


## Article

# On the Potential of MP-QUIC as Transport Layer Aggregator for Multiple Cellular Networks

Zsolt Krämer<sup>1,2,\*</sup>, Felicián Németh<sup>1,3</sup>, Attila Mihály<sup>2</sup>, Sándor Molnár<sup>1</sup>, István Pelle<sup>1,3</sup>, Gergely Pongrácz<sup>2</sup> and Donát Scharnitzky<sup>1</sup>

<sup>1</sup> High Speed Networks Laboratory, Department of Telecommunications and Media Informatics, Faculty of Electrical Engineering and Informatics, Budapest University of Technology and Economics, Műegyetem rkp. 3, H-1111 Budapest, Hungary; nemeth.felician@vik.bme.hu (F.N.); molnar.sandor@vik.bme.hu (S.M.); pelle.istvan@vik.bme.hu (I.P.); scharnitzky.donat@vik.bme.hu (D.S.)

<sup>2</sup> Ericsson Hungary, H-1117 Budapest, Hungary; attila.mihaly@ericsson.com (A.M.); gergely.pongracz@ericsson.com (G.P.)

<sup>3</sup> MTA-BME Network Softwarization Research Group, Műegyetem rkp. 3, H-1111 Budapest, Hungary

\* Correspondence: zsolt.kramer@ericsson.com

**Abstract:** Multipath transport protocols have the ability to simultaneously utilize the different paths and thus outperform single-path solutions in terms of achievable goodput, latency, or reliability. In this paper our goal is to examine the potential of connecting a mobile terminal to multiple mobile networks simultaneously in a dynamically changing environment. To achieve this, first we analyze a dataset obtained from an LTE drive test involving two operators. Then we study the performance of MP-QUIC, the multipath extension of QUIC, in a dynamic emulated environment generated from the collected traces. Our results show that MP-QUIC may leverage multiple available channels to provide uninterrupted connectivity, and a better overall goodput even when compared to using only the best available channel for communication. We also compare the MP-QUIC performance with MPTCP, identify challenges with the current protocol implementations to fill in the available aggregate capacity, and give insights on how the achievable throughput could be increased.

**Keywords:** QUIC; MP-QUIC; multipath; LTE; cellular networks



check for updates

**Citation:** Krämer, Z.; Németh, F.; Mihály, A.; Molnár, S.; Pelle, I.; Pongrácz, G.; Scharnitzky, D. On the Potential of MP-QUIC as Transport Layer Aggregator for Multiple Cellular Networks. *Electronics* **2022**, *11*, 1492. <https://doi.org/10.3390/electronics11091492>

Academic Editors: Piotr Zwierzykowski, Erich Leitgeb and Paweł Śniatała

Received: 31 March 2022

Accepted: 4 May 2022

Published: 6 May 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

TCP has been the transport protocol used for most of the Internet traffic for more than 40 years. Many proposals have been made to improve its efficiency and fairness during this time. The speed of evolution in the transport layer of the Internet has been increasing since the arrival of the QUIC protocol [1]. Originally designed by Google and now being standardized by the IETF, QUIC brought numerous novel features and since then has been widely deployed.

Multipath transport is a more recent field which achieved some maturity and has seen recent deployments in smartphones. Over the years, multipath transport protocols have been improved and studied by different teams [2], however, the vast majority of the works are focusing on Wifi-cellular aggregation and handovers. A recent survey [3] categorized transport layer multi-connectivity solutions as either Above-the-Core or Core-Centric. Above-the-Core represents a simple case when multipath transport is deployed on both the client and the server, and the aggregation of path happens end-to-end. A Core-Centric solution is when the client is running multipath transport, however, the multipath aggregation is terminated by a proxy in the core network and the connection between the proxy and the server is single-path. This concept is leveraged by the recently proposed ATSSS (Access Traffic Steering, Switching, and Splitting) architecture.

In this paper we aim to explore the feasibility of using multipath (MP-)QUIC in next-generation cellular networks, assuming that the mobile terminals support multiple

simultaneous access to the cellular network, similarly as they do today for 3GPP and WIFI accesses. Based on this assumption, we envision a scenario where the mobile terminal connects to multiple cells of potentially multiple networks, where the connected cells would use different frequency bands to avoid self-interference, and the networks would provide a separate IP address for the terminal for every accessible cell. The terminal would then use a multi-path extension of QUIC that can switch the traffic among different accesses, retransmit packets on a new access if the first becomes congested, or even send packets on both accesses simultaneously if they are both available. In this approach, terminal mobility is not about moving the cellular link from one cell to another in a terminal (and transport) agnostic way, but keeping a list of reachable cellular interfaces from which the transport layer can choose which connection to add or remove from the active connection list handled by MP-QUIC. Such a mobility solution may require specific network cooperation mechanisms and business models as well as new terminal features; however, this is out of the scope of the present paper.

