Analyzing the Adoption of QUIC From a Mobile Development Perspective

Diego Madariaga
NIC Chile Research Labs,
University of Chile
diego@niclabs.cl

Lucas Torrealba
NIC Chile Research Labs,
University of Chile
lucas@niclabs.cl

Javier Madariaga
NIC Chile Research Labs,
University of Chile
javier@niclabs.cl

Javiera Bermúdez
NIC Chile Research Labs,
University of Chile
javi@niclabs.cl

Javier Bustos-Jiménez
NIC Chile Research Labs,
University of Chile
jbustos@niclabs.cl

ABSTRACT

First introduced in 2013, QUIC protocol has been in constant development and it has gained great importance over time, currently being standardized by the Internet Engineering Task Force (IETF). Previous experiments have shown that the performance of QUIC can be of particular interest for lossy wireless networks. Thus, it becomes relevant to measure and profile QUIC traffic from mobile devices in the wild. This paper analyzes the adoption of QUIC from a mobile development perspective, by using crowdsourced data from real Android users during the normal use of their mobile devices. The results obtained show a significant increase in the number of Android apps using QUIC and evidence newer efforts of companies to adopt this protocol. To the best of the authors' knowledge, this is the first work focused on analyzing and profiling QUIC traffic from crowdsourced network measurements taken by mobile end-user devices.

CCS CONCEPTS

• **Networks → Transport protocols**; **Network measurement**;

KEYWORDS

QUIC, passive measurements, mobile devices

EPIQ '20, August 10–14, 2020, Virtual Event, NY, USA

ACM ISBN 978-1-4503-8047-8/20/08…$15.00
https://doi.org/10.1145/3405796.3405830

ACM Reference Format:
Diego Madariaga, Lucas Torrealba, Javier Madariaga, Javiera Bermúdez, and Javier Bustos-Jiménez. 2020. Analyzing the Adoption of QUIC From a Mobile Development Perspective. In *Workshop on Evolution, Performance, and Interoperability of QUIC (EPIQ '20), August 10–14, 2020, Virtual Event, NY, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3405796.3405830

1 INTRODUCTION

QUIC protocol was designed by Google engineers to improve performance for HTTPS traffic, replacing most of the traditional HTTPS stack: HTTP/2, TLS, and TCP [9]. After years of experimentation, QUIC started to attract the attention of the community, being adopted by the IETF in 2016 for standardization[1], developing IETF QUIC in parallel with the development of Google QUIC (gQUIC). Accordingly, there has been an increasing amount of literature on analyzing QUIC protocol, as its standardization promotes a wider adoption.

Experiments from several studies have identified that QUIC performance outperforms TCP in most cases. Hence, QUIC has been shown to improve throughput [6, 7], to reduce page load time [2, 6, 18] and to improve user quality of experience (QoE) [1, 6, 9, 17]. In addition, research studies have found that QUIC connections are much less sensitive to loss than TCP connections [1, 2, 9]. Therefore, the adoption of QUIC takes on a great importance on more unstable networks, such as wireless networks.

Given the aforementioned, it becomes relevant to analyze the adoption of QUIC from the traffic generated by mobile devices. Internet traffic from mobile devices mostly goes through wireless networks, using WiFi or mobile data networks. The analysis of wireless traffic gains even more importance considering that by 2022, 71% of total IP traffic is expected to be wireless (51% WiFi and 20% Mobile) [4].

[1]https://datatracker.ietf.org/wg/quic/

For this work, an Android tool was developed to analyze all Internet traffic generated by users' applications in their normal use. This tool inserts itself in user-space as man-in-the-middle (MITM) into existing connections and extracts relevant information from each network flow.

By using this crowdsourcing measurement approach, the Android tool collected traffic information from mobile end-user devices accessing the Internet using both WiFi and mobile data networks. Therefore, the findings presented in this work take on great relevance as they are based on empirical traffic data measured in user-space, which accurately represent the Internet usage of mobile users in the wild.

This paper analyzes the collected data, in order to profile the adoption of QUIC from a mobile development perspective. That is, all the Android applications that were found to use QUIC are examined in detail, by inspecting their web-traffic behavior regarding the protocols used (HTTP, HTTPS and QUIC). In addition, the different IP addresses contacted by these applications are also analyzed, in order to profile the adoption of QUIC from the perspective of the companies serving content to mobile devices. As a result, the recent efforts of some organizations to adopt QUIC can be clearly identified.

