

QUIC vs TCP: A Performance Evaluation over LTE with NS-3

Apostolos I. Kyratzis, Panayotis G. Cottis

School of Electrical and Computer Engineering, National Technical University of Athens, Athens, Greece

Email: apostolos@kyratzis.gr

How to cite this paper: Kyratzis, A.I. and Cottis, P.G. (2022) QUIC vs TCP: A Performance Evaluation over LTE with NS-3. *Communications and Network*, 14, 12-22. <https://doi.org/10.4236/cn.2022.141002>

Received: October 29, 2021

Accepted: December 13, 2021

Published: December 16, 2021

Copyright © 2022 by author(s) and Scientific Research Publishing Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

In the vast majority of mobile applications, the Transmission Control Protocol (TCP) is still leveraged at the transport layer of the Internet's protocol stack. But, in many cases, the performance of TCP over mobile networks has been proven sub-optimal in practice, thus causing substantial bottlenecks. Quick UDP Internet Connections (QUIC) is a new protocol, currently being standardized by the Internet Engineering Task Force (IETF), that aims at solving some of the inherent problems of TCP. The purpose of this paper is to provide a comprehensive overview of QUIC and compare the performance of QUIC and TCP in wireless networks. To compare QUIC with TCP under various transmission scenarios over LTE networks, the ns-3 network simulator has been employed. The simulations performed showed that 1) under good or average transmission conditions, QUIC is characterized by a better steady state throughput at the same time achieving quite lower file download times; and 2) under poor transmission conditions, the two protocols exhibit a similar performance.

Keywords

Transport Protocol, TCP, QUIC, LTE, Simulation, NS-3

1. Introduction

Ever since its creation, the Internet has experienced an impressive rise both in the number of users base and the volume of data traffic. According to a report issued by Cisco [1], this trend is expected to continue in the following years. The report foresees that 1) the volume of IP traffic will be 235 Exabytes per month in 2021, that is three times the corresponding traffic in 2016; and 2) the number of Internet users is expected to reach 5.3 billion by the end of 2021.

To support the rapidly increasing mobile and desktop traffic over the Internet,

changes are required both in network infrastructure and communication protocols. The majority of web applications today leverage TCP and Hyper Text Transport Protocol (HTTP) to perform functionalities, such as connection establishment, reliable data transmission and congestion control. However, the original design of TCP and HTTP did not predict the increasing complexity of modern web pages and the dynamic nature of Internet traffic. Consequently, the TCP and HTTP protocols currently constitute a major performance bottleneck in the modern Internet. In wireless networks, particularly, these protocols have an adverse effect on web access times, thus deteriorating the Quality of Experience (QoE) offered to the users.

To address this challenge, the Internet research community has developed an intense interest in the design and implementation of novel transport protocols. In 2013, Google introduced QUIC (Quick UDP Internet Connections), a transport protocol that aims at improving throughput performance and reducing page load times for web applications [2]. Since 2016, QUIC is undergoing a standardization process by the IETF. QUIC incorporates new features that overcome the shortcomings of TCP, while also maintaining the functionalities offered by TCP, such as retransmissions, acknowledgements and congestion control. First of all, QUIC runs on top of the User Datagram Protocol (UDP), that is QUIC packets are encapsulated in UDP datagrams. This is necessary because many middlebox devices across the Internet process packets originating only from TCP and UDP. Moreover, QUIC is not integrated in the kernel space of the Operating System (OS) but, instead, it is developed as an application-layer transport protocol enabling rapid adoption as any protocol updates require changes only in the application code and not in the OS kernel. Furthermore, QUIC extends the application layer stream mechanism to the transport layer by mapping HTTP streams to independent QUIC streams, effectively solving the TCP Head of Line (HOL) problem. Finally, QUIC provides improved packet encryption and reduced connection establishment times.

Though its standardization by IETF is still ongoing, QUIC is already supported by all Google services and has been deployed to a number of commercial products, notably the Google Chrome browser. As of March 2021, QUIC is supported by approximately 5% of all websites on the Internet [3]. Therefore, given the rising popularity of QUIC, there has been a number of studies that have analyzed its new features [4] [5] and have compared QUIC and TCP performance [6] [7]. However, only a limited number of studies have examined the performance of QUIC in the framework of mobile networks [8] [9]. In this paper, we utilize an implementation of QUIC for the ns-3 [10] in an attempt to explore the performance of QUIC over LTE networks and compare it to that of TCP.