Our three main contributions are the following. First, in order to get a realistic figure about how many possible connections a user can have in a real mobile network, we analyze the data retrieved from real-life multi-carrier LTE drive tests. Second, based on these results, we calculate the number and capacity of the available paths a user can get resulting in a time-varying connection graph for each user. We also calculate the theoretical performance improvement achievable by different levels of transport layer aggregation. Third, we evaluate the proposed solution based on long file downloads applied on this connection graph and show crucial aspects of the behavior of MP-QUIC.

This paper is organized as follows. Section 2 presents an overview of related work, then Section 3 shows statistical insights gathered from an LTE drive test, Section 4 describes our measurement setup and performance evaluation results, then Section 5 concludes the paper.

## 2. Related Work

### 2.1. *Dynamic Behavior of Transport Protocols in 4G and 5G Cellular Networks*

A comprehensive analysis of LTE physical layer statistics and behavior in high-velocity environments can be found in [4]. Based on this, the authors of [5] conducted a performance evaluation of congestion control algorithms in an LTE drive test, also presenting the measured LTE characteristics. In their measurements, only 35% of TCP flows experienced at least one handover. While only one of 720 TCP sessions experienced connection failure, the authors show that handovers still have a significant negative impact on performance, especially by confusing the BBR (Bottleneck Bandwidth and RTT) [6] congestion control algorithm.

In [7] the authors use the MONROE platform to conduct a large-scale measurement campaign on commercial cellular networks in Northern Europe. A comprehensive analysis of the captured dataset is presented, showing that the average RTT for LTE networks was around 38 ms, while for 3G it was 51 ms. When comparing the downlink data rate of stationary and mobile nodes, the authors show that the difference is only 6 Mbps in the median range in favor of the stationary nodes.

The performance of different TCP variants in 5G reference scenarios (high-speed train, dense urban) was investigated in [8]. In the high-speed train scenario, the mmWave access resulted (as expected) in much higher, but more volatile goodput compared to LTE, and the authors also showed that by leveraging dual-connectivity and fast secondary cell handover, it is possible to provide uninterrupted connectivity if the deployment of the base stations adheres to 3GPP guidelines.

### 2.2. *Application of Multipath Transport Solutions for Cellular Networks*

The design rationale of a deployable Multipath TCP was presented in [9], where the authors also implemented MPTCP in the Linux kernel and showed the performance benefits of using 3G and WiFi. A very important contribution of the paper is the effort towards making MPTCP deployable in today's Internet and studying all the potential middlebox interferences that MPTCP may encounter. The authors show that their implementation can

fall back to TCP if the *mp-capable* option is removed by a middlebox and it also utilizes DSM checksums to detect payload-modification by application-level gateways and supports middleboxes that split or coalesce segments. Today MPTCP is getting deployed on mobile devices at an increasing rate, and the authors of a recent, comprehensive study [10] carried out a measurement campaign consisting of both active- and passive measurements and presented some valuable lessons learned. One interesting finding was MPTCP's inefficiency regarding short flows, and the paper discusses a 0-RTT connection establishment as a proposed improvement, which is something that QUIC and MP-QUIC are designed for. Another important finding is that a single-path TCP can outperform MPTCP if the difference between the quality of the two paths is too large, and this is due to the receiver buffer size, which needs to be increased.

In [11] the authors discuss MPTCP as a possible solution to provide end-to-end multi-connectivity, meaning that a UE may connect to an LTE and a mmWave eNB or a number of mmWave eNBs simultaneously without the need for coordination from the network. The paper also evaluates the performance aspects of this via a simulation study, showing that a reliable secondary LTE path delivers higher performance gains to a 28 GHz mmWave link compared to a 73 GHz secondary mmWave path. It has also been showed that using LTE for the uplink connection (sending TCP ACKs) does not provide a clear benefit because of the added latency.

