services for cloud gaming, namely Stadia, GeForce Now and PS Now. They could potentially be used by millions of users in the close future, accounting for a large part of the Internet traffic. Indeed, Stadia

[11] We cannot infer if the server IP address acts as a ingress load balancer or reverse proxy.

[12] We map an IP address to the corresponding AS using an updated RIB from `http://www.routeviews.org/`.

and GeForce Now stream data up to 44 Mbit/s, which is, for instance, much higher than a 4K Netflix movie. However, fast 4G mobile networks can often sustain this load, while traditional 3G connections struggle. Cloud gaming applications rely on the standard RTP protocol for multimedia streaming, except for PS Now which does not use any documented protocol.

The aforementioned services entered the market between 2019 and 2020, and, as such, a lot of work is still to be done for characterizing them and understanding their impact on the network. Future directions include the study of their infrastructures from different points of view around the world with the goal of finding similarities and differences in the deployments. Moreover, we believe it is important to conduct campaigns aiming at measuring the subjective Quality of Experience (QoE) enjoyed by the users with human-in-the-loop controlled experiments and also with AI-bot players [28]. Finally, Microsoft and Amazon are going to release their cloud gaming platforms, namely xCloud and Luna, which are in early deployment at the time of writing this article. Any future work on this topic must include them.

Acknowledgments

The research leading to these results has been funded by the SmartData@PoliTO center for Big Data technologies.

References

[1] M. Trevisan, D. Giordano, I. Drago, M. M. Munafò, and M. Mellia, "Five years at the edge: Watching internet from the isp network," *IEEE/ACM Trans. on Networking*, vol. 28, no. 2, pp. 561–574, 2020.

[2] "Newzoo 2020 global games market report," https://newzoo.com/insights/articles/newzoo-games-market-numbers-revenues-and-audience-2020-2023/, accessed: January 2021.

[3] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "Rtp: A transport protocol for real-time applications," Internet Requests for Comments, RFC Editor, RFC, 07 2003. [Online]. Available: https://www.rfc-editor.org/rfc/rfc3550.txt

[4] W. Cai, R. Shea, C.-Y. Huang, K.-T. Chen, J. Liu, V. C. Leung, and C.-H. Hsu, "A survey on cloud gaming: Future of computer games," *IEEE Access*, vol. 4, pp. 7605–7620, 2016.

[5] W. Cai, R. Shea, C. Huang, K. Chen, J. Liu, V. C. M. Leung, and C. Hsu, "A survey on cloud gaming: Future of computer games," *IEEE Access*, vol. 4, pp. 7605–7620, 2016.

[6] C.-Y. Huang, C.-H. Hsu, Y.-C. Chang, and K.-T. Chen, "Gaminganywhere: an open cloud gaming system," in *Proceedings of the 4th ACM multimedia systems conference*, 2013, pp. 36–47.

[7] H.-J. Hong, D.-Y. Chen, C.-Y. Huang, K.-T. Chen, and C.-H. Hsu, "Placing virtual machines to optimize cloud gaming experience," *IEEE Transactions on Cloud Computing*, vol. 3, no. 1, pp. 42–53, 2014.

[8] A. Ojala and P. Tyrvainen, "Developing cloud business models: A case study on cloud gaming," *IEEE software*, vol. 28, no. 4, pp. 42–47, 2011.

[9] M. Jarschel, D. Schlosser, S. Scheuring, and T. Hoßfeld, "An evaluation of qoe in cloud gaming based on subjective tests," in *2011 Fifth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing.* IEEE, 2011, pp. 330–335.

[10] Y.-T. Lee, K.-T. Chen, H.-I. Su, and C.-L. Lei, "Are all games equally cloud-gaming-friendly? an electromyographic approach," in *2012 11th NetGames workshop.* IEEE, 2012, pp. 1–6.

[11] M. S. Hossain, G. Muhammad, B. Song, M. M. Hassan, A. Alelaiwi, and A. Alamri, "Audio–visual emotion-aware cloud gaming framework," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 12, pp. 2105–2118, 2015.