QUIC is a novel transport protocol proposed by Google that aims at replacing TCP in the near future. IETF has undertaken the task of standardizing QUIC and regularly publishes Internet Drafts in order to document the required func-

functionalities and other relevant information for QUIC implementations [11].

The protocol stack configuration of a secure HTTP connection over QUIC or TCP is shown in **Figure 1**. Though QUIC shares common features with TCP, the two protocols exhibit significant differences in various functionalities. QUIC is developed as an application-layer transport protocol that utilizes UDP as a sub-layer. The reasoning behind this approach is that the deployment or the upgrade of transport protocols in the kernel space of the OS is a very difficult and time-consuming task. In contrast, any required changes at the application code of an application-layer protocol, even substantial ones, can quickly be tested and distributed. As such, QUIC will allow for rapid and wide-scale protocol deployment, updates and optimizations in the future.

As stated earlier, QUIC runs on top of UDP. However, as a “best effort” protocol, UDP does not offer reliable data transmission neither guarantees in-order data delivery to the receiver. Therefore, QUIC must also implement a reliable and in-order data delivery mechanism. In a way similar to TCP, QUIC uses acknowledgments (ACK) in order to detect losses of QUIC packets and perform the necessary retransmissions. A key difference, though, is that QUIC uses unique packet numbers to identify each packet that is sent over a QUIC connection. These QUIC packet numbers are monotonically increasing and do not repeat for the lifetime of a QUIC connection. In comparison, TCP uses sequence numbers to identify TCP segments, but the same sequence number may appear more than once over the entire duration of the connection, *i.e.* in case of packet retransmissions. This feature of TCP causes a problem known as ACK ambiguity, that is TCP is unable to differentiate lost packets from packets that are delivered out-of-order. The introduction of unique QUIC packet numbering enables QUIC to differentiate between lost and out-of-order packets, thus allowing for a more accurate estimation of the connection Round Trip Time (RTT). As a result, QUIC offers 1) improved congestion control by reducing the number of unnecessary retransmissions and 2) more efficient detection of and recovery from packet losses.

Another long-standing problem of TCP is that any internet connection is identified by a unique 4-tuple value that contains the client’s and server’s IP addresses and port numbers. However, any change in the underlying IP address or port number of a TCP connection will cause the termination of this connection.

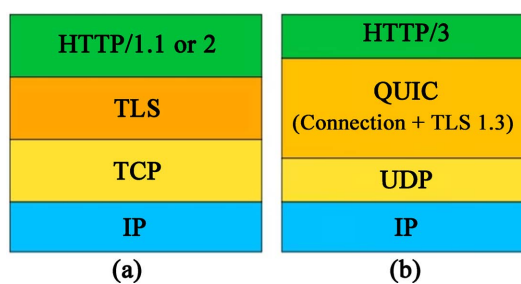


Figure 1. QUIC and TCP protocol stacks. (a) Current configuration; (b) QUIC configuration.

Therefore, TCP connections do not exhibit resilience to any updates of the above parameters that can be initiated by internet middleboxes and/or client mobility in the case of mobile or Wi-Fi networks. QUIC deals with this issue by introducing a Connection ID which is an 8-byte number that uniquely identifies a QUIC connection and is independent from the IP address and UDP port parameters of underlying protocol layers. As such, even if the IP address or UDP port number of a client changes during the communication with a server, the connection ID provides robustness to the QUIC connection which is not terminated.

Moreover, QUIC is designed to offer end-to-end data encryption, which means that both application data and QUIC headers are authenticated and encrypted over a QUIC connection. As illustrated in **Figure 1**, in the case of TCP the TLS protocol runs on top of the TCP layer. In the case of QUIC, however, the cryptographic functionalities of the TLS protocol are directly integrated into the QUIC layer of the protocol stack. This feature allows QUIC to minimize the latency introduced during the initial connection establishment phase. In the case of TCP, two handshakes must be completed before the transmission of application data: first the TCP protocol handshake and then the TLS cryptographic handshake. The added latency is equal to either 3-RTT or 2-RTT, depending on which version of the TLS protocol is used, TLS 1.2 or TLS 1.3, respectively. Instead, QUIC embeds the necessary cryptographic parameters into the first QUIC packet that is sent from the client to the server, thus reducing the connection establishment latency to 1-RTT. In the special case where the client and the server have communicated again in the past, stored cryptographic information can be used to perform a 0-RTT handshake, where encrypted application data are encapsulated into the first QUIC packet that is sent. The 0-RTT handshake represents the optimal scenario in terms of connection establishment latency.

