



UNICA

UNIVERSITÀ  
DEGLI STUDI  
DI CAGLIARI



Università di Cagliari

## UNICA IRIS Institutional Research Information System

This is the Author's *accepted* manuscript version of the following contribution:

S. Porcu, L. Suyunov, A. Floris and L. Atzori, "NetRate: An Application for Collecting QoE, Energy, and Network Data on Android Devices," *2024 16th International Conference on Quality of Multimedia Experience (QoMEX)*, Karlshamn, Sweden, 2024, pp. 50-53.

© 2024 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

**The publisher's version is available at:**

<http://dx.doi.org/10.1109/QoMEX61742.2024.10598287>

**When citing, please refer to the published version.**

# NetRate: An Application for Collecting QoE, Energy, and Network Data on Android Devices

Simone Porcu<sup>1,2</sup>, Lazizjon Suyunov<sup>1</sup>, Alessandro Floris<sup>1,2</sup>, and Luigi Atzori<sup>1,2</sup>

<sup>1</sup>DIEE, University of Cagliari, 09123 Cagliari, Italy

<sup>2</sup>CNIT, University of Cagliari, 09123 Cagliari, Italy

{simone.porcu, lazizjon.suyunov, alessandro.floris84, l.atzori}@unica.it

**Abstract**—This paper proposes NetRate, an Android system monitoring application that can measure and collect network and energy information from Android devices and enables end users to rate the Quality of Experience (QoE) of selected applications. Moreover, NetRate includes a VPN service to limit the network capabilities of the end device to simulate constrained network conditions. NetRate supports the conduction of QoE research studies in the laboratory and the field by collecting different kinds of data from the end device, assisting in setting test conditions (e.g., with the VPN), and collecting subjective feedback.

**Index Terms**—Quality of Experience, Video Streaming, Android, QoE monitoring, Data collection.

## I. INTRODUCTION

The recent advancement of ICT technology has improved video streaming performance, enhancing user Quality of Experience (QoE). However, while delivering high-quality video content improves QoE, it also results in significant energy usage across the end-to-end video delivery chain. In particular, the overall ICT contribution to global emissions is estimated to be between 1.8% and 2.8% [1]. While the majority of literature studies propose QoE-centric management solutions with no concern for sustainability, a few recent studies have included energy consumption in their QoE-aware management solutions. It emerged that some users are willing to (slightly) sacrifice the perceived QoE if carbon emissions can be reduced [2]. However, the perceived QoE should be kept at an acceptable level to avoid the user becoming a churning [3].

For video streaming services, it was found that energy consumption can be reduced by streaming videos encoded at a lower resolution than the end device's native resolution. This would not necessarily lead to a noticeable QoE decrease especially on small-screen devices, such as smartphones [4]. However, further research studies are needed to be able to define accurate QoE models including energy-related factors that can support network and application providers to find the optimal trade-off between the provision of sustainable services while keeping acceptable QoE to the end users. Nonetheless,

This work has been partially supported by the European Union under the Italian National Recovery and Resilience Plan (NRRP) of NextGenerationEU, "Sustainable Mobility Center" (Centro Nazionale per la Mobilità Sostenibile, CNMS, CN\_00000023), by the PON "Ricerca e Innovazione" 2014-2020 (PON R&I) "Azione IV.4 Dottorati e contratti di ricerca su tematiche dell'innovazione" assigned with D.M. 1062 on 10.08.2021, and by the Italian Ministry of Enterprises and Made in Italy (MIMIT) within the 5G technology support program, on axis 1 "House of Emerging Technologies" (CTE), Project Name "Cagliari Digital Lab" (ID: G27F22000040008).

the data collection required to define these models is not straightforward.

For these reasons, we have developed an Android application, called NetRate, to collect network and energy-related data of Android devices, and to allow end users to provide subjective feedback concerning the perceived QoE of selected video streaming applications. Differently from previously developed applications, such as YoMoApp [5], NetRate can measure network (e.g., downlink/uplink speed) and energy (used/remaining battery capacity) information from Android devices while selected applications are used by the users. Moreover, NetRate includes a VPN service that can limit the network capabilities of the end device to simulate impaired network conditions.