To the best of the authors' knowledge, this is the first work focused on analyzing and profiling QUIC traffic from crowdsourced network measurements taken by mobile end-user devices.

2 RELATED WORK

A considerable amount of literature has been published on analyzing QUIC protocol. Some of these studies have focused on analyzing QUIC traffic from the overall network traffic, commonly using data provided by an ISP. These research studies estimated that between 2017 and 2018, QUIC accounted for 7-9% of total traffic volume [9, 10, 16]. However, these percentages are expected to be increased, since at the time of these studies almost all QUIC traffic was served by Google with few exceptions, such as the case of Snapchat [10]. By 2020, there are other notable cases of technology companies adopting QUIC. For example, Facebook started adopting QUIC by using its own implementation of IETF QUIC [5]. Additionally, Uber Technologies Inc. adopted QUIC for its services, using an implementation of gQUIC [11].

As QUIC started to attract increasing attention in the community, its performance has also been studied in a wide range of network scenarios. Indeed, there is a consensus that QUIC outperforms TCP for short-lived applications in the presence of random loss [13], and that the use of QUIC can improve throughput [6, 7], reduce page load time [2, 6, 18] and improve user quality of experience (QoE) [1, 6, 9, 17].

Closer to this work, some studies have focused on analyzing the performance of QUIC in wireless networks. Kharat et al. [7] reported that QUIC throughput outperforms TCP on a local test bed setup connected to a wireless network. Arisu et al. [1] indicated that QUIC provides better streaming and seeking experience than TCP in wireless networks. In addition, Li et al. [10] presented an analysis of the volume of QUIC traffic from a mobile network operator perspective. However, these results do not necessarily represent the overall experience of real users using QUIC, as these experiments are based on emulated network scenarios [7], the use of QUIC connections to a unique server [1] and the use of network data gathered from the infrastructure of an ISP and not in user-space [10].

3 MEASUREMENT METHODOLOGY

The network measurements used in this work refer to the development of an Android framework capable of taking network flow measurements through passive monitoring of active connections [8]. This is achieved by taking advantage of Android WorkManager API and Android VpnService.

3.1 Periodic Behavior

One of the main concerns when developing a monitoring system for mobile devices is the overhead it may produce in CPU and battery consumption. Therefore, a periodic monitoring system was chosen over a continuous one. With enough periodicity, the system is able to obtain a good representation of the network's traffic without overloading the devices.

WorkManager is an API provided by Android since 2018. This API makes it easy to schedule deferrable, asynchronous tasks that are expected to run even if the app exits or device restarts[2]. By using this API, the monitoring system is scheduled to start running for 1 minute every 15 minutes. The use of 15-minute intervals was chosen because it is the minimum duration that WorkManager allows for periodic tasks without them interfering with the device's battery optimization system.

3.2 Passive Monitoring

The monitoring system was designed to use Android's VpnService by implementing a local VPN server [15]. Thus, the measurement system gains packet-level access without requiring root privileges. This local VPN server inserts itself in user-space as man-in-the-middle (MITM) into existing connections on user devices and extracts, among others, the following relevant information from each network flow:

- Destination IP and destination port
- Transport protocol used

[2]https://developer.android.com/topic/libraries/architecture/workmanager

- Start time and end time with microsecond precision
- Number of transmitted and received bytes
- Connection type used (WiFi or Mobile Network)
- Network technology (UMTS, HSDPA, LTE, etc.), in case of using mobile Internet

3.3 Mobile Crowdsourcing App

To test the measurement methodology with real users, a recruitment campaign was conducted to distribute a mobile crowdsourcing app to people affiliated to the University of Chile in the city of Santiago, Chile. From February to April 2020, network data was collected from ~160 real users that voluntarily accepted to install the app in their personal devices. In total, the application registered information about ~175,000 executions of the 1-min measurement system. Thus, the monitoring system collected information about ~1,850,000 Internet traffic flows (~1,635,000 TCP connections and ~215,000 UDP connections). These connections were established by 831 different Android applications to ~35,000 different IP addresses. As the users were recruited following a convenience sampling process, the Android applications present in the collected dataset are influenced by the over-representation of university students in the population sample.

