Poster: QUIC is not Quick Enough over Fast Internet

Xumiao Zhang
University of Michigan

Shuowei Jin
University of Michigan

Yi He
University of Michigan

Ahmad Hassan
University of Southern California

Z. Morley Mao
University of Michigan

Feng Qian
University of Southern California

Zhi-Li Zhang
University of Minnesota

ABSTRACT

QUIC is a multiplexed transport-layer protocol over UDP and comes with enforced encryption. It is expected to be a game-changer in improving web application performance. Together with the network layer and layers below, UDP, QUIC, and HTTP/3 form a new protocol stack for future network communication, whose current counterpart is TCP, TLS, and HTTP/2. In this study, to understand QUIC's performance over high-speed networks and its potential to replace the TCP stack, we carry out a series of experiments to compare the UDP+QUIC+HTTP/3 (QUIC) stack and the TCP+TLS+HTTP/2 (HTTP/2) stack. Preliminary measurements on file download reveal that QUIC suffers from a data rate reduction compared to HTTP/2 across different hosts.

ACM Reference Format:
Xumiao Zhang, Shuowei Jin, Yi He, Ahmad Hassan, Z. Morley Mao, Feng Qian, and Zhi-Li Zhang. 2023. Poster: QUIC is not Quick Enough over Fast Internet. In *Proceedings of the 2023 ACM Internet Measurement Conference (IMC '23), October 24–26, 2023, Montreal, QC, Canada.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3618257.3625002

1 INTRODUCTION

QUIC is a transport-layer protocol, initially developed by Google [13] with the goal of enabling fast, reliable, and secure connections. In 2016, an IETF working group was launched to improve the original QUIC design which fuses the transport, cryptographic handshakes, and upper-layer HTTP by teasing various functionalities into components, and standardize it into IETF QUIC (RFC 9000 [11]) as the transport layer basis of HTTP/3 (RFC 9114 [8]). QUIC is now responsible for over 75% of Meta's Internet traffic [3], and its adoption continues to grow fast [4, 5]. Key advantages of the QUIC design include 0/1-RTT handshake, removal of head-of-line blocking, and connection migration. However, there are also potential downsides. One notable concern is the overhead associated with processing and copying data between the kernel space and user space. Downloading data over QUIC can become very slow in some cases.

QUIC has attracted wide research attention. There is a plethora of literature on characterizing QUIC [10, 12, 14, 19, 20, 23]. Nevertheless, existing studies use diverse QUIC implementations, compute environments (mobile vs. desktop), and network conditions (wired vs. wireless). Due to such diversity, their findings are understandably a mixture of performance gains and degradations compared to TCP or earlier generations of HTTP. Moreover, a majority of these studies focus on low-throughput use cases. In this study, we propose to systematically examine QUIC's performance over modern fast Internet (*i.e.,* with close to or over 1 Gbps bandwidth). We have performed a series of experiments to compare the UDP+QUIC+HTTP/3 (QUIC) and TCP+TLS+HTTP/2 (HTTP/2) stacks and are planning on more comprehensive evaluations.

IMC '23, October 24–26, 2023, Montreal, QC, Canada

ACM ISBN 979-8-4007-0382-9/23/10.
https://doi.org/10.1145/3618257.3625002

2 PRELIMINARY RESULTS

We first conduct a preliminary experiment on both desktop and mobile Chrome browsers to download 1 GB files. We deploy a server equipped with an Intel Xeon E5-2640 CPU, hosting an HTTP server using OpenLiteSpeed [6]. On the client side, we use a desktop machine featuring an Intel Core i7-6700 CPU. Both machines run Ubuntu 18.04 and are connected through a 1 Gbps Ethernet. We execute mobile tests on a Google Pixel 5 supporting both low-band and mmWave 5G cellular networks. Low-band 5G typically has a throughput of hundreds of Mbps and mmWave 5G can easily achieve up to 2 Gbps [16, 24]. Table 1 presents the results averaged over 10 trials. The download throughput when QUIC is enabled is about half of the throughput with QUIC disabled. This throughput disparity is even larger on the smartphone. The program's CPU usage is also higher during the QUIC download. Note that the CPU usage for the desktop is measured from the browser's network service while the measurement refers to the entire browser's CPU usage for the smartphone.

