# Assessment of Multipath QUIC Protocol Performance: A Survey

Abhishek Kittur, Sujay Sreedhar, Pranav Vishnu Anandakumar, Ashish Ajayakumar
*Department of Electrical and Computer Engineering*
*North Carolina State University*
Raleigh, NC 27606
(amkittur, ssreedh, aaja, pananda)@ncsu.edu

***Abstract* - The Multipath QUIC (MPQUIC) protocol is an extension of the existing QUIC transport protocol used by Google, which is standardized by the IETF. As Multipath QUIC is not yet standardized, it presents an opportunity to conduct further experiments on the use of multiple network paths between senders and receivers by dividing data packets into multiple streams. This paper presents an extensive survey, synthesizing research from a variety of studies focused on the development and functionalities of QUIC, MPTCP, and ultimately MPQUIC. It aims to illuminate the evolutionary path leading to MPQUIC's current capabilities and potential applications. In addition to a thorough literature review, this paper introduces a experimental framework designed to assess the performance of MPQUIC. The experimental setup is composed of two Virtual Machines (VMs) configured to operate within a 5G mmWave environment, a cutting-edge wireless technology known for its high bandwidth and speed. This setup is meticulously designed to test the resilience and efficiency of MPQUIC, particularly in challenging conditions such as Non-Line-of-Sight (NLOS) scenarios, where the signal may be impeded by physical obstructions like buildings or walls, simulating realistic urban environments. Through this dual approach of survey and empirical investigation, the paper seeks to offer a comprehensive analysis of MPQUIC's operational strengths and limitations, contributing valuable insights to the field of network protocols and their optimization for next-generation wireless technologies.**

***Keywords* - QUIC, Multipath QUIC, Multipath TCP, 5G mmWave, LOS, NLOS, Head-Of-Line-Blocking**

## I. Introduction

With the increasing demand for mobile broadband services, a new generation of mobile and wireless networks called 5G has been introduced [1]. 5G is designed to support large mobile traffic volumes and will be a critical enabler for the Internet of Things [2]. The fifth-generation technology is expected to deliver higher data rates, lower latency, and improved connectivity than its predecessors [1]. However, the use of higher frequencies and millimeter waves in 5G networks makes them susceptible to signal blocking due to non-line-of-sight obstacles. This can result in packet loss and affect the reliability of data transmission [3].

Transport protocols manage data flow over a network, ensuring efficient, reliable, and secure transmission. In 5G networks, they are crucial for reliable data transmission, but traditional protocols like TCP may not be optimal due to their limitations. TCP is susceptible to head-of-line (HOL) blocking [4], where a single packet loss can cause delays in the transmission of subsequent packets, even if they were not affected by the initial loss. This can lead to uneven network utilization and reduced network efficiency. Therefore, there is a need for innovative transport protocols that can better handle the unique requirements of 5G networks, such as high bandwidth, low latency, and multiple connections.

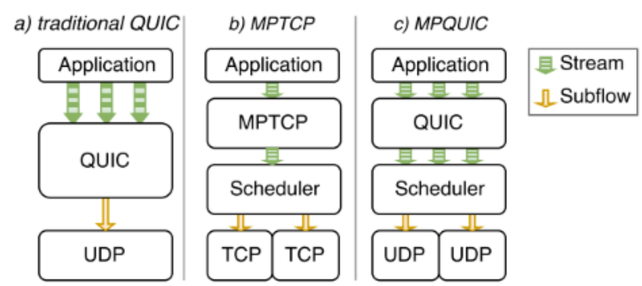


Fig. 1. Network stack comparison of QUIC, MPTCP, and MPQUIC. QUIC multiplexes application streams on a single UDP flow, whereas MPTCP splits a single stream into multiple TCP subflows. MPQUIC combines both features by multiplex- ing application streams on multiple UDP sub-flows.

On the other hand, QUIC (Quick UDP Internet Connections) is a recently introduced transport protocol developed by Google that uses UDP (User Datagram Protocol) instead of TCP as its substrate [5]. QUIC improves web application performance by reducing latency and packet loss. It has inbuilt encryption and supports multiplexing for multiple data streams over a single connection. But with the introduction of 5G networks and most of the devices having more than one network interface, there is a call for multipath transport protocols to pool network resources and provide reliable transmissions. MPTCP [6] extends TCP to allow data transmission over multiple paths, reducing network congestion and improving reliability. It's beneficial for 5G networks with multiple paths, such as small cell networks. In recent years, Multipath QUIC

(MPQUIC) has emerged as a promising transport protocol for 5G networks.

MPQUIC is designed to leverage the capabilities of modern networks, including 5G, by using the concept of multiple paths and streams to transmit data, improving both efficiency and performance [7]. A ”path” refers to a specific network interface that is used for sending and receiving packets, while a ”stream” refers to a logical stream of data that is being transmitted between the client and the server. In addition, MPQUIC includes adaptive congestion control and supports multiple channels for data transmission, improving 5G network reliability by mitigating *Head of Line (HOL) blocking* which is a lingering issue in MPTCP. It also includes mechanisms for path management, and packet reordering to ensure the reliable delivery of data over multiple paths and packet retransmission. MPQUIC is a combination of MPTCP and QUIC that takes advantage of the strengths of both protocols by combining the stream and path capabilities to transmit the traffic as shown in Figure 1. MPQUIC claims to use the multipath capabilities of MPTCP to ensure reliable transmission of packets even in the event of non-line-of-sight blocking [7]. At the same time, it retains the low-latency and high-performance benefits of QUIC. This makes MPQUIC an ideal transport protocol for 5G networks, where reliability and performance are critical. The MPQUIC scheduler is responsible for assigning streams to different paths. Currently, the packet scheduler used in the multipath QUIC implementation is the default lowest RTT scheduler and the congestion control algorithm in use is Opportunistic Linked Increase Algorithm (OLIA).

