QUIC/UDP will replace HTTP over TCP

Vishwas Gowdihalli Mahalingappa
Rutgers University - New Brunswick
vg421@scarletmail.rutgers.edu

ABSTRACT

QUIC/UDP will replace HTTP over TCP in the near future. Google originally introduced QUIC as an experimental extension to their Chrome browser in 2013. The preferred protocol for networked communication of Google services during the past few years has been QUIC. QUIC has been globally deployed at Google on thousands of servers and is used to serve traffic to a range of clients including a widely used web browser (Chrome) and a popular mobile video streaming app (YouTube) [2]. It is estimated that 7% of Internet traffic is now QUIC. It is critical to comprehend QUIC since it will be the Internet's future.

1 INTRODUCTION

As we can see in the figure (1), QUIC is built on top of UDP and includes encryption [1]. UDP is a connectionless protocol, therefore QUIC conducts all the calculations to ensure a trustworthy connection between a client and a server. Additionally, because QUIC is integrated into the application layer, any updates don't call for an OS upgrade. The primary advantages of QUIC over TCP in terms of performance are as follows:

1. Connection handshake
2. Multiplexing

Connection handshake: In addition to negotiating the TLS connection, TCP needs a 3-way handshake to establish a connection. Since QUIC is based on UDP, the connection-establishing process only needs a single packet, which includes TLS. A zero-handshake connection occurs 75% of the time when the client and server have already spoken to one another.

Multiplexing: By multiplexing client and server communication, head-of-line blocking problems that are typical with TCP connections are avoided [7]. This gives QUIC an advantage over TCP along with a few other capabilities (better congestion control, forward error correction). Google has conducted thorough testing and measuring and discovered that it reduces Google search latency by 3.6–8% and YouTube video rebuffers by 15–18%. These figures might appear small but considering that Google has given existing TCP applications a high level of optimization, even a small percentage point change can have a big impact. This accounts for 35% of Google's outflow traffic, which equates to 7–10% of all Internet traffic [12]. The user experience and Internet traffic are significantly impacted by these advancements. QUIC is here to stay, especially given that Google supports it. In the upcoming years, additional significant corporations will also adopt it.

This paper will highlight the benefits of QUIC over TCP. In the parts that follow, the details of the arguments and counterclaims presented will be discussed, along with the difficulties that will encounter when switching from TCP to QUIC and their justification. This paper will support the migration from TCP to QUIC using the recent transition that occurred from HTTP to HTTPS in support of the cost migration.

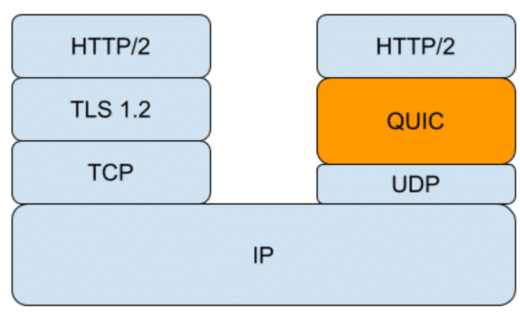

Figure 1: QUIC and TCP layers.

2 QUIC

QUIC used to stand for "Quick UDP Internet Connections," despite the fact that the abbreviation is no longer used. A general-purpose transport layer network protocol created by Google is now referred to as "QUIC." It was initially put into use and made available in 2012. At an IETF meeting in 2013, it was made known to the public as experimentation expanded. A draft of the specification was submitted to the IETF in June 2015 for standardization. A QUIC working group was established in 2016. With the introduction of "HTTP/3" in October 2018, HTTP mapping over QUIC is now destined to become a global standard. The IETF officially standardized it in RFC 9000 in May 2021.