Ethical Considerations. In order to keep the ethical considerations of network measurements and management of user data, this study was conducted ensuring that full permissions are sought from and provided by users of the measurement app (PePa Ping app [8]). After being shown a full description of PePa Ping project (the collected information, how data will be used and who will access and use the dataset), we ask participants to grant us permission to use their collected information for project-related research activities before installing the crowdsourcing application. Collected measurements are appropriately anonymized and only researchers affiliated to this project (and external researchers after signing an NDA) are allowed to access the data. Furthermore, this study has been approved by the ethics committee of the University of Chile.

3.4 Data Processing

In order to gain further insights about network flows, we performed a data processing on the measurements to obtain more information of each flow captured from our passive Android application.

We used nmap tool [3] to check the service running on each pair `dst_ip:dst_port` from the collected dataset. Thus, we can list the pairs identified as HTTP or HTTPS services. This provides a more complete service analysis than common protocol analysis, due to the use of non-default ports for HTTP and HTTPS services in mobile applications [14].

Regarding the network flows that appears to be using QUIC protocol (UDP flows to a destination IP on port 443), we performed an analysis to check the availability of QUIC on these IP addresses. We used two complementary methodologies to check the supported versions of both IETF QUIC and gQUIC protocols on these web servers. Firstly, we tried to connect to these IP addresses using TCP/TLS connections. In case of success, we checked the `Alt-Svc` header in their HTTP responses (if available), where the servers can advertise support for different QUIC versions [9]. Secondly, we used LSQUIC library[4] to check the support for different QUIC versions on the servers analyzed. Hence, by following these two methodologies, ~160.000 network flows were classified with certainty as QUIC connections (from a total of ~215.000 UDP connections).

Therefore, all network flows for web traffic were classified into HTTP, HTTPS or QUIC traffic. Additionally, for each IP address running a web service, we tried to establish a secure HTTP connection. In case of success, we analyzed its SSL certificate and obtained the server's common name and organization. Using this methodology, the 82% of all IP addresses running a web service were successfully resolved to a common name and an organization. In particular, all IP addresses running QUIC were included in this 82%, that is to say, there was no endpoint in our dataset that only supported QUIC and no TCP.

4 CHARACTERIZING QUIC TRAFFIC FROM ANDROID DEVICES

4.1 General Overview

At analyzing the overall network traffic information, that is, all traffic going through both WiFi and mobile data networks, 23.05% of total traffic volume generated by Android devices was found to use QUIC. Table 1 shows the percentages of HTTP, HTTPS and QUIC traffic from mobile devices in WiFi and mobile data networks. According to these results, QUIC traffic accounted for 26.16% of WiFi traffic, and 10.56% of mobile data traffic. These values are higher than the percentages of QUIC traffic previously estimated: 7-9% of total traffic by 2017-2018 [9, 10, 16]. This higher presence of QUIC in the overall network traffic can be explained by the following:

- YouTube app uses QUIC protocol for Android devices since 2016 [9]. Then, the increase in the use of video traffic [4], and particularly in YouTube traffic, is also reflected in an increase of QUIC traffic.
- This study only considered traffic from Android mobile devices. Thus, there may be an overestimation of QUIC

[3] https://nmap.org/

[4] https://github.com/litespeedtech/lsquic

traffic, as there is a high number of Google apps that use QUIC and came pre-installed on Android devices.

- As mentioned in Section 2, more non-Google applications have started to adopt QUIC. This supports the idea that the wider adoption of QUIC is a real phenomenon.

Table 1: Percentage of web traffic volume for each protocol used: HTTP, HTTPS and QUIC.

	HTTP	HTTPS	QUIC
All Traffic	3.16%	70.48%	23.05%
WiFi	3.63%	66.93%	26.16%
Mobile	1.53%	82.46%	10.56%

Table 1 also evidences a higher percentage of QUIC traffic for WiFi with respect to mobile data networks (26.16% for WiFi and 10.56% for mobile). This can be explained by taking into account the different Internet usage behaviors when using WiFi or mobile networks. Figure 1 shows that the use of applications related to YouTube (YouTube, YouTube Music, YouTube Go and YouTube Vanced) was higher when using WiFi (25.6%) than when using mobile Internet (9.1%). This represents an increase in the percentage of QUIC traffic in WiFi networks as, according to the collected dataset, 96.41% of YouTube traffic used QUIC.