This report aims to investigate the potential application of MPQUIC within 5G networks, with a focus on ensuring reliable data transmission across multiple channels. It also seeks to assess the performance of MPQUIC under various conditions, particularly situations where one or more channels experience non-line-of-sight blocking, a common challenge in 5G networks. Furthermore, the survey intends to offer an overview of existing research studies related to MPTCP, QUIC, and MPQUIC to evaluate their respective performance in this context.

The remainder of this report is structured as follows: Section II offers an extensive review of existing literature, encompassing the selected papers relevant to our survey topic, providing both a summary and comparison of the various research papers encountered. Section III outlines the specifics of our experimental setup, detailing the tools and code base utilized for implementation. The evaluation of QUIC performance is presented in Section IV. Lastly, Section V serves as the conclusion, encapsulating the findings and their implications.

## II. Related Work

Several studies have investigated the use of multipath protocols for improving the reliability of wireless networks.

1) In [6] Qiuyu Peng *et al* proposes a way to enhance application performance by utilizing multiple paths for TCP connections. The authors propose a fluid model to analyze a broad class of MP-TCP algorithms and explore the equilibrium characteristics such as existence, uniqueness, and stability. Key design criteria are identified to ensure these properties, highlighting how algorithm parameters influence TCP-friendliness, responsiveness, and window oscillation. The paper introduces the Balia algorithm, which generalizes existing MP-TCP algorithms, achieving a balance among TCP-friendliness, responsiveness, and window oscillation. The implementation of Balia in the Linux kernel and comparative analysis with existing MP-TCP algorithms are discussed, demonstrating its performance benefits. The findings provide insights into the structural understanding of MP-TCP algorithms, enabling systematic trade-offs among different properties like TCP-friendliness and responsiveness. This work significantly contributes to the development and optimization of MP-TCP protocols, offering a foundation for future research and implementation in networking systems.

2) In [7] Christoph Paasch *et al* further provides the implementation and evaluation of MPTCP in the Linux kernel, which was the first of its kind a substantial enhancement to the traditional TCP that allows a single TCP connection to utilize multiple paths between two endpoints. This capability aims to improve redundancy, performance, and the efficient use of network resources, addressing the limitations of TCP in environments where devices, such as smartphones and computers, are increasingly multihomed. However, this introduces complexities, particularly in implementing such a system within the existing network infrastructure and ensuring its compatibility with current internet protocols and middleboxes. The authors begin by discussing the evolution of internet devices, noting that many now feature multiple network interfaces, enabling them to connect to the internet via multiple paths. Despite this, the dominant protocol, TCP, traditionally binds a connection to a single interface, thereby underutilizing the potential for increased bandwidth and reliability provided by multiple interfaces. The paper outlines prior attempts at network layer solutions and transport layer enhancements to address multihoming and multipath challenges, including SCTP and various TCP extensions. However, these solutions have not seen widespread deployment, primarily due to compatibility issues with existing infrastructure and the need for application-level changes. MPTCP is introduced as a robust solution designed to operate seamlessly with the existing internet architecture, supporting multiple paths for a single TCP connection without requiring changes to applications or significant updates to network infrastructure. MPTCP's operation is explained through scenarios that illustrate how connections are established, managed, and utilized across multiple paths, leveraging new TCP options to manage connection identifiers and support the dynamic addition and removal of paths.The paper delves into the implementation of MPTCP in the Linux kernel, highlighting the architectural considerations and challenges faced, such as the division between user and interrupt contexts in processing network packets. Key aspects of the implementation include the management of subflows within a connection, data scheduling across these subflows, and the handling of packet reception and data reordering. A

novel architecture comprising master subsockets, a multipath control block (mpcb), and slave subsockets is introduced to manage the complexities of multipath data transfer, including connection establishment, data scheduling, and reception. The authors present a comprehensive performance evaluation, conducted within the HEN testbed, to assess the effectiveness of MPTCP's coupled congestion control mechanism and its overall performance under various conditions. The tests demonstrate that MPTCP, with its coupled congestion control, behaves fairly when sharing a bottleneck link with standard TCP traffic, maintaining equitable bandwidth distribution. Additional experiments investigate the impact of factors like receive buffer size, packet loss ratio, and Maximum Segment Size (MSS) on MPTCP performance, showing that the protocol can efficiently utilize available network paths and adjust to varying network conditions.However, the study raises concerns about MPTCP security. In [8] Mathieu Jadin *et al* through their paper tried to mitigate the problem in [7] with the use of MPTCPsec, a proposed extension for MultiPath TCP that integrates authentication and encryption directly into the protocol. The design utilizes an adaptation of the ENO option, modified for the multipath environment, and the proposed extension is implemented in the Linux kernel's reference implementation of MultiPath TCP.

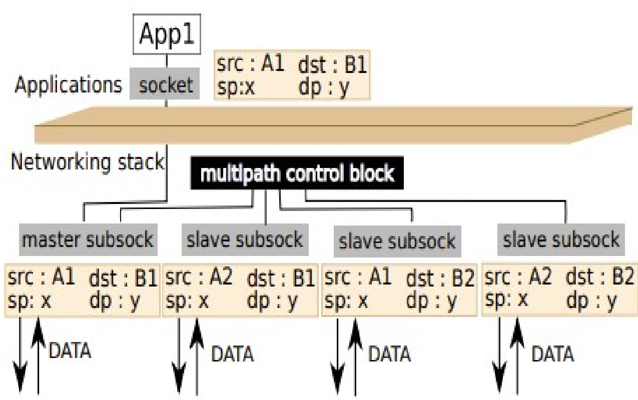


Fig. 2. Overview of the multipath architecture.[7]