Finally, QUIC implements a stream multiplexing mechanism at the transport layer, which is similar to that of HTTP/2 at the application layer. HTTP/2 maps each different web object to an independent application-layer data stream, meaning that if one stream loses data due to packet loss, the other streams can continue to deliver data to the application. However, TCP is not able to differentiate application data streams at the transport layer, an issue that is identified as the Head-of-Line (HOL) blocking problem. When a TCP connection experiences a packet loss, TCP blocks all application-layer streams from delivering their data until the lost packet is recovered, even if the lost packet contains data from a single application stream which is independent from other streams. Therefore, the HOL blocking problem represents a significant adverse factor for HTTP performance. QUIC solves this issue by extending the application stream mechanism to the transport layer, by mapping one-to-one application data streams to independent QUIC data streams that can be transmitted concurrently over the same QUIC connection. Consequently, even if a packet is lost, QUIC can identify the QUIC streams whose data were encapsulated in the lost packet and, thus, block only the affected application-layer streams. All the other streams that

are unaffected from the packet loss can continue to deliver data in-order to the application at the receiver. Consequently, the performance of HTTP is significantly improved.

Based on simulations employing the ns3 network simulator, this paper aims at comparing QUIC with TCP with respect to basic performance criteria such as the steady state throughput, the fairness in distributing the available bandwidth to competing connection flows and the file download time. To the best of the authors' knowledge, such a simulation based quantitative attempt has not been performed so far.

The rest of the paper is organized as follows. Sec. 2 presents the simulation setup and the scenarios considered for the performance evaluation of QUIC and TCP. The analysis of the simulation results is provided in Sec. 3. Finally, Sec. 4 concludes the paper.

2. Simulation Scenarios

The simulations aim at investigating the performance differences of the TCP and QUIC protocols over LTE networks. In this course, the Network Simulator—3.29 was utilized [12], which allows for the implementation of various communication scenarios between nodes under various network architectures and topologies. For the LTE part of the networks examined, the LENA (LTE-EPC Network simulator) module of ns-3 has been employed, which provides an accurate representation of the LTE protocol stack and Evolved Packet Core (EPC) services, with the sole limitation that the simulated network has a single Serving & Packet Gateway (S/P-GW) node.

The architecture of the simulation model is depicted in **Figure 2**. Across each link involved, the maximum transmission rate (link capacity) and link delay is given. All queues are DropTail queues with a maximum length of 100 packets. The S/P-GW node is connected with point-to-point links (P2P) to two remote QUIC and TCP servers. Each User Equipment (UE) device uses either the TCP or QUIC protocol and the generated traffic is routed towards the corresponding

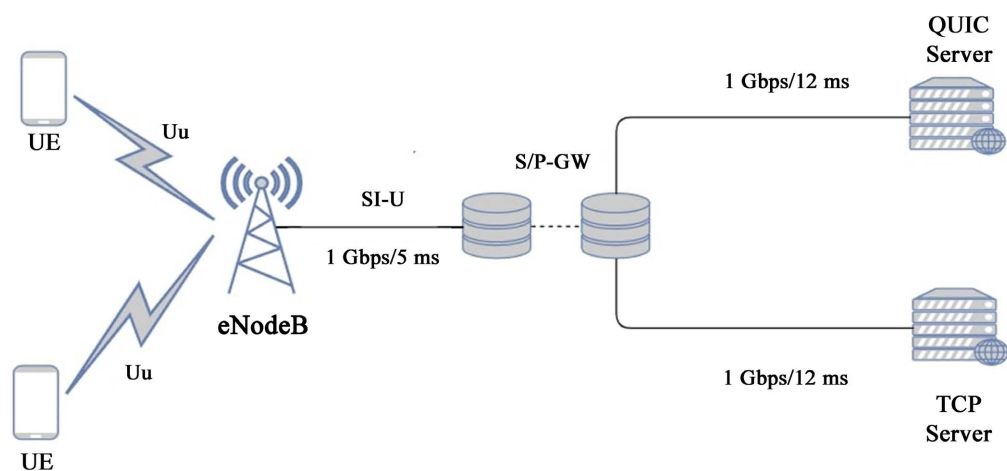


Figure 2. The simulation model.