Gaming, streaming media, and VoIP services frequently use UDP, a protocol that is used by Google's QUIC, a low-latency Internet transportation protocol. In exchange for being substantially less resource-intensive than TCP, UDP offers much fewer error-correction functions. Google wants to bring together some of the best aspects of UDP and TCP with cutting-edge security measures via QUIC. Google's QUIC Protocol aims to speed up the web [13].

3 COUNTERCLAIMS

3.1 Cost to upgrade hardware and software: In order to implement QUIC, it is necessary to convert the entire application layer and transport layer to UDP and to create an entirely new server-side and client-side solution (the cost to upgrade hardware and software is more). Small streaming merchants with minimal resources will find this difficult (cost to upgrade for small and medium businesses). In other words, QUIC does not give enough benefits for the cost.

3.2 Recovery of lost packets: Another claim to add to the list is QUIC which is made to perform the same functions as TCP but better. However, it adopts UDP (User Datagram Protocol), a faster internet protocol than TCP but without TCP's technique for recovering lost packets.

3.3 Infrastructure transition from old to new: At the fundamental level of data transmission, it is incredibly difficult to improve the internet. The older infrastructure, which has endured for decades, is used by countless devices, programs, and services.

3.4 Firewalls: It is claimed that QUIC has drawbacks when it comes to firewalls. Firewalls perform crucial security checks using the TCP handshake and SSL session state data. The firewall is unable to determine the status of a session after QUIC encryption setup. Reduced security results from the inability to inspect handshakes and control operations as permitted with TCP/SSL. High-end firewall features are expensive, but they can analyze the initial QUIC session setup with deep packet hardware-based inspection. Using software and central CPU processor resources instead of hardware filter arrays will probably result in performance concerns while updating software on most firewalls.

4 THE ARGUMENT

First claim states that QUIC does not give enough benefits for the cost. A brief case study on migration will, however, address the above-mentioned assertion. Only a small portion of traffic in the beginning of 2012 was HTTPS, and that traffic was exclusively generated by the same internet giants, such as Google and Facebook. There was a significant movement from HTTP to HTTPS in 2015. There has been the shift in favor of HTTPS-based websites in SEO, which is crucial for mid-sized and small enterprises to remain profitable [15]. Small website owners were encouraged to switch to HTTPS once this occurred.

Regarding the second claim, UDP (User Datagram Protocol), which is quicker than TCP but does not have TCP's technique for recovering lost packets, is what QUIC employs. However, QUIC features a faster recovery mechanism than TCP, which is different from it. In fact, QUIC considers the prospective application of Forward Error Correction (FEC) methods. The method used XOR-based FEC, which has the disadvantage that it is ineffective if two or more packets are lost. However, a new IETF draft mentions the usage of FEC to enhance QUIC performance with real-time sessions, contending that FEC renders packet loss recovery indifferent to the round-trip time. This is a promising evolution.

The third assertion claims that it is extremely challenging to improve the internet at the most fundamental level of data transmission. Given the advantages QUIC provides over TCP, replacing all of the current infrastructure would be a challenging one-time investment, but it would be worthwhile.

The fourth argument is based on Firewalls. Connections made using QUIC avoid the proxy since it operates over UDP rather than TCP. By forcing the connection to fall back to TCP when this traffic is blocked, the proxy will be forced to process all incoming web traffic, making it impossible to get around filtering. However, the protocol or the technology itself is not the problem here. The supposed benefit of QUIC is that it speeds up and enhances the efficiency of web communications [14]. The issue is that security equipment like firewalls does not yet support it. Therefore, if the firewall team establishes a common rule to implement and support QUIC protocol, the firewall issue can be easily rectified.

5 Why QUIC?

Unlike TCP, the QUIC protocol only permits communications in encrypted form. Since QUIC explicitly forbids unencrypted communication modes, privacy and security are fundamental components of QUIC data transfers. This is unquestionably a benefit for cybersecurity, but it could also be an unnecessary burden if encryption is not really necessary. But the real breakthrough, however, may be seen in how much quicker [3] QUIC can create a secure connection than TCP + TLS.