Based on the preliminary results, we raise a couple of questions: When is QUIC data transfer slower than HTTP/2? What underlying reasons contribute to the performance gap? Can users benefit from the current deployment of QUIC?

To dig deeper, we compare the QUIC and HTTP/2 stacks in a simplified environment to isolate different potential factors. Specifically, we employ two download tools, `cURL` [1] and `quic_client` [7]. `cURL` is a command-line data transfer tool that supports both QUIC and HTTP/2. `quic_client` is a standalone QUIC client, built with the same QUIC stack as Chrome and Chromium. We use the two clients to run 1 GB file download experiments on the previously described Server-Ethernet-Desktop setup. We make the best efforts to ensure consistency across other factors, such as congestion control algorithms (CUBIC), TLS configurations, server software/configurations, and network conditions, between the two protocol stacks. We run `cURL` over both QUIC and HTTP/2 and run `quic_client` over QUIC. We control the available network

Table 1: Preliminary file download tests.

Testbed	Download Throughput (Mbps)		CPU Usage (%)	
	HTTP/2	HTTP/3	HTTP/2	HTTP/3
Desktop, Ethernet	924	472 (-49%)	77.5	96.9
Pixel 5, low-band 5G	234	113 (-52%)	121.6	161.8
Pixel 5, mmWave 5G	324	136 (-58%)	128.4	165.2

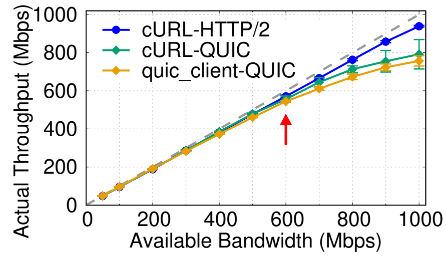

Figure 1: File download throughput under limited bandwidth.

bandwidth from 50 Mbps to 1000 Mbps using Linux `tc` [2]. Figure 1 shows the actual download throughput and controlled available bandwidth results. We find that, when the bandwidth is low, QUIC and HTTP/2 exhibit similar performance. Both QUIC clients can catch up with the available bandwidth, though `quic_client`'s throughput is slightly lower. However, when the bandwidth grows over around 600 Mbps, QUIC consistently falls behind HTTP/2 by up to 49%. The performance gap becomes more pronounced as the available bandwidth increases.

3 RELATED WORK

Since its advent in 2013, QUIC has been widely researched in numerous studies. Google reports its experience with QUIC after years of Internet-wide deployment [13]. QUIC's rapid evolution has led to efforts investigating the interoperability between QUIC implementations [14]. There are solutions proposed for rigorous evaluation of QUIC [12, 18]. QScanner [25] is implemented to analyze early QUIC deployments. Existing research has also studied the impact of QUIC on congestion control [9, 15] and various applications [10, 19–21, 23]. In addition, some works have explored QUIC optimizations [17, 22].

4 ONGOING WORK AND CONCLUSION

We are continuously working to generate results on more complicated workloads such as file transfers using commercial web browsers, web page loading, and video streaming. We also plan to expand the experiments to include more network types. At the same time, we aim to find out the root causes for QUIC's slowness. At a high level, we advocate careful examinations of upper-layer protocols over emerging networks, applications, and services. This study instantiates this idea by conducting a pioneering study on QUIC over high-speed Internet. We hope that it can spur more research to improve QUIC for future deployment.

ACKNOWLEDGMENTS

This work was supported in part by NSF under Grants CNS-2106771, CNS-2128489, and CNS-2220286.