remote server. Moreover, default values are given to the basic parameters of the Radio Access Network (RAN) of the LTE component. LTE networks provide multiple frequency configurations for the various LTE Bands used. Without loss of generality, LTE Band 4 was selected for the simulations. Similarly, as the LTE system bandwidth can either be 5, 10, 15 or 20 MHz, the maximum value of 20 MHz was selected. That is, these values were not determined based on a specific rationale. The numerical results revealed that this choice does not affect the general outcome of the simulations. The propagation losses are assumed to follow the logarithmic model while the wireless channel fading losses follow the Rayleigh distribution. The transmission buffer of the Evolved Node B (eNB) is set at 512 kB. The transmission power of the eNB and UEs are set at 46 dBm and 23 dBm respectively, which are representative values of a typical macro cell site configuration [13].

The main target of this paper is to investigate the possible performance gains offered by the new features of QUIC compared to TCP, when data are transmitted over a wireless channel.

2.1. Steady State Throughput

The first simulation scenario studies the steady state throughput achieved by QUIC and TCP. Two separate UEs are placed at equal distance from the eNB, one of which communicates with the TCP server and the other with the QUIC server. To examine the performance of the protocols in the case of constant data flow in the downlink (DL) direction, the BulkSend application of ns-3 is utilized, according to which each remote server continuously sends 512 bytes of data to the corresponding UE. The packet loss ratio (PLR) of the P2P links is set at a low value of 0.5%. The duration of the simulation is set at 40 s and is repeated varying the distance between the UEs and the eNB. The results obtained are presented and discussed in Sec. 4.

2.2. Flow Fairness

The second simulation scenario examines the fairness principle with which any transport protocol must comply as to the fair distribution of the available bandwidth to competing connection flows. If fair bandwidth allocation is not ensured, a connection flow that uses an “unfair” protocol will be able to acquire a disproportional amount of the available resources and, thus, achieve better performance compared to other competing flows. It should be noted that in wired networks it has been shown that QUIC is unfair towards TCP, even when the two protocols employ the same congestion control algorithms [8]. To investigate the fairness issue arising between QUIC and TCP in LTE networks, the BulkSend applications of ns-3 are again utilized employing the same parameters as in Sec. 3. A, but assuming varying numbers of QUIC and TCP flows. The distance between each UE and the eNB is fixed at 250 m. The results obtained are presented and discussed in Sec. 4.

2.3. File Downloads

The third simulation scenario investigates the file download performance of QUIC and TCP. The distance between the UEs and the eNB is again set at 250 m. Five file sizes in the range 64 kB - 2 MB have been considered. The file download time (DLT) has been selected as the relevant performance metric. The results obtained are presented and discussed in Sec. 4.

2.4. Effect of QUIC Streams on Throughput

As analyzed in Sec. 2, one of the novelties of QUIC is the extension of the HTTP stream mechanism to the transport layer by introducing the QUIC stream mechanism towards solving the HOL blocking problem of TCP. To quantify the effect of QUIC streams on throughput over LTE networks, an ns-3 QUIC application is used. The QUIC application sends 512 bytes of data every 300 μ s from the QUIC server to one UE. The simulation is repeated varying the number of QUIC streams and distance between the UE and the eNB. The results obtained are presented and discussed in Sec. 4.

3. Simulation Results and Discussion

In this section, the results obtained under the ns-3 simulation scenarios described in Sec. 3 are presented and discussed.

3.1. Steady State Throughput

Table 1 presents the simulation results for the steady state throughput of the TCP and QUIC protocols when the distance between the UEs and the eNB varies. In order to exclude from the measurements the slow-start phase of the connection establishment, the steady state is defined as the simulation duration from 5 s to 40 s.

From **Table 1** it is deduced that the performance gains offered by QUIC in wired networks are also observed in LTE networks, despite the adverse characteristics of wireless channels that deteriorate the operation of a transport protocol. However, as the distance from the eNB increases that causes reduction in the received signal quality at the UE, the throughput of QUIC becomes comparable to the throughput of TCP. A possible explanation behind this phenomenon is given at [14], where it was identified that, in wired networks, QUIC exhibits

Table 1. Average TCP and QUIC throughput.