In other words, the major objective of QUIC is to significantly lower the overhead during connection creation. This is made possible by the way QUIC was created. In fact, QUIC speeds up the first handshake phase by facilitating the exchange of configuration keys and supported protocols. Particularly, when a sender establishes a connection, the answer packet also contains the information required for any subsequent packets that must use encryption. The need to establish the TCP connection and then negotiate the security protocol through additional packets is removed by this step[4].

Zero RTT Connection Establishment[11], often known as better connection speeds and a considerable reduction in reaction time down to 0ms during inter-host reconnection. As shown in figure 3, it typically takes two or three round trips for a secure TCP connection before the sender can actually begin receiving data [5,12]. This could take 300ms or longer. While employing QUIC, the sender can begin speaking to a recipient right away with whom it has already spoken. Because it has some TCP characteristics that UDP lacks, like congestion control and automated re-transmission, QUIC prevails over UDP in this comparison.

It is significantly more dependable as a result than pure UDP. In more detail, loss recovery is a part of QUIC, even if UDP serves as its foundation. This is because QUIC operates similarly to TCP and verifies each stream separately before retransmitting missing data. Additionally, QUIC can continue serving other flows independently if an error happens in one flow. This feature can be highly helpful for enhancing speed on link that are prone to errors because a significant amount of additional data may be received before TCP detects a lost or missing packet. This data can be processed without restriction in QUIC while the flow is being fixed.

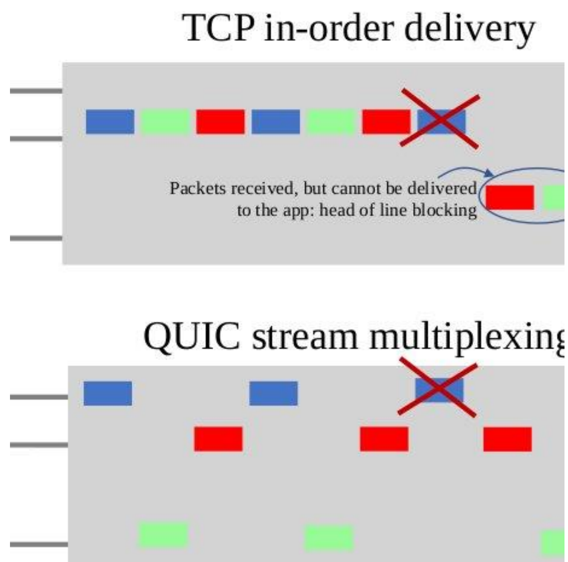

Figure 2: Elimination of the Head-of-Line Blocking Problem. With QUIC, only the stream that is experiencing the packet loss is put to a halt; packets in other streams will keep flowing.

In QUIC, the Head-of-Line blocking problem is eliminated by introducing streams also at the transport layer [7]. If a packet is lost in one of the streams, only that stream will be put to a halt until the packet is recovered, while packets will keep flowing on the other streams [8] as seen in Figure 2. This is one reason that makes QUIC more robust to packet loss than TCP [6].

6 QUIC in IPv4

Figure 4 depicts the expansion of QUIC support up to September 2017; during this period, the number of IPs tripled, and we can see a very active version space. Many updated versions came out, while others became obsolete. A closer look reveals that versions like Version 35 are supported throughout the entirety of our observations. Also, 53% of these hosts are run by Google, with the remaining 40% or so by Akamai. However, starting in September 2017, circumstances have altered. When QUIC support in IPv4 dramatically increased after Akamai's official announcement of its rollout (Figure 5).