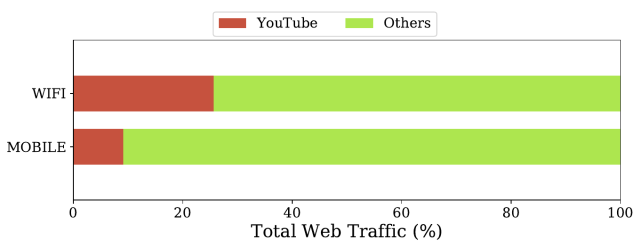

Figure 1: Percentage of web traffic volume generated by YouTube and non-YouTube applications.

4.2 Supported QUIC implementations

According to the crowdsourced dataset collected, all QUIC flows were served by a set of 1792 different IP addresses. By following the methodology explained in Section 3.4, 1681 of these IP addresses were resolved to Google LLC. Figure 2 shows the supported QUIC versions by these 1681 IP addresses.

As previously presented by Rüth et al. [16], Figure 2 shows a gradual adoption of newer versions of gQUIC, keeping backwards compatibility with some older versions. In this way, almost all these 1681 Google IP addresses supported the same set of six different gQUIC versions. At the time of research and according to the collected dataset, the most recent versions of gQUIC used in the wild corresponded to Google QUIC 50 (Q050) and Google QUIC with TLS 50 (T050).

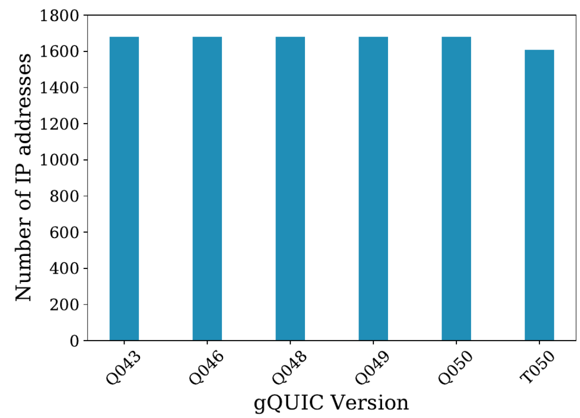

Figure 2: Number of Google IP addresses supporting different gQUIC versions.

At analyzing the remaining 111 non-Google IP addresses supporting QUIC, there was more diversity regarding the support of gQUIC and IETF QUIC versions. Figure 3 shows more homogeneity in the supported gQUIC versions, reporting a gradual adoption of newer versions of gQUIC, keeping backwards compatibility with some older versions (as for Google IP addresses in Figure 2). In contrast, IP addresses supporting IETF QUIC tend to support only its last version without keeping backwards compatibility, as previously discussed by Piraux et al. [12]. At the time of research and according to the collected dataset, the most recent version of

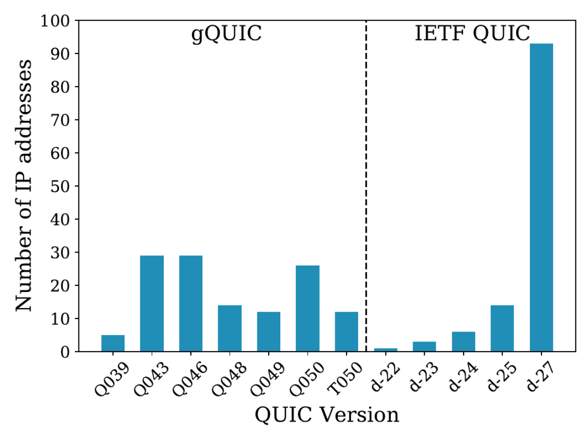

Figure 3: Number of non-Google QUIC IP addresses supporting different QUIC versions.

IETF QUIC used in the wild corresponded to the IETF QUIC Specification, Draft 27.

4.3 Android apps using QUIC

Li et al. [10] concluded that by 2018 Snapchat was the only mobile application using QUIC other than Google's applications. In the same way, Razaghpanah et al. [14, 15] found that by 2018 only 32 Android apps (from over 67K apps) used QUIC, where all of these 32 apps but Snapchat were developed by Google. In contrast to these earlier findings, the present study found 173 different Android apps using QUIC, from a total of 831 apps analyzed. That is to say, 20.8% of all Android apps present in the collected data used QUIC. This represents an important increase in the number of Android applications adopting QUIC protocol, considering the lower number of applications taken into account (831) in comparison to other studies.