We used the NetRate App to conduct experimental tests in the field involving 11 people, who were asked to install the App on their mobile devices and to use it to rate the QoE of video sessions consumed on two video streaming applications (TikTok and YouTube). Furthermore, using the VPN feature of the NetRate App, we collected the subjective feedback of the participants when their smartphones were connected to both a constrained (VPN on) and unconstrained (VPN off) network.

We present the NetRate application in Section II, whereas Section III discusses the conducted experimental test and achieved results. Finally, Section IV concludes the paper.

## II. NETRATE APPLICATION

*NetRate* is an Android system monitoring application providing the following main features:

- monitoring of the applications running on the Android device and identification of selected applications;
- monitoring of the changes in the battery charge status of the device to trigger actions;
- enabling the user to rate the QoE perceived for selected applications and showing push notifications;
- monitoring of the device's network download and upload speed;
- providing a VPN service to limit the network download speed of the Android device.

### A. Development details

NetRate has been developed using Kotlin and Java languages. The navigation system, crafted with Voyager alongside a customized structure, ensured a seamless user experience.

Adhering to software design principles such as SOLID (Single Responsibility, Open/Closed, Liskov Substitution, Interface Segregation, Dependency Inversion), KISS (Keep It Simple, Stupid), and DRY (Don't Repeat Yourself), the development incorporated clean architecture, clean code practices, and followed the Model-View-Intent (MVI) pattern. Jetpack Compose powered the user interface, facilitating smooth interaction, while Kotlin Coroutines and Flows managed parallel tasks efficiently. Background tasks were handled using Services and Broadcaster Receivers to facilitate communication between the Android system and background routines. Database management relied on SQLite and JSON, with Room serving as a crucial database helper. ViewModel classes were utilized to prepare data for presentation in application Activities, and Dagger Hilt was integrated to ensure dependency injection.

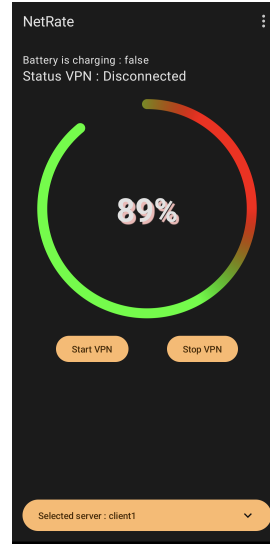
### B. Main features

The NetRate Home screen, illustrated in Fig. 1a, displays information concerning the device's battery charge status and the VPN status. The battery percentage is prominently displayed in the center of the screen, accompanied by buttons for starting and stopping the VPN service. A button at the bottom reveals a drop-down menu allowing VPN server selection (up to six servers). Upon selecting a new server, the user has to re-click the start button to connect to the VPN.

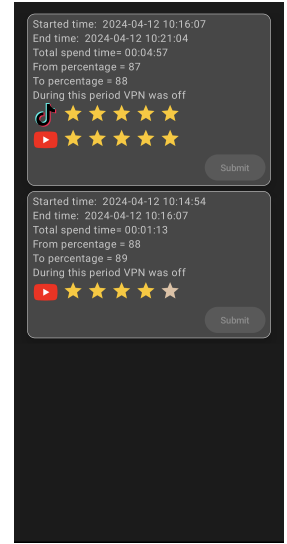
Push notifications for asking the user to provide feedback were triggered by a combination of two events: the identification of selected applications running in the foreground and the decrease in the battery percentage of the Android device. The reason is that Android does not allow us to check in real time which applications are running in the foreground. Still, it only allows us to check which foreground applications are running when the battery percentage is down by 1%. Thus, the user's feedback was collected every time the device's battery decreased its capacity and at least one of the monitored applications was used by the user. If the user were using multiple monitored applications during the battery's discharge cycle, he would be asked to rate each of these applications.

The Analytics screen, illustrated in Fig. 1b, presents information regarding the provided user feedback for the selected monitored applications (YouTube and TikTok are shown as an example in this case). It displays data in a list format, including information concerning the battery status (start time, end time, and duration related to the monitored battery percentage change, which is also displayed). The maximum, current, and remaining battery capacity in mAh are also measured and recorded although not displayed in the app. The VPN status (on/off) during the monitored period is also displayed. Finally, the subjective user feedback for the monitored applications is displayed. Note that users were asked to rate the QoE using the Absolute Category Rating (ACR) scale, which ranges from 1 (Bad) to 5 (Excellent) stars.