3) In addition to the aforementioned studies, there have been several other research efforts aimed at assessing the QUIC protocol. In [5] Adam Langley *et al* introduce QUIC, an encrypted, multiplexed, and low-latency transport protocol designed from the ground up to improve transport performance for HTTPS traffic. QUIC is designed to be secure, multiplexed, and low-latency, aiming to overcome the limitations of traditional protocols like TCP and TLS, particularly in terms of connection establishment time, stream multiplexing, and resilience to packet loss. One of the significant achievements of QUIC is its ability to reduce connection establishment time to zero round-trips for repeated connections between the same client and server, leveraging encrypted transport from the outset to prevent ossification by middleboxes. It operates entirely in user-space over UDP, which not only expedites its deployment and iterative development but also ensures its packets can traverse the majority of the internet's infrastructure without being blocked. The QUIC protocol's design reflects lessons learned from deploying at scale, including the challenges posed by network middleboxes and the complexities of ensuring compatibility and performance across diverse internet conditions. The paper discusses these experiences in detail, including the challenges of determining optimal packet sizes, handling UDP blocking and throttling by some networks, and the decision to remove forward error correction after finding it provided limited benefit. In conclusion, QUIC represents a significant step forward in internet transport protocols, addressing longstanding issues with TCP and TLS that have become more pressing with the growth of secure, latency-sensitive web applications. Its development and deployment by Google demonstrate the feasibility and benefits of introducing a new transport protocol at internet scale, and ongoing efforts to standardize QUIC through the IETF promise to extend these benefits across the broader internet ecosystem.

4) Quentin De Coninck *et al* propose Multipath QUIC, a deployable multipath transport protocol in [9], The authors outline the necessity for multipath communication in modern internet usage, especially with the prevalence of multihomed devices such as smartphones and dual-stack hosts, which can benefit from simultaneous use of multiple network paths (e.g., Wi-Fi and LTE). MPQUIC is designed as an extension to QUIC, a transport layer network protocol initiated by Google, primarily to improve web page load times by reducing latency. QUIC operates over UDP and integrates features from HTTP/2, TLS, and TCP to establish secure, multiplexed, and reliable connections. It encrypts all data and most headers to prevent interference from middleboxes. The paper identifies the lack of multipath support in QUIC as a limitation, noting the success of Multipath TCP (MPTCP) in utilizing multiple network paths for a single connection. The authors propose MPQUIC to allow a QUIC connection to operate over multiple paths simultaneously, aiming to improve data transfer efficiency and provide seamless network handovers. The design of MPQUIC includes new strategies for path identification, reliable data transmission, path management, packet scheduling, and congestion control. Key innovations include the use of path IDs to distinguish different paths within the protocol and modifications to the packet and frame structures to support multipath operation. The implementation is based on an extension of the quic-go, an open-source QUIC implementation in Go. The paper presents a comprehensive evaluation of MPQUIC compared to TCP, MPTCP, and single-path QUIC under various network conditions, using Mininet simulations. The scenarios explored include large and short file downloads and network handovers, with and without packet loss. The results demonstrate that MPQUIC can significantly outperform both single-path QUIC and MPTCP in many scenarios, particularly in environments with packet loss or when utilizing network handovers. MPQUIC shows the ability to aggregate bandwidth effectively and maintain

high performance even in challenging network conditions. The authors suggest future directions for research, including further optimization of MPQUIC for specific network conditions and the exploration of its performance and compatibility in real-world internet environments. They also highlight the importance of integrating native multipath capabilities into QUIC to enhance its utility for modern internet applications. This work contributes significantly to the ongoing development and standardization efforts for QUIC by introducing multipath capabilities, potentially influencing future transport protocols and networking practices.

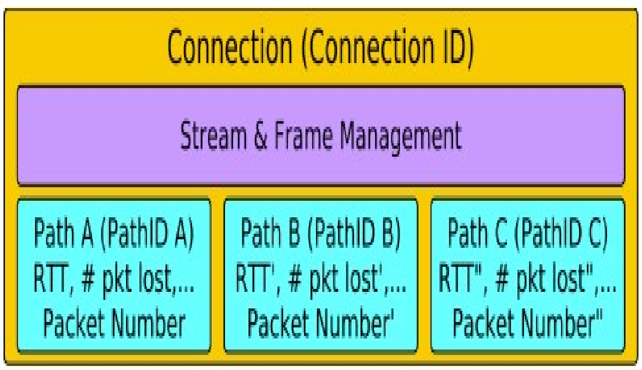


Fig. 3. High-level architecture of Multipath QUIC.[9]

5) Several research efforts have focused on comparing the two multipath protocols. In [10] Tobias Viernickel *et al* evaluate and compare the performance of MPQUIC with traditional QUIC, TCP. The authors aim to showcase how MPQUIC significantly surpasses existing transport protocols like Multipath TCP (MPTCP) by offering advanced features such as fine-grained stream-to-path scheduling, minimized head-of-line blocking, and expedited subflow establishment.The authors present a comprehensive design for MPQUIC that builds upon the strengths of QUIC, including userland deployability, encrypted communication, and stream multiplexing. They detail the protocol's mechanisms for managing multiple network paths, flow and congestion control on a per-subflow basis, and packet scheduling across subflows. Their implementation, based on the open-source quic-go project, demonstrates MPQUIC's viability and performance advantages. The research illustrates MPQUIC's potential to serve as a universal stream transport protocol, capable of optimizing the use of diverse network interfaces in today's internet landscape. The authors suggest that MPQUIC's design benefits, coupled with its compatibility with existing QUIC implementations, position it as a leading solution for future transport protocol development. They also highlight opportunities for further exploration, including optimized scheduling strategies and application-specific adaptations, to fully harness MPQUIC's capabilities. In conclusion, the paper advocates for the adoption of MPQUIC as a superior alternative to existing multipath transport protocols, offering enhanced deployability, security, and performance for a wide range of internet applications.