MP-QUIC is a multipath extension to the QUIC protocol. The design, a prototype implementation in Go, and an initial performance evaluation has been presented in [12] (and an alternative implementation later by the authors of [13]). Path identification is achieved in MP-QUIC by placing an explicit *Path ID* in the public header of each packet. This also allows the protocol to preserve states (congestion control, lost packets) for the paths even after an IP address change. MP-QUIC uses a separate packet number space for each path, and also adds a *Path ID* to ACK frames. The protocol has a path manager component that is responsible for the creation and deletion of paths. Unlike MPTCP, MP-QUIC is able to send data in the first packet as it opens a new path, while MPTCP requires a three-way handshake before any data is sent. MP-QUIC also employs a more flexible retransmission mechanism compared to MPTCP, as lost packets can be retransmitted on different paths. For congestion control, the implementation uses the OLIA [14] mechanism which has been shown to perform well in MPTCP.

One of the main design goals of MPTCP was to be Pareto-optimal, meaning that the performance increase of a user utilizing multiple paths should not result in decreased performance for other users sharing the same network resources. However, the interplay and impact of scheduling and congestion control (CC) on resource sharing in transport layer multipath are very complex, and it has been proven that initially MPTCP was not yet Pareto-optimal. There are numerous recent efforts to optimize multipath resource sharing by improving the scheduler. The current implementation of MPTCP in the Linux kernel uses the Lowest-RTT-First (LRF) scheduler, aiming to maximize throughput by trying to utilize the paths with the lowest smoothed RTT. However, this algorithm was designed for symmetric paths and in the case of path-asymmetry, it may underperform. According to a comprehensive measurement study [15] the current state-of-the-art MPTCP schedulers, even if designed with asymmetric paths in mind, do not perform as well as expected. The authors propose two novel scheduling algorithms to enable low-latency scheduling in MPTCP and show that significant latency reduction is achievable with both while retaining comparable throughput. The prototype implementation of MP-QUIC [12] uses a fairly simple round-robin (RR) scheduler. A comparison of potential MP-QUIC schedulers in [16] showed that in dynamic environments, none of the candidates clearly outperforms the others. The authors also propose a novel, online learning scheduler called Peekaboo with a stochastic adjustment strategy and show that it is able to provide over 30% performance improvement compared to the other candidates.

The idea of multipath proxies for cellular networks has inspired a number of different approaches, built on different transport layer protocols. An MPTCP proxy as an enabler

for unlocking the performance benefits of MPTCP with legacy remote hosts was discussed in [17], as part of a mobility architecture that provides better utilization of multiple interfaces and mitigates the negative effects of handoffs. In [18], the design, implementation, and evaluation of a lightweight MPTCP proxy solution were presented. The elegant design proposed by the authors provides TCP-MPTCP protocol conversion without splitting the connection. The functionality of the solution was evaluated in experiments, and it was shown that the lightweight MPTCP proxy is a viable solution for seamless mobility.

The authors of [19] implemented an MP-QUIC proxy using the SOCKS protocol and evaluated the performance compared to an MPTCP proxy in live networks. Regarding web page load times, the solution (called QSOCKS) outperformed the MPTCP proxy by 42% on the top 1000 websites. This shows how the advantages of QUIC compared to TCP in web page downloads (faster connection establishment) can carry over to MP-QUIC versus MPTCP.

Another candidate framework for this problem was described in [20]. MP-DCCP is a multipath extension of DCCP (Datagram Congestion Control Protocol). The paper presents the architecture of the novel multipath framework built on MP-DCCP, where virtual network interfaces provide the integration into existing architectures and gateways, and the DCCP tunneling enables link quality estimation which is utilized by the pluggable scheduling and reordering algorithms. The authors present performance evaluation results showing that MP-DCCP is better at handling latency variation than MP-TCP, and is able to provide in-order delivery of packet streams to the applications which could lead to significant multimedia QoE enhancement. The framework also supports unreliable traffic, which further distinguishes it from MPTCP.

### *2.3. Multipath Performance and Cellular-WiFi Handovers*

The impact of network handovers on multipath performance has been investigated in prior works, however, these focused exclusively on WiFi-cellular handovers. The seminal work carried out in [21] showed that MPTCP is able to provide connectivity for applications like Skype during a WiFi-cellular handover. The authors also proposed improvements for MPTCP to provide better support for handovers, which were implemented in the Linux kernel version of the protocol. In [22] an application—and protocol agnostic framework was proposed to enhance QoE by optimized switching between WiFi and Cellular paths. The architecture is made protocol-independent by a tunneling mechanism that runs in a userspace proxy. One of the main insights in the paper is that the reordering caused by the switching is preventable if one can estimate the downlink queueing delay based on the bytes in-flight. The authors tested the framework on a YouTube application that switches paths based on playout buffer health and found that the optimized switching between cellular and WiFi interfaces can eliminate stalling events.