Finally, NetRate measures and records the device's average network download and upload speeds every 10 s. All data was stored locally on the device's database for analysis purposes.



(a) Home screen.



(b) Analytics screen.

Fig. 1: Home and Analytics screens of NetRate.

### C. VPN server

The VPN back-end software works on a server machine with an Intel(R) Xeon(R) Gold 5118 CPU and 8 GB of RAM running an Ubuntu Linux-based operating system. We have chosen to use OpenVPN version 2.4.4 because it enables us to handle multiple clients simultaneously. Each client is given a unique configuration file, facilitating their connection to the VPN server endpoint.

Once a client initiates a connection to the VPN server, the server processes the request and performs the following steps:

- **Client Authentication:** the VPN server authenticates the client based on the provided credentials (such as username and password or certificate). This ensures that only authorized clients can establish a connection.
- **IP Address Assignment:** to uniquely identify each connected client, the server assigns an internal static IP address, which is associated with the client during the entire session. By using static IP addresses, the server can maintain consistency and track individual users.
- **Linking IP to VPN Tunnel Interface:** the assigned static IP address is linked to the VPN tunnel interface on the server. This association allows the server to route traffic to and from the client using the correct IP address.
- **Routing and Forwarding:** the VPN server sets up routing rules to direct traffic between the client's static IP address and the rest of the network. It ensures that data flows correctly through the VPN tunnel.
- **Speed Limit Configuration:** as part of the process, the server also applies the predefined download speed limit to the client. This ensures fair resource allocation and prevents any single client from monopolizing bandwidth.

### III. EXPERIMENT AND RESULTS

As introduced in Section I, streaming lower video-quality content (in terms of spatial resolution and bitrate) consumes less energy through the end-to-end video streaming chain. However, users may not notice or partially notice this quality reduction, especially in small-screen end devices like smartphones. To investigate this subject, we set and conducted an experimental test in the field involving 11 people (4 females, 7 males, average 23 years old), who were asked to use their Android smartphones (screen resolution: Full-HD, size: 5/6") to watch and rate the quality of video sessions on YouTube and TikTok applications. Participants had to install the proposed NetRate application on their smartphones and were given instructions on the test conduction.

#### A. Experiment settings

The test lasted for six days during which test participants were asked to watch videos on YouTube and TikTok applications. NetRate was set to identify when these two applications were running on the Android devices of the test participants. NetRate showed push notifications to ask participants to provide QoE feedback for each of the identified applications each time the battery percentage of the device was decreased by 1%. Moreover, the VPN server was set to limit the download speed limit at a maximum of 1 MB/s when the VPN service was activated in the NetRate application. The reason was to force the two video streaming applications to show lower-quality video content since they are based on adaptive video streaming algorithms that adapt the video quality based on the available network capabilities of the client device. Participants had to activate the VPN on alternate days; they used the smartphone for 3 days under normal network conditions and 3 days under VPN-constrained network conditions. Since TikTok can also be used for messaging, we requested users not to rate those sessions even if videos were viewed during that time.

#### B. Results

A total of 817 ACR quality scores were collected at the end of the experimental test, 387 (195 TikTok, 192 YouTube) and 430 (231 TikTok, 199 YouTube) concerning the video sessions consumed under unconstrained (VPN-Off) and limited (VPN-On) network conditions, respectively. Table I summarizes the collected QoE results in terms of the Mean Opinion Score (MOS) with a 95% confidence interval. It can be seen that the MOS computed for video sessions consumed when the VPN was activated (3.14) is quite lower than the MOS computed when no VPN was enabled (3.82). This was expected since the average session downlink speed concerning the video sessions with VPN-On (106.31 kB/s) was lower than that measured with VPN-Off (130.98 kB/s). The same applies to the average maximum downlink speed, which was 223.84 kB/s for VPN-On and 671.29 kB/s for VPN-Off. Also, the limit of 1 MB/s forced by the VPN server is confirmed by the measured absolute maximum downlink speed, which was 936.46 kB/s for VPN-On and 7267.22 kB/s for VPN-Off. Moreover, on average, the participants perceived higher QoE for the TikTok

TABLE I: Collected results.  $N$ : number of samples. Session downlink speed measured in kB/s. TT: TikTok, YT: YouTube.