6) [11] provides a comprehensive comparison of the multipath protocols in the general public Internet and concentrates on throughput and delay for single and multiple parallel flows. The paper starts by highlighting the limitations of traditional Quality of Service (QoS) architectures in addressing the last/first-mile link problems and their inapplicability in non-proprietary networks due to network neutrality laws. It then points out the potential of devices equipped with multiple network interfaces to enhance network performance by utilizing different network paths simultaneously. This multipath approach challenges traditional transport protocols like TCP, which struggles with efficiency in multipath scenarios and suffers from head-of-line blocking, prompting the development of MPTCP and the emergence of MPQUIC. MPTCP and MPQUIC are presented as advancements over TCP, designed to improve data transfer efficiency by leveraging multiple network paths. While MPTCP extends TCP to support multipath operations, MPQUIC is a newer protocol that integrates the advantages of QUIC, MPTCP, and SCTP, offering improved performance and security features. The paper notes MPQUIC's limited operating system support, necessitating its implementation within applications, but highlights its transparency to network devices like firewalls and routers. The evaluation reveals that while multipath protocols increase throughput, they also introduce additional protocol delays, particularly notable with MPTCP. However, MPQUIC exhibits significantly lower delay increases, making it a more suitable choice for applications sensitive to delays. The study identifies head-of-line blocking, both path-related (PHoL) and stream-related (SHoL), as a significant challenge for multipath protocols, with MPQUIC showing better management of HoL blocking than MPTCP. The research concludes that MPQUIC generally outperforms MPTCP in managing protocol delays and throughput, offering a more efficient solution for general-purpose networking across multiple network paths. However, the authors note that the benefits of MPQUIC may not always justify the effort required to implement a new protocol stack and adapt existing applications. The paper suggests that improvements to multipath protocols should focus on mitigating adverse network phenomena like drops and buffer bloat rather than solely addressing their symptoms. For time-sensitive applications, both MPQUIC's Stream and Path schedulers should be optimized to reduce delay further and enhance performance. This detailed comparison sheds light on the practical considerations of deploying MPTCP and MPQUIC in real-world networking scenarios, providing valuable insights for developers and network administrators aiming to optimize network performance across multiple paths.

7) Furthermore, the effectiveness of the multipath protocols in a 5G setup has also been a subject of research. In [12] for example, Mahmud, Imtiaz *et al* provides the performance of MPTCP schedulers and congestion control algorithms (CCAs) in both 4G and 5G networks. The paper focuses on enhancing network performance in scenarios where devices can connect

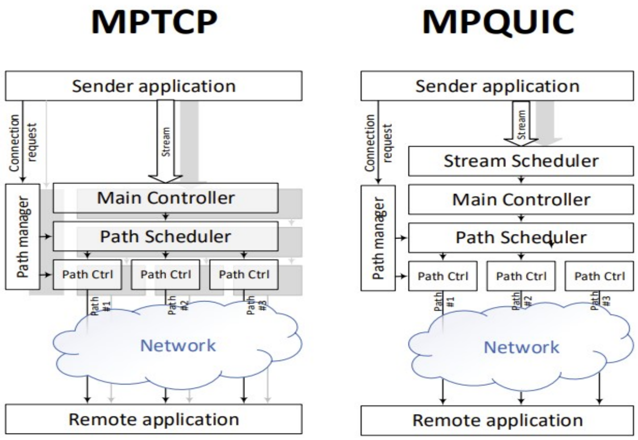


Fig. 4. Comparison of MPTCP and MPQUIC architecture: two streams are transferred in parallel (one has been shaded). [11]

to both 5G and 4G networks simultaneously. This situation is increasingly common due to the uneven deployment of 5G networks and the technological capability of modern devices to support multiple network interfaces. The key challenge addressed is the frequent handoffs between 5G and 4G networks, leading to significant delays and packet losses due to the high susceptibility of 5G signals to obstacles. The introduction outlines the promise of 5G networks in delivering high data rates with low latency and notes the current global efforts in deploying these networks. However, the limited coverage of 5G and its vulnerability to signal blockage by obstacles necessitate frequent switches between 5G and 4G, impacting data transmission quality. Multipath TCP (MPTCP) emerges as a solution, enabling devices to use multiple network paths simultaneously to improve throughput and reliability. The study aims to evaluate the performance of various MPTCP schedulers and congestion control algorithms (CCAs) in a mixed 5G/4G environment, identifying the combinations that offer the best performance. The paper provides an overview of MPTCP, emphasizing its ability to handle multiple data paths through different network interfaces. It discusses various MPTCP schedulers like Shortest RTT First, Round-Robin, Redundant, Earliest Completion First, and Block Estimation, each designed to optimize data flow across multiple paths. Additionally, it reviews CCAs such as Linked Increase Algorithm (LIA), Opportunistic Linked Increase Algorithm (OLIA), Balanced Linked Adaptation (BALIA), and Weighted Vegas, focusing on their approach to balancing load and ensuring fairness across paths. The research utilizes a simulation environment built on NS3-DCE with a 5G and 4G network scenario to evaluate the performance of different MPTCP schedulers and CCAs. This setup aims to reflect real-world conditions, including the movement of user equipment (UE) and the dynamic nature of network connections. The simulation assesses metrics such as throughput, delay, and the handling of packet losses. The study's findings highlight the superior performance of the Block Estimation (BLEST) scheduler combined with the BALIA congestion control algorithm, demonstrating the highest throughput and lowest delay among the tested combinations. This outcome suggests that BLEST and BALIA are best suited for enhancing network performance in mixed 5G/4G scenarios, particularly for applications requiring high reliability and low latency. In summary, the paper contributes valuable insights into the application of MPTCP in enhancing data transmission over 5G and 4G networks, providing a pathway for leveraging the full potential of multi-network connectivity in advancing next-generation mobile communications.