Table 2 reports the 20 Android apps that account for the higher volume of QUIC traffic. 11 of these 20 apps are developed by Google, where YouTube app presents a remarkably higher volume of QUIC traffic compared to the rest of the applications. The diversity of developers in the remaining 9 apps is meaningful as it shows that other popular non-Google apps are adopting QUIC. It is important to mention that, although the pre-installed applications depend upon the device manufacturer and the version of Android, most of the Google apps in Table 2 usually come pre-installed on Android devices [3].

Table 2: Android apps with higher volume of QUIC traffic.

Application Name	Developer	Total QUIC Traffic [bytes]
YouTube	Google LLC	23,173,697,299
Google Chrome	Google LLC	1,578,289,356
Google Photos	Google LLC	1,362,299,992
Facebook	Facebook	1,223,299,088
YouTube Music	Google LLC	758,320,272
YouTube Vanced	Team Vanced	608,753,978
Google	Google LLC	480,459,141
Maps	Google LLC	151,257,267
Google Play Store	Google LLC	132,914,279
Gmail	Google LLC	102,183,775
WhatsApp Messenger	WhatsApp Inc.	23,428,138
Instagram	Instagram	20,466,458
Hangouts	Google LLC	19,636,498
Google Duo	Google LLC	14,761,958
Candy Crush Soda Saga	King	10,435,317
Pinterest	Pinterest	10,023,735
Snapchat	Snap Inc	8,512,697
Messenger	Facebook	7,294,965
Google News	Google LLC	7,204,840
Uber	Uber Technologies, Inc.	6,484,651

At analyzing the traffic from these 173 apps, 93% of all their QUIC traffic was found to be served by Google LLC, even though only 20 of the 173 apps using QUIC are developed by Google. The remaining QUIC traffic is mainly served by Facebook Inc., Snap Inc. and Uber Technologies Inc. As mentioned in Section 2, Facebook QUIC servers use an implementation of IETF QUIC, whereas QUIC servers from Uber Technologies Inc. use an implementation of gQUIC. In addition, QUIC servers from Snap Inc. were found to also implement gQUIC. Figure 4 shows the distribution of all web traffic served by these four major organizations that support QUIC.

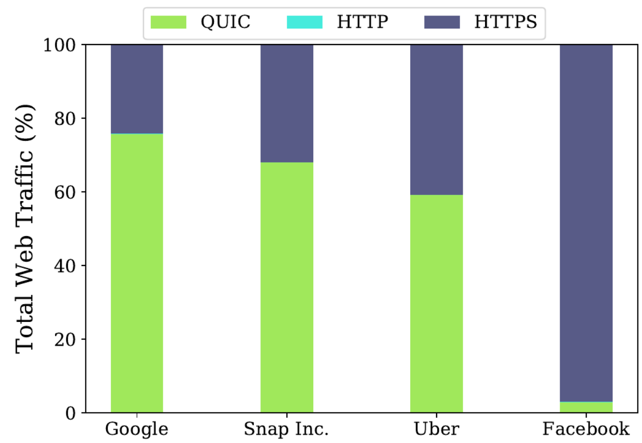

Figure 4: Percentage of web traffic volume for each protocol used: QUIC, HTTP, HTTPS. Traffic separated according to different organizations.

According to the figure, most of the web traffic served by Google, Snap and Uber to mobile devices used QUIC protocol, accounting for 75.77%, 67.98% and 59.12% of their total web traffic volume, respectively. In contrast, a low portion of the traffic served by Facebook to mobile applications (including Instagram and WhatsApp) used QUIC, accounting for 3.07% of total web traffic volume served by this company. It is important to mention that the total web traffic volume served by Google (QUIC, HTTP and HTTPS traffic) corresponded to 29.8% (43.2 GB) of total web traffic volume in the collected dataset, whereas the web traffic volume served by Facebook corresponded to 40.0% (57.9 GB) of total web traffic volume. In addition, the web traffic volume served by Snap Inc. and Uber Technologies Inc, corresponded to 0.001% and 0.011% of total web traffic volume, respectively.