[12] W. Cai, C. Zhou, V. C. Leung, and M. Chen, "A cognitive platform for mobile cloud gaming," in *2013 IEEE 5th Int. Conference on Cloud Computing Technology and Science*, vol. 1. IEEE, 2013, pp. 72–79.

[13] K. Lee, D. Chu, E. Cuervo, J. Kopf, Y. Degtyarev, S. Grizan, A. Wolman, and J. Flinn, "Outatime: Using speculation to enable low-latency continuous interaction for mobile cloud gaming," in *Proceedings of the 13th MobySys*, 2015, pp. 151–165.

[14] S. Shi, C.-H. Hsu, K. Nahrstedt, and R. Campbell, "Using graphics rendering contexts to enhance the real-time video coding for mobile cloud gaming," in *Proceedings of the 19th ACM international conference on Multimedia*, 2011, pp. 103–112.

[15] H.-J. Hong, C.-F. Hsu, T.-H. Tsai, C.-Y. Huang, K.-T. Chen, and C.-H. Hsu, "Enabling adaptive cloud gaming in an open-source cloud gaming platform," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 12, pp. 2078–2091, 2015.

[16] Z. Qi, J. Yao, C. Zhang, M. Yu, Z. Yang, and H. Guan, “Vgris: Virtualized gpu resource isolation and scheduling in cloud gaming,” *ACM Transactions on Architecture and Code Optimization (TACO)*, vol. 11, no. 2, pp. 1–25, 2014.

[17] S. Choy, B. Wong, G. Simon, and C. Rosenberg, “The brewing storm in cloud gaming: A measurement study on cloud to end-user latency,” in *2012 11th NetGames workshop*. IEEE, 2012, pp. 1–6.

[18] M. Claypool, D. Finkel, A. Grant, and M. Solano, “Thin to win? network performance analysis of the onlive thin client game system,” in *2012 11th Annual Workshop on Network and Systems Support for Games (NetGames)*. IEEE, 2012, pp. 1–6.

[19] 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.

[20] 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, 2013.

[21] M. Manzano, J. A. Hernandez, M. Uruena, and E. Calle, “An empirical study of cloud gaming,” in *2012 11th Annual Workshop on Network and Systems Support for Games (NetGames)*. IEEE, 2012, pp. 1–2.

[22] M. Carrascosa and B. Bellalta, “Cloud-gaming: Analysis of google stadia traffic,” *arXiv preprint arXiv:2009.09786*, 2020.

[23] M. Trevisan, A. Safari Khatouni, and D. Giordano, “Errant: Realistic emulation of radio access networks,” *Computer Networks*, vol. 176, p. 07289, 2020.

[24] A. D. Domenico, G. Perna, M. Trevisan, L. Vassio, and D. Giordano, “A network analysis on cloud gaming: Stadia, geforce now and psnow,” 2021. [Online]. Available: https://dx.doi.org/10.21227/eb5y-ag12

[25] M. Trevisan, A. Finamore, M. Mellia, M. Munafo, and D. Rossi, “Traffic analysis with off-the-shelf hardware: Challenges and lessons learned,” *IEEE Communications Magazine*, vol. 55, no. 3, pp. 163–169, 2017.

[26] C. Holmberg, S. Hakansson, and G. Eriksson, “Web Real-Time Communication Use Cases and Requirements,” RFC 7478, Mar. 2015. [Online]. Available: https://rfc-editor.org/rfc/rfc7478.txt

[27] M. Petit-Huguenin and G. Salgueiro, “Multiplexing Scheme Updates for Secure Real-time Transport Protocol (SRTP) Extension for Datagram Transport Layer Security (DTLS),” RFC 7983, Sep. 2016. [Online]. Available: https://rfc-editor.org/rfc/rfc7983.txt

[28] G. Sviridov, C. Beliard, A. Bianco, P. Giaccone, and D. Rossi, “Removing human players from the loop: Ai-assisted assessment of gaming qoe,” in *to appear at 2020 INFOCOM Workshop on Network Intelligence - Learning and Optimizing Future Networks*, 2020.