8) In [13], Jonas Hammar evaluates that MPTCP suffers from performance loss over asymmetric links. In [14] Christian Markmøller *et al* proposes a Selective Redundant MP-QUIC solution simultaneously provides bandwidth aggregation and higher speeds for background data. The proposed Selective Redundant MP-QUIC protocol introduces selective data duplication with strict prioritization, allowing for a mix of critical and non-critical data streams over a single end-to-end connection. This is particularly aimed at applications like self-driving cars, where critical drive control commands (with high priority) and non-critical map updates (with lower priority) can be managed efficiently over the same connection. The protocol is designed to leverage multiple User Equipment (UE) connections to maximize bandwidth aggregation for non-critical background data while ensuring maximum reliability for critical data through duplication across available paths. This novel approach aims to achieve the high reliability and low latency required for mission-critical applications, surpassing the capabilities of traditional multipath transport protocols like MP-TCP and standard MP-QUIC, which either lack proper prioritization mechanisms or do not support multi-path redundancy effectively. The evaluation of Selective Redundant MP-QUIC involved both emulated network conditions and real-world drive tests. The emulation included fixed-delay channels and a well-known Internet connection model to simulate varying network conditions. The real-world tests were conducted on a highway with two LTE modems, simulating a typical scenario for autonomous vehicles. The performance metrics focused on the protection of critical data flow under background loads and various channel conditions, with particular attention to latency and reliability at the 99.9 percentile. The results demonstrate that Selective Redundant MP-QUIC significantly improves critical traffic latency in the presence of background data loads, under both emulated and real-world conditions. When compared to using a single-path connection or state-of-the-art MP-QUIC protocols, the proposed solution showed a fivefold improvement in the latency of critical traffic at the 99.9 percentile. This highlights the protocol's ability to offer both bandwidth aggregation for non-critical data and enhanced reliability for critical data, making it a promising solution for supporting mission-critical applications in 5G networks.

9) [15] Presents a comprehensive study on XLINK, a multi-path QUIC (Quick UDP Internet Connections) transport solution optimized for video streaming applications. This

work aims to address operational challenges faced by service providers, particularly in delivering high-quality video streaming experiences (robustness, smoothness, responsiveness, and mobility) while minimizing the cost overhead associated with content delivery networks (CDNs). XLINK leverages the user-space capabilities of QUIC to directly integrate user-perceived video Quality of Experience (QoE) into the control of multi-path scheduling and management. This integration aims to overcome challenges such as multi-path head-of-line (HoL) blocking, network heterogeneity, and rapid link variations. XLINK is particularly designed for environments like mobile networks, where devices frequently switch between 5G and 4G connections, potentially affecting the streaming quality due to signal obstruction issues inherent to 5G technology. The core strategy involves selective data duplication for critical data streams, ensuring robust delivery of essential information while aggregating bandwidth for non-critical background data. The paper evaluates XLINK through extensive real-world tests, including over 3 million short video plays within the Taobao Android app, demonstrating significant improvements in video-chunk request completion times, first-video-frame latency, and buffering rates, with only a minimal increase in redundant traffic. This research presents the first large-scale experimental study of multi-path QUIC video transport in production environments, highlighting the practical benefits and challenges of deploying such a system. By demonstrating a successful application of QoE-driven scheduling and path management in a multi-path QUIC setup, the authors contribute significantly to the field of network protocol design, especially for video services over heterogeneous and rapidly changing network conditions. Furthermore, XLINK's approach to addressing the specific requirements of video streaming applications, such as minimizing startup delays and buffering, while efficiently managing network resources, provides valuable insights into the design of network protocols that can meet the growing demands for high-quality video content over the internet. In summary, XLINK exemplifies how advanced transport protocols can leverage application-level feedback to optimize the delivery of data over the internet, offering a promising solution for improving the efficiency and reliability of video streaming services in the face of diverse and changing network conditions.

Previous studies do not provide concrete guidelines for evaluating the performance of MPQUIC and comparing it with MPTCP in the same real-time 5G mmWave environments, especially when Line-of-sight is blocked. Therefore, we carried an attempt to fill this gap by providing a comprehensive analysis of MPQUIC's performance and efficiency, especially in cases where Line-of-sight has been blocked and also compares with MPTCP in similar conditions. (Discussed with the professor about giving the experiment a try)

## III. EXPERIMENTAL SETUP

To investigate the efficacy of Multipath QUIC (MPQUIC) within a 5G mmWave setup, we established a test environment comprising virtual machines that function as the client and the server, utilizing MPQUIC as the transport mechanism. This configuration incorporates several network interfaces, or pathways, to facilitate data transmission between the two machines. Additionally, non-line-of-sight conditions were simulated by obstructing one of these transmission paths to assess performance under varied connectivity scenarios.

### A. Creation of 5G Network Topology using Virtual Machines

Numerous simulation tools exist for modeling complex 5G networks, yet most are tailored to specific segments of a network or focus on particular OSI layers. Consequently, simulators or emulators that comprehensively mimic the behavior across all OSI layers are exceedingly rare. After reviewing multiple alternatives, we opted for a real-time configuration employing virtual machines. This setup allows for the creation of multiple network interfaces, each capable of supporting a bandwidth of 1 GHz, thereby closely approximating the bandwidth conditions characteristic of actual 5G network links.

The client and server virtual machines are set up on Oracle Virtual Box, hosted on a physical machine powered by an AMD Ryzen™ 9 5950X CPU, which features 16 cores and 64GB DDR4 RAM at 5600 MHz. Both virtual machines operate on Ubuntu 20.04, utilizing a Linux Kernel version 5.4.230.mptcp. This particular kernel version is an out-of-tree implementation of MultiPath TCP version 0.96, specifically chosen to maintain consistency with the MPTCP configuration for future comparative analyses with MPQUIC. Each virtual machine is allocated a 2 core and 2 thread CPU configuration, with 4GB of RAM and 20GB of disk space.