Distance from eNB (m)	Avg. TCP Throughput (Mbps)	Avg. QUIC Throughput (Mbps)
250	2.7504	5.7696
750	2.8855	5.4658
1500	2.9941	3.8316
2500	1.5284	1.3247

worse performance than TCP when packet re-ordering is necessary. When the received signal quality over a wireless channel is reduced, severe packet re-ordering is necessary to deal with the adverse effect of fast fading losses. Therefore, our simulation results indicate that packet re-ordering decreases QUIC performance also in the case of wireless networks. Overall, under good wireless channel conditions, QUIC outperforms TCP, but, as the channel conditions become worse, the two protocols exhibit a similar throughput performance.

3.2. Flow Fairness

The simulation results examining the fairness between competing QUIC and TCP flows are summarized in **Table 2**.

In the case of two competing QUIC flows, they exhibit similar throughput and therefore are fair to each other. Next, the co-existence of one QUIC flow with two and five TCP flows is examined. The two and five TCP flows in each case examined achieve approximately the same throughput as denoted by the low relevant standard deviation. This means that the TCP flows are fair with regard to each other. But, the single QUIC flow outperforms the TCP flows though both protocols use congestion control algorithms of similar features. Finally, the case of two QUIC coexisting with two TCP flows is examined. As denoted by the relevant low standard deviation values, fairness is observed between the two QUIC flows as well as between the two TCP flows examined. It is also observed that each QUIC flow acquires almost double the bandwidth compared to that acquired by each TCP flow. It is evident that, though the two protocols exhibit an individual internal fairness, in LTE networks QUIC connections outperform TCP ones with regard to the throughput acquired, thus causing a severe lack of fairness against TCP.

3.3. File Downloads

Table 3 summarizes the simulation results concerning the file download performance of the QUIC and TCP protocols.

The simulation results show that QUIC achieves quite lower download times for all the files sizes considered. This performance gain of QUIC over TCP is

Table 2. Throughput of competing flows.

Flows	Protocol	Throughput (Mbps)	Std. Dev.
2 QUIC	QUIC	5.3852	0.37
1 QUIC vs 2 TCP	QUIC	5.9790	-
	TCP	2.5713	0.12
1 QUIC vs 5 TCP	QUIC	6.3422	-
	TCP	2.7768	0.52
2 QUIC vs 2 TCP	QUIC	5.2883	0.50
	TCP	2.4048	0.01

reduced as the file size increases. Additionally, it should be taken into consideration that QUIC allows endpoints that have communicated before in the past, to establish a connection with a 0-RTT handshake. This mechanism speeds up the transmission of application data by at least 1-RTT compared to TCP. When the QUIC 0-RTT handshake was enabled in the examined simulation scenario, the connection establishment time was reduced from 72 ms to 22 ms for all file sizes examined. Given that approximately 40% of all HTTPS connections are reused connections [15], it is clear that QUIC will significantly enhance the user experience in Web Browsing applications.

3.4. Effect of QUIC Streams on Throughput

The simulation results concerning the effect of the QUIC streams mechanism on throughput are presented in **Table 4**.

Overall, the introduction of stream multiplexing at the transport layer improves the throughput performance of QUIC. As intuitively expected, the steady state throughput is higher when the number of QUIC streams served by the connection is increased. However, the gain in throughput is reduced as the number of QUIC streams increases. Moreover, as the distance between the UE and the eNB increases, the throughput gain resulting from the use of QUIC streams gets lower. In conclusion, the use of independent QUIC streams for the transmission of application data deals effectively with the Head of Line Blocking problem caused by TCP, especially in LTE networks where the HOL problem occurs frequently due to random bandwidth fluctuations in the wireless channel.

Table 3. File download times for QUIC & TCP.

File Size	DLT TCP (s)	DLT QUIC (s)	Reduction (%)
64 kB	0.419	0.323	-22.9
256 kB	0.566	0.412	-27.2
512 kB	0.674	0.539	-20.0
1 MB	0.768	0.66	-14.1
2 MB	0.926	0.884	-4.5

Table 4. Effect of QUIC streams on throughput.