REFERENCES

[1] 1998. cURL - command line tool and library for transferring data with URLs. https://curl.se/.
[2] 2001. Linux Traffic Control (tc). https://man7.org/linux/man-pages/man8/tc.8.html.
[3] 2020. How Facebook is bringing QUIC to billions. https://engineering.fb.com/2020/10/21/networking-traffic/how-facebook-is-bringing-quic-to-billions/.
[4] 2021. HTTP/3 and QUIC: Past, Present, and Future. https://www.akamai.com/blog/performance/http3-and-quic-past-present-and-future.
[5] 2022. HTTP RFCs have evolved: A Cloudflare view of HTTP usage trends. https://blog.cloudflare.com/cloudflare-view-http3-usage/.
[6] 2023. OpenLiteSpeed. https://openlitespeed.org/.
[7] 2023. The Chromium Projects. https://www.chromium.org/Home/.
[8] Mike Bishop. 2022. HTTP/3. *RFC 9114, IETF* (2022).
[9] Mathis Engelbart and Jörg Ott. 2021. Congestion control for real-time media over QUIC. In *Proceedings of the 2021 Workshop on Evolution, Performance and Interoperability of QUIC*. 1–7.
[10] Anirudh Ganji and Muhammad Shahzad. 2021. Characterizing the Performance of QUIC on Android and Wear OS Devices. In *2021 International Conference on Computer Communications and Networks (ICCCN)*. IEEE, 1–11.
[11] Jana Iyengar and Martin Thomson. 2021. QUIC: A UDP-based multiplexed and secure transport. *RFC 9000, IETF* (2021).
[12] 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.
[13] 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.
[14] Robin Marx, Joris Herbots, Wim Lamotte, and Peter Quax. 2020. Same standards, different decisions: A study of QUIC and HTTP/3 implementation diversity. In *Proceedings of the Workshop on the Evolution, Performance, and Interoperability of QUIC*. 14–20.
[15] Ayush Mishra, Sherman Lim, and Ben Leong. 2022. Understanding speciation in QUIC congestion control. In *Proceedings of the 22nd ACM Internet Measurement Conference*. 560–566.
[16] Arvind Narayanan, Xumiao Zhang, Ruiyang Zhu, Ahmad Hassan, Shuowei Jin, Xiao Zhu, Xiaoxuan Zhang, Denis Rybkin, Zhengxuan Yang, Zhuoqing Morley Mao, et al. 2021. A variegated look at 5G in the wild: performance, power, and QoE implications. In *Proceedings of the 2021 ACM SIGCOMM 2021 Conference*. 610–625.
[17] Gustavo Pantuza, Marcos AM Vieira, and Luiz FM Vieira. 2021. eQUIC gateway: Maximizing QUIC throughput using a gateway service based on eBPF+ XDP. In *2021 IEEE Symposium on Computers and Communications (ISCC)*. IEEE, 1–6.
[18] 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.
[19] 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.
[20] Tanya Shreedhar, Rohit Panda, Sergey Podanev, and Vaibhav Bajpai. 2021. Evaluating QUIC Performance Over Web, Cloud Storage, and Video Workloads. *IEEE Transactions on Network and Service Management* 19, 2 (2021), 1366–1381.
[21] 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.
[22] Xiangrui Yang, Lars Eggert, Jörg Ott, Steve Uhlig, Zhigang Sun, and Gianni Antichi. 2020. Making quic quicker with nic offload. In *Proceedings of the Workshop on the Evolution, Performance, and Interoperability of QUIC*. 21–27.
[23] Alexander Yu and Theophilus A Benson. 2021. Dissecting performance of production QUIC. In *Proceedings of the Web Conference 2021*.
[24] Xumiao Zhang, Xiao Zhu, Yihua Ethan Guo, Feng Qian, and Z Morley Mao. 2019. Poster: characterizing performance and power for mmWave 5G on commodity smartphones. In *Proceedings of the 2019 on Wireless of the Students, by the Students, and for the Students Workshop*. 14–14.
[25] Johannes Zirngibl, Philippe Buschmann, Patrick Sattler, Benedikt Jaeger, Juliane Aulbach, and Georg Carle. 2021. It's over 9000: analyzing early QUIC deployments with the standardization on the horizon. In *Proceedings of the 21st ACM Internet Measurement Conference*. 261–275.