Test	N	Session Avg Downlink		MOS	CI
		Avg	Max (Avg)		
All	817	117.99	435.79	3.46	0.07
VPN-Off-TT-YT	387	130.98	671.29	3.82	0.10
VPN-Off-TT	195	150.47	750.83	3.97	0.13
VPN-Off-YT	192	111.17	590.51	3.67	0.15
VPN-On-TT-YT	430	106.31	223.84	3.14	0.10
VPN-On-TT	231	123.56	220.03	3.34	0.12
VPN-On-YT	199	86.28	228.27	2.90	0.16

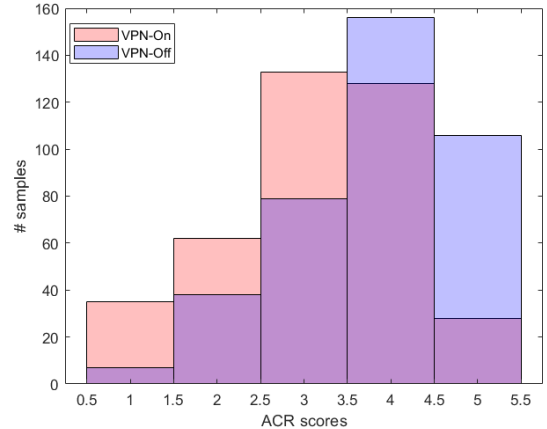


Fig. 2: Distribution of the collected ACR scores for the cases VPN-On and VPN-Off.

app than the YouTube one. The reason may be that TikTok videos are shorter than YouTube videos; also, TikTok applies caching techniques to preload the next videos that would be shown to the user.

In Fig. 2, we show the distribution of the collected ACR scores for the cases VPN-On and VPN-Off. It is evident that for VPN-Off the majority of scores (263) were provided for ACR 4-5, 79 for ACR 3, and 45 for ACR 1-2. Thus, most of the video sessions were perceived as good and excellent. On the other hand, for VPN-On the majority of scores (288) were provided for ACR 3-4, only 32 for ACR 5, and 110 for ACR 1-2. Thus, the constrained network has reduced the perceived QoE by 1 point of the ACR scale. However, most of the scores are still regarded as sufficient to good quality.

### IV. CONCLUSION

We have presented the capabilities of the proposed NetRate application, which can monitor the utilization of selected applications on Android systems and trigger the appearance of a rating screen to allow users to provide QoE feedback concerning consumed services. Experimental tests have shown that limiting the network download speed of the end devices decreased the perceived QoE for video streaming applications. However, the users still experienced a sufficient QoE. Further studies can be conducted to find the optimal trade-off between QoE and network/energy resources.

- [1] C. Freitag, M. Berners-Lee, K. Widdicks, B. Knowles, G. S. Blair, and A. Friday, "The real climate and transformative impact of ICT: A critique of estimates, trends, and regulations," *Patterns*, vol. 2, no. 9, p. 100340, 2021.
- [2] T. Hossfeld, M. Varela, L. Skorin-Kapov, and P. E. Heegaard, "A Greener Experience: Trade-offs between QoE and CO2 Emissions in Today's and 6G Networks," *IEEE Communications Magazine*, pp. 1–7, 2023.
- [3] M. Fiedler, A. Popescu, and Y. Yao, "QoE-Aware Sustainable Throughput for Energy-Efficient Video Streaming," in *2016 IEEE International Conferences on Big Data and Cloud Computing (BDCloud), Social Computing and Networking (SocialCom), Sustainable Computing and Communications (SustainCom)*, 2016, pp. 493–500.
- [4] G. Bingöl, A. Floris, S. Porcu, C. Timmerer, and L. Atzori, "Are Quality and Sustainability Reconcilable? A Subjective Study on Video QoE, Luminance and Resolution," in *2023 15th International Conference on Quality of Multimedia Experience (QoMEX)*, 2023, pp. 19–24.
- [5] F. Wamser, M. Seufert, P. Casas, R. Irmer, P. Tran-Gia, and R. Schatz, "YoMoApp: A tool for analyzing QoE of YouTube HTTP adaptive streaming in mobile networks," in *2015 European Conference on Networks and Communications (EuCNC)*, 2015, pp. 239–243.