Distance from eNB	Number of QUIC Streams				
	1	2	4	8	16
	Throughput (Mbps)				
250 m	4.34	4.92	5.15	5.6	5.94
750 m	3.9	4.6	4.7	5.11	5.5
1500 m	2.65	3.14	3.30	3.37	3.78
2500 m	1.40	1.49	1.95	2.13	2.18

4. Conclusions

The key findings of the simulation experiments can be summarized as follows. 1) Under good or average received signal quality at the UE, the QUIC protocol has a better steady state throughput performance compared to TCP. However, when the received signal quality at the UE is poor, the two protocols have a similar performance. 2) QUIC connections acquire a larger portion of the available bandwidth compared to TCP ones. This indicates that fairness between TCP and QUIC does not exist, even when the radio resources at the eNB are equally distributed to all the UEs. 3) QUIC demonstrates quite lower file download times compared to TCP, for all file sizes examined. 4) The 0-RTT handshake feature of QUIC greatly reduces the connection establishment times. 5) The biggest benefit of QUIC lies in the stream multiplexing mechanism at the transport layer, which increases the protocol's throughput performance and effectively counters the HOL blocking problem of TCP.

In future research, it is worth investigating potential optimization schemes for QUIC over wireless networks. As QUIC encrypts all the information contained in the QUIC packet headers, all the TCP accelerator mechanisms that are currently deployed in modern wireless networks will have to be properly modified towards serving and optimizing mobile QUIC traffic. Moreover, it is important for network providers to exploit specific characteristics of QUIC (such as the Spin Bit [16]) towards monitoring the status of QUIC connections within their networks. Furthermore, an interesting further research could be to investigate how QUIC affects the performance of scheduling algorithms in LTE networks [17]. Finally, an important field of future research could be the development of novel congestion control algorithms that can optimally utilize the new characteristics of QUIC.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

- [1] Cisco (2020) Cisco Annual Internet Report (2018-2023). 1-41.
- [2] Langley, A., *et al.* (2017) The QUIC Transport Protocol: Design and Internet-Scale Deployment. *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*, Los Angeles, 21-25 August 2017, 183-196. <https://doi.org/10.1145/3098822.3098842>
- [3] Usage Statistics of QUIC for Websites, March 2021. <https://w3techs.com/technologies/details/ce-quic>
- [4] 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>
- [5] Gagliardi, E. and Levillain, O. (2020) Analysis of QUIC Session Establishment and

- Its Implementations. In: *Information Security Theory and Practice*, Springer, Berlin, 169-184. https://doi.org/10.1007/978-3-030-41702-4_11
- [6] 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>
- [7] 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>
- [8] Kakhki, A.M., Jero, S., Choffnes, D., Nita-Rotaru, C. and Mislove, A. (2017) Taking a Long Look at QUIC: An Approach for Rigorous Evaluation of Rapidly Evolving Transport Protocols. *Proceedings of the 2017 Internet Measurement Conference*, London, 1-3 November 2017, 290-303. <https://doi.org/10.1145/3131365.3131368>
- [9] 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>
- [10] De Biasio, A., Chiariotti, F., Polese, M., Zanella, A. and Zorzi, M. (2019) A QUIC Implementation for Ns-3. *Proceedings of the 2019 Workshop on Ns-3*, Florence, 19-20 July 2019, 1-8. <https://doi.org/10.1145/3321349.3321351>
- [11] Iyengar, J. and Thomson, M. (2021) QUIC: A UDP-Based Multiplexed and Secure Transport. No. RFC 9000. Internet Engineering Task Force, 1-151. <https://doi.org/10.17487/RFC9000>
- [12] ns-3|A Discrete-Event Network Simulator for Internet Systems. <https://www.nsnam.org>
- [13] Nguyen, B., et al. (2014) Towards Understanding TCP Performance on LTE/EPC Mobile Networks. *Proceedings of the 4th Workshop on All Things Cellular: Operations, Applications, Challenges*, Chicago, 22 August 2014, 41-46. <https://doi.org/10.1145/2627585.2627594>
- [14] Kakhki, A.M., Jero, S., Choffnes, D., Nita-Rotaru, C. and Mislove, A. (2019) Taking a Long Look at QUIC. *Communications of the ACM*, **62**, 86-94. <https://doi.org/10.1145/3330336>
- [15] Introducing Zero Round Trip Time Resumption (0-RTT). <https://blog.cloudflare.com/introducing-0-rtt>
- [16] Kühlewind, M. and Trammell, B. (2021) Manageability of the QUIC Transport Protocol. Internet Engineering Task Force, 1-34.
- [17] Kuboye, B.M. (2018) Performance Evaluation of Scheduling Algorithms for 4G (LTE). *Communications & Networks*, **10**, 152-163. <https://doi.org/10.4236/cn.2018.104013>