To assess the effectiveness of Multipath QUIC (MPQUIC), we established two distinct network interfaces, identified as enps08 and enps09, connecting the client and server virtual machines, as illustrated in Figure 5. These interfaces were configured using two Host-only adapters, vboxnet0 (192.168.111.0/24) and vboxnet1 (192.168.112.0/24), both utilizing Intel PRO/1000 MT Desktop (8250M) adapters. We manually disabled the DHCP servers on these adapters to assign static IP addresses directly to the respective virtual machines. Moreover, any network interface configured by default for NAT to provide Internet access was deactivated to prevent interference during testing. Precise static IP assignments for the interfaces between the client and server were established using the netplan command to ensure consistent communication paths. The network configuration was further secured and customized by implementing firewall rules specifically permitting traffic between the client and server across the designated network links. Additionally, a supplementary firewall rule was introduced to enable host-to-guest communication. To emulate a realistic network environment resembling a 5G infrastructure, we meticulously defined the bandwidth capacities of these links using the traffic control (TC) tool, setting each to handle equivalent data transfer rates as anticipated in 5G networks. This configuration not only provides a controlled environment for effectively testing MPQUIC performance against MPTCP but also replicates the high-bandwidth and low-latency characteristics of 5G networks, thus serving as a robust platform for future comparative studies. The completed network topology utilized for our experiments is elaborated in Figure 5.

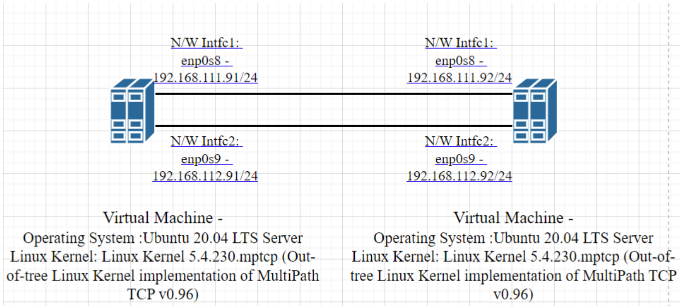


Fig. 5. Client Server virtual Machine running Operating System:Ubuntu 20.04 LTS Server, Linux Kernel: Linux Kernel 5.4.230.mptcp (Out-of-tree Linux Kernel implementation of MultiPath TCP v0.96) with 2 network links between them

### *B. Implementation of Multipath QUIC in the 5G Setup*

To conduct our experimental investigation, we utilize the MPQUIC implementation, which is built upon the quic-go [16] library. Quic-go is an open-source framework for the QUIC protocol in Go language, extended to incorporate Multipath QUIC support [17]. Within the mp-quic library, the scheduler operates as a separate goroutine, running concurrently with the primary event loop of the quic-go server. Alongside the scheduler, mp-quic offers support for various other functionalities of Multipath QUIC, including path management, congestion control, and retransmission.

Golang version 1.17 was installed on both virtual machines, and subsequently, the quic-go library was acquired. Furthermore, the mp-quic library was integrated remotely, and subsequent updates were retrieved to the same directory. This ensures that any future modifications to the mp-quic library will be closely monitored and seamlessly integrated into the local repository [18]. This library is then incorporated into the server and client scripts, with the multipath feature activated by configuring the CreatePaths variable to true within the QUIC Config object. Access to the code utilized for Server and Client is provided through the GitHub repository: https://github.com/AbhishekKittur/MPQUIC

To delve into network performance intricacies, we employ qperf [19], a network measurement tool tailored for analyzing QUIC traffic, akin to Iperf. qperf facilitates UDP throughput measurement and evaluates network attributes such as bandwidth, delay, and jitter. This library, derived from https://github.com/h2o/quicly, is coded in C language. It operates as a command-line interface, establishing TCP or UDP data streams between endpoints to gauge diverse data transfer metrics. Thus, by activating the multipath option, we can gauge the throughput of QUIC traffic traversing via two network interfaces under diverse network conditions.

In each experimental iteration, we gather comprehensive data encompassing various parameters. Firstly, we measure the throughput, quantified in bits per second (bps), representing the volume of data transmitted between the client and server within a defined timeframe. Additionally, we capture the Congestion Window size for both network interfaces, providing insights into congestion management. Our experimental design encompasses multiple trials wherein we systematically manipulate several variables. These include adjusting the bandwidth (measured in Mbps), varying the packet loss rate, and introducing different delay intervals between the client and server. Furthermore, we explore the impact of diverse congestion control algorithms and schedulers. To ensure robustness and reliability, each experiment is meticulously conducted five times, thereby enabling the calculation of the mean throughput. This rigorous approach allows us to analyze the network performance comprehensively and derive meaningful insights.

To assess the robustness of the multipath network, our primary objective is to investigate its response to line of sight loss and its reliability within 5G networks. We aim to explore two specific scenarios: a) In instances where a path becomes unavailable due to shifting network conditions, it is imperative that all traffic seamlessly redistributes to the remaining active paths. b) When a path is obstructed due to line of sight loss during transmission, packets from this path should be promptly re-routed over other active paths in real-time. To achieve this, we utilize goroutines and various interrupt time intervals. One routine generates interrupts at fixed intervals, while another thread awaits these interrupts. The interrupt handler acquires a lock before processing the interrupt and releases it upon completion. Subsequently, the second goroutine waits for the lock's release before acquiring it to initiate interrupt processing. By measuring the time elapsed between interrupt generation and lock acquisition by the second thread, we can ascertain interrupt latency. By adjusting interrupt intervals, we effectively simulate and analyze the impact of Non-Line-of-Sight (NLOS) conditions on the multipath network's performance.

## IV. PERFORMANCE EVALUATION

### *A. Challenges*

We opted for a survey project. However, upon observing numerous experiments, we were inspired to pursue experimentation, drawing from previous studies related to our chosen topic. In this section, our aim is to comprehensively assess the performance of MPQUIC across various network conditions and scenarios, including the challenging Non-Line-of-Sight (NLOS) scenario, where 5G signals may encounter obstruction from physical barriers like buildings or walls. To achieve this objective, we meticulously planned an experimental setup. However, during the setup process, we encountered several challenges. The primary hurdle was the utilization of an outdated version of the Go programming language for the MPQUIC tool, which posed compatibility issues with our system running the latest Go version (1.17).