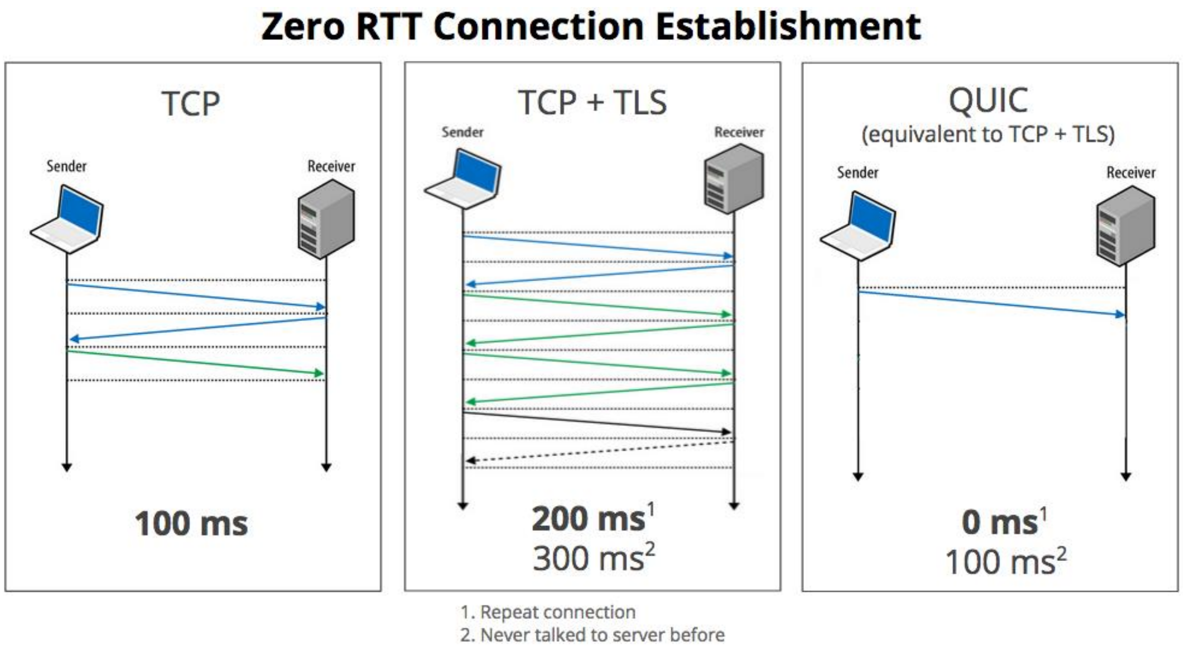

Figure 3: TCP vs. TCP + TLS vs. QUIC
Source: Google

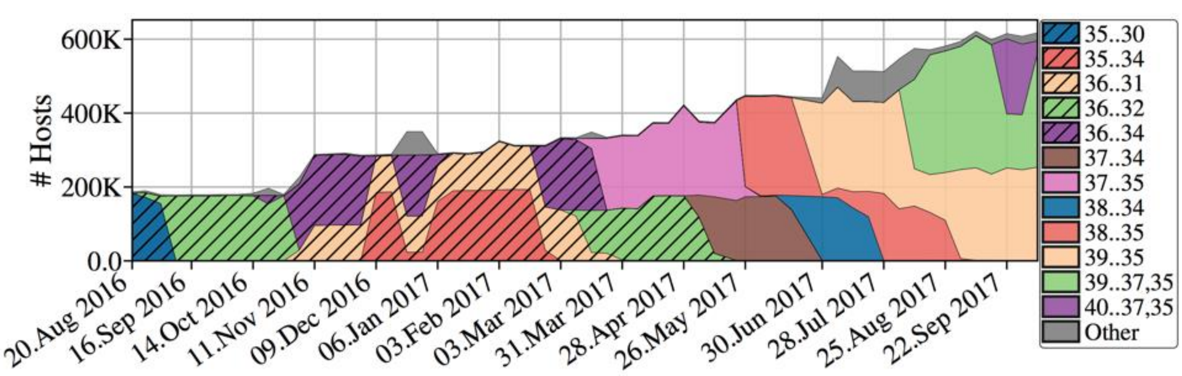

Figure 4: The number of QUIC-capable IPs and support for sets of certain QUIC versions. Here we display versions when there was support by at least 20,000 hosts. Versions that first appeared in 2016 are hatched.
Source: APNIC.