Additionally, the 173 Android apps that were found to use QUIC, can be aggregated in the following meaningful groups:

- Google Apps: Apps developed by Google

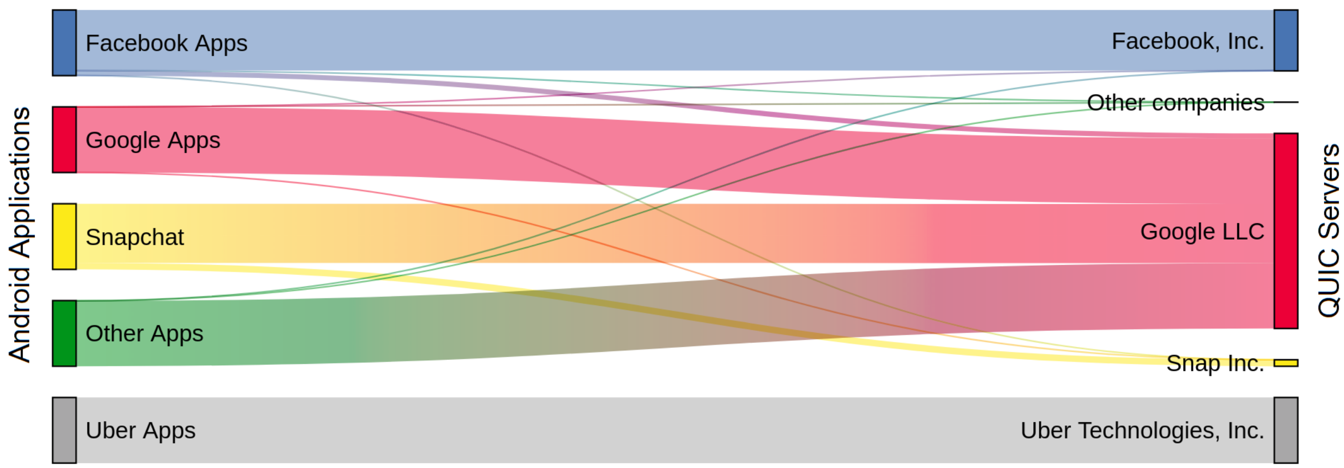

Figure 5: QUIC traffic distribution from Android applications to different organizations.

- Facebook Apps: Apps developed or owned by Facebook (including Instagram and WhatsApp)
- Uber Apps: Apps developed by Uber
- Snapchat: Snapchat app
- Other Apps: All remaining apps

Figure 5 shows the distribution of QUIC traffic from these Android apps to QUIC servers belonging to different organizations. According to the dataset used in this study, all QUIC traffic generated by Uber's apps was served by their own servers. Similarly, almost all QUIC traffic generated by Facebook apps was served by Facebook itself, and almost all QUIC traffic generated by Google apps was delivered by Google. In contrast, the majority of QUIC traffic generated by Snapchat app was served by Google, whereas only a short portion was delivered by Snap servers. Finally, regarding the group of Other Apps, which aggregates the majority of the apps using QUIC (144 out of 173), almost all of their QUIC traffic was served by Google. At analyzing in detail these apps, 80% of all their QUIC connections were resolved to only five different common names:

(1) *.g.doubleclick.net
(2) dns.google
(3) *.google.com
(4) *.googlevideo.com
(5) *.google-analytics.com

All these common names belong to Google, and seems to be highly related to common embedded Google SDKs, such as Google Analytics SDK and Google Mobile Ads SDK.

5 CONCLUSION

In this paper, we presented a study on profiling the adoption of QUIC based on empirical crowdsourced traffic data measured in mobile end-user devices. By using this crowdsourcing approach, we collected valuable traffic information directly in user-space, providing new insights with respect to previous related research studies.

The results obtained show the increased QUIC adoption at analyzing the network traffic from Android devices. We reported a significant larger number of Android applications using QUIC and evidenced an increasing QUIC adoption by major mobile applications and technology companies. In this context, we performed an empirical study on QUIC traffic profiling the Android apps using QUIC, the organizations serving this content and the flow of QUIC traffic among these actors.

The findings present in this work take on great relevance as they are based on empirical traffic data measured in user-space, which accurately represent the Internet usage of mobile users in the wild.

As future work, it is possible to perform a wider analysis of the adoption of QUIC in mobile devices, by continuously monitoring the data collected by our crowdsourcing application. Then, a temporal analysis could be performed to track the evolution of QUIC from a mobile development perspective during a longer period.

ACKNOWLEDGEMENTS

This work was partially funded by the National Agency for Research and Development (ANID) / Scholarship Program / Doctorado Nacional 2019 - 21190450.

REFERENCES

[1] Sevket Arisu and Ali C Begen. 2018. Quickly starting media streams using QUIC. In *Proceedings of the 23rd Packet Video Workshop.* 1–6.