Moreover, during our exploration, we encountered a limitation with the widely used iPerf tool, which, while supporting TCP and UDP traffic, did not offer explicit support for QUIC traffic, a crucial requirement for our experiments. In response to this challenge, we embarked on a quest for alternative tools and came across qPerf. Although qPerf possessed similar functionalities to iPerf, it lacked support for Multipath QUIC, which was essential for our research. In subsequent experiments, recognizing the need for QUIC traffic generation, we developed a rudimentary traffic generator akin to iPerf. This tool generates random data streams with a buffer size of 1024000 bytes over a specified duration. Subsequently, we calculated throughput while varying delay and loss parameters to gauge network performance under diverse conditions. Despite our efforts, our attempts to create a synchronized interrupt timing goroutine to simulate Non-Line-of-Sight (NLOS) scenarios proved to be challenging and ultimately unsuccessful, despite our best implementation efforts.

## B. Network parameter measurements

Researchers employ a combination of tools and methodologies to scrutinize the performance of MPQUIC across diverse network conditions. These conditions encompass variations in bandwidth, delay, loss, and Line-of-Sight (LOS) obstruction. Through systematic experimentation, researchers measure these parameters under various scenarios, including the manipulation of the number of available paths and the duration of qperf traffic transmission. By meticulously varying these factors, researchers gain insights into how MPQUIC operates under different circumstances, enabling a comprehensive assessment of its capabilities and limitations in real-world networking scenarios. This approach facilitates a nuanced understanding of MPQUIC's behavior, guiding further optimizations and improvements to enhance its efficacy and reliability in practical applications.

To assess MPQUIC's performance in various network setups, we may use a variety of parameters, including:

[1] Bandwidth: MPQUIC's performance can be influenced by varying levels of available bandwidth, particularly when leveraging multiple pathways simultaneously. For instance, employing two links each with a bandwidth of 500 Mbps can potentially meet the mmWave specifications required for robust 5G connectivity. [2] Delay: Latency of [(40,0), (40,40), (40,80)] microseconds with a 5 percent normal distribution provides the network characteristics such as distance, routing, and processing time. [3] Loss: Currently, we have not introduced any packet loss on either link. However, it is feasible to conduct further tests to evaluate MPQUIC's performance in handling packet loss across varying loss rates. [4] Congestion control algorithms: Various congestion control algorithms, including Balia, Cubic, and Olia, are employed to examine their impact on MPQUIC's performance under diverse network congestion scenarios. This investigation aims to ascertain how MPQUIC behaves in congested network environments and assess the effectiveness of different congestion control strategies. [5] Schedulers: We evaluate the performance of MPQUIC under different schedulers, including Blest, Default Lowest RTT, and ECF. This analysis aims to understand how these schedulers influence MPQUIC's behavior and performance in network scenarios, providing valuable insights into optimal scheduling strategies for MPQUIC deployment.

In light of the numerous challenges encountered in our project, we conducted experiments by manipulating delay and loss parameters, introducing distinct delays in the two network links between the client and server. The experimentation encompassed a spectrum of delay combinations, including durations of 12ms, 30ms, and 60ms for each delay combination, coupled with a buffer size of 102400 bytes (equivalent to 100KB), and varied losses on each link.

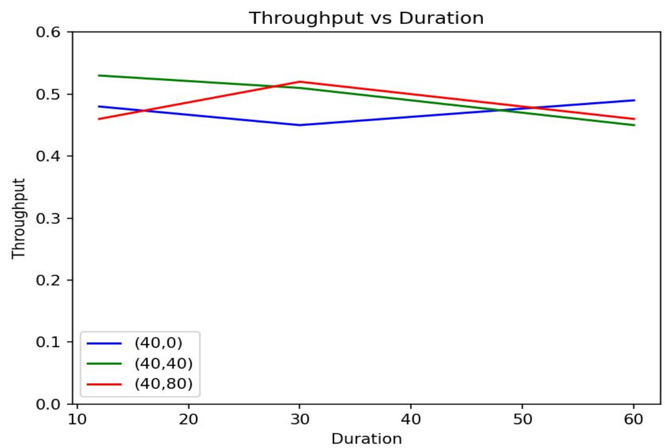


Fig. 6. Throughput vs duration for different delays

According to the data presented in Figure 6, analyzing the throughput values for MPQUIC under varying delay configurations between the two links reveals several observations: The throughput values across all three delay configurations (40s-0s, 40s-40s, and 40s-80s) exhibit relatively similar trends, albeit with minor fluctuations influenced by the duration of delay and the specific time intervals measured. This consistency suggests that MPQUIC demonstrates a stable level of performance across different delay scenarios, effectively adapting to varying network conditions while maintaining reliable throughput.

The influence of delay on MPQUIC throughput becomes evident, particularly with the 40s-40s delay configuration exhibiting slightly lower throughput values compared to other configurations. This disparity can be attributed to the additional latency introduced between the two links, impacting the overall throughput. However, it's important to note that MPQUIC is engineered to dynamically adapt to varying network conditions, including congestion, packet loss, and latency. MPQUIC employs a sophisticated congestion control mechanism that continually adjusts the data transmission rate based on network feedback, thereby ensuring consistent throughput levels despite fluctuations in network parameters. Furthermore, the utilization of multiple concurrent streams in MPQUIC enhances flexibility and redundancy in data transmission, effectively mitigating the impact of network-related issues such as latency and congestion.

## V. CONCLUSION AND FUTURE WORK

Based on our comprehensive survey findings, it is evident that MPQUIC holds significant promise for effectively managing data packets in Non-Line-of-Sight (NLOS) scenarios. However, despite its potential, further efforts are required to refine its implementation and conduct thorough testing. As of now, MPQUIC has not achieved standardization, and a dedicated team is actively working on the IETF draft to address this gap. Additionally, the libraries associated with MPQUIC have not undergone extensive testing to ensure compliance with established standards, indicating a need for rigorous validation procedures. Moreover, the progress of MPQUIC deployment could be hindered by the existing infrastructure's limitations in terms of bandwidth capabilities. Enhancements in infrastructure to support higher bandwidth requirements may be necessary to fully realize the benefits of MPQUIC in real-world applications. In conclusion, while MPQUIC demonstrates considerable potential, ongoing efforts in standardization, testing, and infrastructure improvement are essential to maximize its effectiveness and widespread adoption in diverse networking environments.