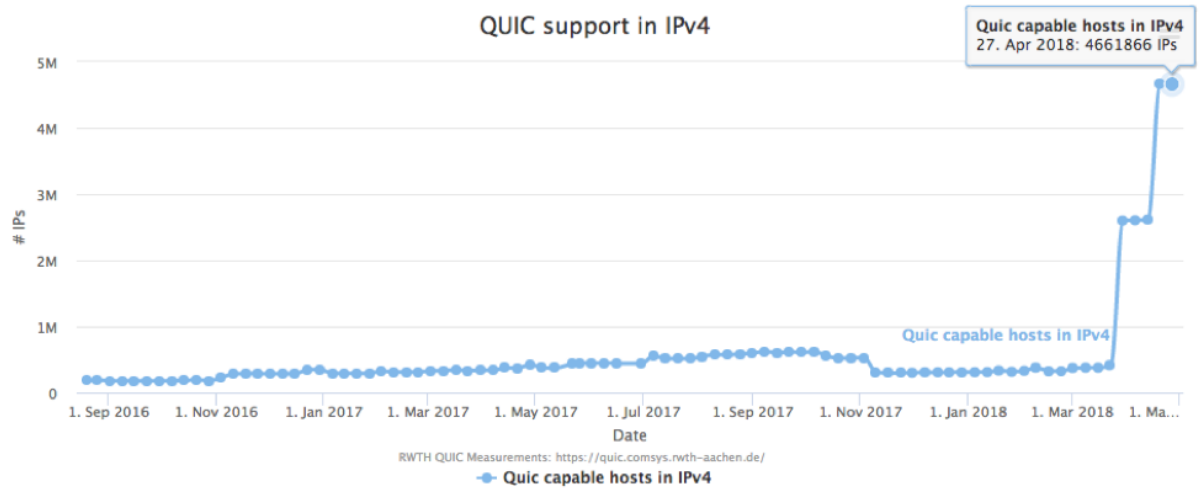

Figure 5: Akamai has stepped up their QUIC support since March 2018. We now observed more than 4.6M IPs supporting QUIC, most hosted by Akamai.
Source: APNIC.

7 QUIC in 5G network

Use of QUIC in 5G core network: In the service-based architecture of the 5G core network, QUIC is a preferable alternative to TCP. According to the majority of the literature, QUIC's faster connection setup than TCP accounts for much of its lower latency. QUIC reduces average request latency to less than half that of TCP for most scenarios using the optimized long-lived connection paradigm, which, while crucial in general, is not the main component in obtaining a much lower latency than TCP. This suggests that QUIC would perform noticeably better than TCP, not just when using the short-lived connection model but also when using the long-lived connection model [9]. The majority of the literature also claims that QUIC performs well with high PLR (Packet Loss Rate), high RTT, and little bandwidth in unfavorable network conditions. However, even in extremely advantageous network conditions with PLR of 0% and RTT of 20 ms [9], QUIC decreases latency to a third of that of TCP and doubles the throughput.

8 CONCLUSION

Google created QUIC to replace a number of out-of-date standards. The stream multiplexing method at the transport layer, which boosts the protocol's throughput performance and successfully resolves the TCP HOL blocking issue, is where QUIC's major advantage lies. This is why it's so important to understand it. QUIC is on its way, a new Internet performance protocol that, when enabled by a browser, can triple the performance.

Even though QUIC does not give enough benefits for the cost, a brief case study on migration will address the above-mentioned concern. Only a small portion of the traffic at the beginning of 2012 was HTTPS, and that traffic was exclusively generated by the same internet giants, such as Google and Facebook. There was a significant movement from HTTP to HTTPS in 2015. There has been a shift in favor of HTTPS-based websites in SEO, which is crucial for mid-sized and small enterprises to remain profitable [15]. Small website owners were encouraged to switch to HTTPS once this occurred.