[2] Sarah Cook, Bertrand Mathieu, Patrick Truong, and Isabelle Hamchaoui. 2017. QUIC: Better for what and for whom?. In *2017 IEEE International Conference on Communications (ICC)*. IEEE, 1–6.

[3] Julien Gamba, Mohammed Rashed, Abbas Razaghpanah, Juan Tapiador, and Narseo Vallina-Rodriguez. 2019. An Analysis of Pre-installed Android Software. *arXiv preprint arXiv:1905.02713* (2019).

[4] Cisco Visual Networking Index. 2019. Global Mobile Data Traffic Forecast Update, 2017–2022. *White paper* (2019).

[5] Subodh Iyengar and Luca Niccolini. 2018. Moving Fast at Scale: Experience Deploying IETF QUIC at Facebook. In *Proceedings of the Workshop on the Evolution, Performance, and Interoperability of QUIC (EPIQ'18)*. Association for Computing Machinery, New York, NY, USA, Keynote.

[6] Arash Molavi Kakhki, Samuel Jero, David Choffnes, Cristina Nita-Rotaru, and Alan Mislove. 2017. Taking a long look at QUIC: an approach for rigorous evaluation of rapidly evolving transport protocols. In *Proceedings of the 2017 Internet Measurement Conference*. 290–303.

[7] Prashant K Kharat, Aniket Rege, Aneesh Goel, and Muralidhar Kulkarni. 2018. QUIC Protocol Performance in Wireless Networks. In *2018 International Conference on Communication and Signal Processing (ICCSP)*. IEEE, 0472–0476.

[8] NIC Chile Research Labs. 2020. PePa Ping: Measuring QoS for mobile Internet. (2020). https://niclabs.cl/pepa/

[9] Adam Langley, Alistair Riddoch, Alyssa Wilk, Antonio Vicente, Charles Krasic, Dan Zhang, Fan Yang, Fedor Kouranov, Ian Swett, Janardhan Iyengar, et al. 2017. The quic transport protocol: Design and internet-scale deployment. In *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*. 183–196.

[10] Feng Li, Xiaoxiao Jiang, Jae Won Chung, and Mark Claypool. 2018. Who is the King of the Hill? Traffic Analysis over a 4G Network. In *2018 IEEE International Conference on Communications (ICC)*. IEEE, 1–6.

[11] Rajesh Mahindra, Vinoth Chandar, and Ethan Guo. 2019. Employing QUIC Protocol to Optimize Uber's App Performance. (2019). https://eng.uber.com/employing-quic-protocol/

[12] Maxime Piraux, Quentin De Coninck, and Olivier Bonaventure. 2018. Observing the evolution of QUIC implementations. In *Proceedings of the Workshop on the Evolution, Performance, and Interoperability of QUIC*. 8–14.

[13] Peng Qian, Ning Wang, and Rahim Tafazolli. 2018. Achieving robust mobile web content delivery performance based on multiple coordinated QUIC connections. *IEEE Access* 6 (2018), 11313–11328.

[14] Abbas Razaghpanah, Julien Gamba, Álvaro Feal, Narseo Vallina-Rodriguez, Mark Allman, and Phillipa Gill. 2018. A Multi-Protocol Analysis of Android App's Traffic From The Edge. (2018). http://eprints.networks.imdea.org/1907/1/poster.pdf

[15] Abbas Razaghpanah, Narseo Vallina-Rodriguez, Srikanth Sundaresan, Christian Kreibich, Phillipa Gill, Mark Allman, and Vern Paxson. 2015. Haystack: In situ mobile traffic analysis in user space. *arXiv preprint arXiv:1510.01419* (2015), 1–13.

[16] Jan Rüth, Ingmar Poese, Christoph Dietzel, and Oliver Hohlfeld. 2018. A First Look at QUIC in the Wild. In *International Conference on Passive and Active Network Measurement*. Springer, 255–268.

[17] Jan Rüth, Konrad Wolsing, Klaus Wehrle, and Oliver Hohlfeld. 2019. Perceiving QUIC: do users notice or even care?. In *Proceedings of the 15th International Conference on Emerging Networking Experiments And Technologies*. 144–150.

[18] Konrad Wolsing, Jan Rüth, Klaus Wehrle, and Oliver Hohlfeld. 2019. A performance perspective on web optimized protocol stacks: TCP+ TLS+ HTTP/2 vs. QUIC. In *Proceedings of the Applied Networking Research Workshop*. 1–7.