## REFERENCES


[1] J. A. Adebusola, A. A. Ariyo, O. A. Elisha, A. M. Olubunmi, and O. O. Julius, "An Overview of 5G Technology," 2020 International Conference in Mathematics, Computer Engineering and Computer Science (ICMCECS), Ayobo, Nigeria, 2020, pp. 1-4, doi: 10.1109/ICMCECS47690.2020.240853.

[2] J. M. Khurpade, D. Rao and P. D. Sanghavi, "A Survey on IOT and 5G Network," 2018 International Conference on Smart City and Emerging Technology (ICSCET), Mumbai, India, 2018, pp. 1-3, doi: 10.1109/ICSCET.2018.8537340.

[3] I. K. Jain, R. Kumar and S. S. Panwar, "The Impact of Mobile Blockers on Millimeter Wave Cellular Systems," in IEEE Journal on Selected Areas in Communications, vol. 37, no. 4, pp. 854-868, April 2019, doi: 10.1109/JSAC.2019.2898756.

[4] M. Scharf and S. Kiesel, "NXG03-5: Head-of-line Blocking in TCP and SCTP: Analysis and Measurements," IEEE Globecom 2006, San Francisco, CA, USA, 2006, pp. 1-5, doi: 10.1109/GLOCOM.2006.333.

[5] SIGCOMM '17: Proceedings of the Conference of the ACM Special Interest Group on Data Communication https://dl.acm.org/doi/abs/10.1145/3098822.3098842.

[6] Q. Peng, A. Walid, J. Hwang and S. H. Low, "Multipath TCP: Analysis, Design, and Implementation," in IEEE/ACM Transactions on Networking, vol. 24, no. 1, pp. 596-609, Feb. 2016, doi: 10.1109/TNET.2014.2379698.

[7] Barré, Sébastien Paasch, Christoph Bonaventure, Olivier. (2011). MultiPath TCP: From Theory to Practice. 444-457. 10.1007/978-3-642-20757-0-35.

[8] M. Jadin, G. Tihon, O. Pereira and O. Bonaventure,"Securing multipath TCP: Design implementation," IEEE INFOCOM 2017 - IEEE Conference on Computer Communications, Atlanta, GA, USA, 2017, pp. 1-9, doi: 10.1109/INFOCOM.2017.8057011.

[9] De Coninck, Quentin, and Olivier Bonaventure. "Multipath quick: Design and evaluation." In Proceedings of the 13th international conference on emerging networking experiments and Technologies, pp. 160-166. 2017.

[10] Viernickel, Tobias, Alexander Froemmgen, Amr Rizk, Boris Koldehofe, and Ralf Steinmetz. "Multipath QUIC: A deployable multipath transport protocol." In 2018 IEEE International Conference on Communications (ICC), pp. 1-7. IEEE, 2018.

[11] Morawski, Michał, and Przemysław Ignaciuk. "MPTCP or MPQUIC-Which One is Better for General-Purpose Networking." In 2021 25th International Conference on System Theory, Control and Computing (ICSTCC), pp. 600-605. IEEE, 2021.

[12] Mahmud, Imtiaz, Tabassum Lubna, and You-Ze Cho. "Performance Evaluation of MPTCP on Simultaneous Use of 5G and 4G Networks." Sensors 22, no. 19 (2022): 7509.

[13] Multipath Packet Scheduling for 5G Systems, Karlstad University, Faculty of Health, Science and Technology (starting 2013), Department of Mathematics and Computer Science (from 2013). Karlstad University, Faculty of Economic Sciences, Communication and IT, Department of Computer Science.

[14] R. S. Mogensen, C. Markmoller, T. K. Madsen, T. Kolding, G. Pocovi and M. Lauridsen, "Selective Redundant MP-QUIC for 5G Mission Critical Wireless Applications," 2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring), Kuala Lumpur, Malaysia, 2019, pp. 1-5, doi: 10.1109/VTCSpring.2019.8746482.

[15] Zhilong Zheng, Yunfei Ma, Yanmei Liu, Furong Yang, Zhenyu Li, Yuanbo Zhang, Jiuhai Zhang, Wei Shi, Wentao Chen, Ding Li, Qing An, Hai Hong, Hongqiang Harry Liu, and Ming Zhang. 2021. XLINK: QoE-driven multi-path QUIC transport in large-scale video services. In Proceedings of the 2021 ACM SIGCOMM 2021 Conference (SIGCOMM '21). Association for Computing Machinery, New York, NY, USA, 418–432. https://doi.org/10.1145/3452296.3472893

[16] https://github.com/lucas-clemente/quic-go

[17] https://github.com/qdeconinck/mp-quic

[18] https://multipath-quic.org/2017/12/09/artifacts-available.html

[19] https://github.com/rbruenig/qperf


### A. Appendix

This section elaborates on the individual contributions made by each team member to both the survey project and the experiment. It highlights the equal participation and efforts invested by all members. Detailed descriptions of each member's contributions are provided below, showcasing their respective roles, responsibilities, and accomplishments throughout the duration of the project.

The Abstract and Introduction sections have been enhanced through the collaborative efforts of all four members, incorporating improvements suggested by each individual. In related work each member were assigned the papers and each one has provided the summary for the same.

Abhishek Kittur : was involved in the analysis of papers 1, 2, and 3, and also contributed to setting up the Experimental Setup.

Pranav Vishnu Anandakumar : was involved in the analysis of papers on 4, 5, and also contributed in Performance Evaluation.

Ashish Ajayakumar : was involved in the analysis of papers on 6, 7, and also contributed in Performance Evaluation.

Sujay Sreedhar : was involved in the analysis of papers 8, 9, and also contributed to setting up the Experimental Setup.