9 REFERENCES

[1] Carlucci, G., De Cicco, L. and Mascolo, S. (2015) HTTP over UDP: An Experimental Investigation of QUIC. Proceedings of the ACM Symposium on Applied Computing, Salamanca, 13-17 April 2015, 609-614. https://doi.org/10.1145/2695664.2695706

[2] Gagliardi, E. and Levillain, O. (2020) Analysis of QUIC Session Establishment and A. I. Kyratzis, P. G. Cottis DOI: 10.4236/cn.2022.14100222 Communications and Network Its Implementations. In: Information Security Theory and Practice, Springer, Berlin, 169-184. https://doi.org/10.1007/978-3-030-41702-4_11

[3] Megyesi, P., Krämer, Z. and Molnár, S. (2016) How Quick Is QUIC? 2016 IEEE International Conference on Communications, ICC 2016, Kuala Lumpur, 23-27 May 2016, 1-6. https://doi.org/10.1109/ICC.2016.7510788

[4] Nepomuceno, K., et al. (2018) QUIC and TCP: A Performance Evaluation. 2018 IEEE Symposium on Computers and Communications (ISCC), Natal, 25-28 June 2018, 45-51. https://doi.org/10.1109/ISCC.2018.8538687

[5] Yong Cui et al. “Innovating Transport with QUIC: Design Approaches and Research Challenges”. In: IEEE Internet Computing 21.2 (Mar. 2017), pp. 72–76. issn: 1941-0131. doi: 10.1109/MIC.2017.44.

[6] Sarah Cook et al. “QUIC: Better for What and for Whom?” In: 2017 IEEE International Conference on Communications (ICC). 2017 IEEE International Conference on Communications (ICC). May 2017, pp. 1– 6. doi: 10.1109/ICC.2017.7997281.

[7] Martin Thomson and Jana Iyengar. QUIC: A UDP-Based Multiplexed and Secure Transport. url: https://tools.ietf.org/html/ draft-ietf-quic-transport-27 (visited on 04/27/2020).

[8] Draft-Ietf-Quic-Http-27 - Hypertext Transfer Protocol Version 3 (HTTP/3). url: https://tools.ietf.org/html/draft-ietf- quic-http-27 (visited on 04/27/2020).

[9] PETHRUS GÄRDBORN QUIC a Better Choice than TCP in the 5G Core Network Service Based Architecture? https://www.diva-portal.org/smash/get/diva2:1520258/FULLTEXT01.pdf

[10] Yu, Y., Xu, M. and Yang, Y. (2017) When QUIC Meets TCP: An Experimental Study. 2017 IEEE 36th International Performance Computing and Communications Conference (IPCCC), San Diego, 10-12 December 2017, 1-8. https://doi.org/10.1109/PCCC.2017.8280429

[11] Introducing Zero Round Trip Time Resumption (0-RTT). https://blog.cloudflare.com/introducing-0-rtt

[12] Antonello Zanini, Googles QUIC Protocol Replace TCP https://levelup.gitconnected.com/will-googles-quic-protocol-replace-tcp-6ed991a0ca1e

[13] Frederic Lardinois. Google Wants to Speed Up the Web With Its QUIC Protocol 2015 https://techcrunch.com/2015/04/18/google-wants-to-speed-up-the-web-with-its-quic-protocol/

[14] Stephen Shankland, Quic gives the internet's data transmission foundation a needed speedup 2021https://www.cnet.com/tech/computing/quic-gives-the-internets-data-transmission-foundation-a-needed-speedup/

[15] Erik Newton,” HTTP vs HTTPS and SEO in 2019”, 2019, *https://www.brightedge.com/blog/http-https- and-seo*