MP-QUIC follows a similar approach to handling sudden changes in the availability of networks as the current Linux implementation of MPTCP. After one RTO (Retransmission Timeout), the protocol considers a path potentially failed and the scheduler temporarily ignores these paths, thus it sends a request on the newly available path in case of a network handover [12].

In [23] the authors compared the performance of MPTCP and MP-QUIC during network handovers using an iOS application designed for such measurements. The findings show that WiFi-cellular handovers are not necessarily an abrupt process with MPTCP, as 58% of the experiments observed a handover duration of at least 10 s. It is also clear from the results that handovers can have a severe performance impact on the application (60% of test runs involving a handover experienced an application delay longer than 1 s). When comparing the performance of MPTCP and MP-QUIC, the authors did not find any clear trend in the favor of any protocol if the scheduling algorithm was the same. This result is very promising for MP-QUIC, as it is implemented in userspace and capable of evolving its scheduling and congestion control algorithms faster, thus more efficient scheduling algorithms are expected to reach deployments earlier for MP-QUIC than for MP-TCP, at least once MP-QUIC is widely used.

#### 2.4. Performance Enhancement Approaches for Multipath Transport

The MPTCP Opportunistic Linked Increase Algorithm (OLIA) [14] is a congestion control algorithm that organizes paths into three categories:  $p_{\text{best}}$  are paths with the best transmission rate,  $p_{\text{max}(w)}$  have the largest congestion windows (fully used paths) and  $p_{\text{collected}}$  are viable but unused paths. The algorithm increases congestion windows faster on the paths of  $p_{\text{best}}$  with small windows, but slower on  $p_{\text{max}(w)}$  paths. In this latter case, OLIA re-forwards traffic from  $p_{\text{max}(w)}$  paths to  $p_{\text{collected}}$  paths that still have free capacity.

The Proactive Multipath TCP solution [24] takes cross-layer information into consideration when handing off traffic between a cellular and a WiFi link. At pre-handoff, subflow congestion windows are proactively adjusted according to the current bandwidth (estimated from signal strength) and delay characteristics. At handoff, round-trip time (RTT) and retransmission timeout (RTO) estimations are reset, duplicate acknowledgments are temporarily disabled and the congestion window is proactively set to the bandwidth-delay product estimate of the respective path to enable faster convergence and reduce the effects of packet reordering and timeout issues.

Peekaboo [16] and SmartPS [25] both leverage cross-layer information including signal strength and congestion windows, and employ machine learning in order to select the best path for a packet to be forwarded to.

The client-based multipath TCP framework, cMPTCP [26] combines proprietary path-selection (scheduling) and congestion control mechanisms. The client-initiated multipath selection opportunistically disables slow paths. During this, the client monitors the size of the out of order queue (part of the MPTCP receive buffer), and upon transmission performance degradation due to the slow path (i.e., the out-of-order queue becoming large) the slow path is turned off for a period of time determined by the algorithm. The framework also uses subflow-level rate control in which the server adjusts its congestion window based on the RTT measured by the client. This client-to-server feedback relies on an eBPF-based congestion control framework.

### 3. Cellular Network Dynamics in the Wild: A Statistical Investigation

#### 3.1. Estimating Downlink Capacity

Our dataset is obtained from an LTE drive test conducted in Europe, involving two different operators. The client devices were MONROE [27,28] measurement nodes operating on public buses. Even though only the RSRP (Reference Signal Received Power) values were available, as demonstrated in this section, a feasible estimation of SINR (Signal to Interference plus Noise Ratio) and downlink capacity can be derived from these measurements.

The first step is to calculate an estimation of the SINR [29]:

$$SINR = \frac{RSRP_s}{Q_L \sum_{i \neq s} RSRP_i + N_{thermal}(NF_{UE})} \quad (1)$$

where  $RSRP_s$  is the largest RSRP value for each second,  $Q_L$  is the network load and  $N_{thermal}(NF_{UE})$  is the sum of the thermal noise power and UE noise power. From the SINR, the downlink capacity can be estimated using a downscaled and truncated version of the Shannon formula [30]:

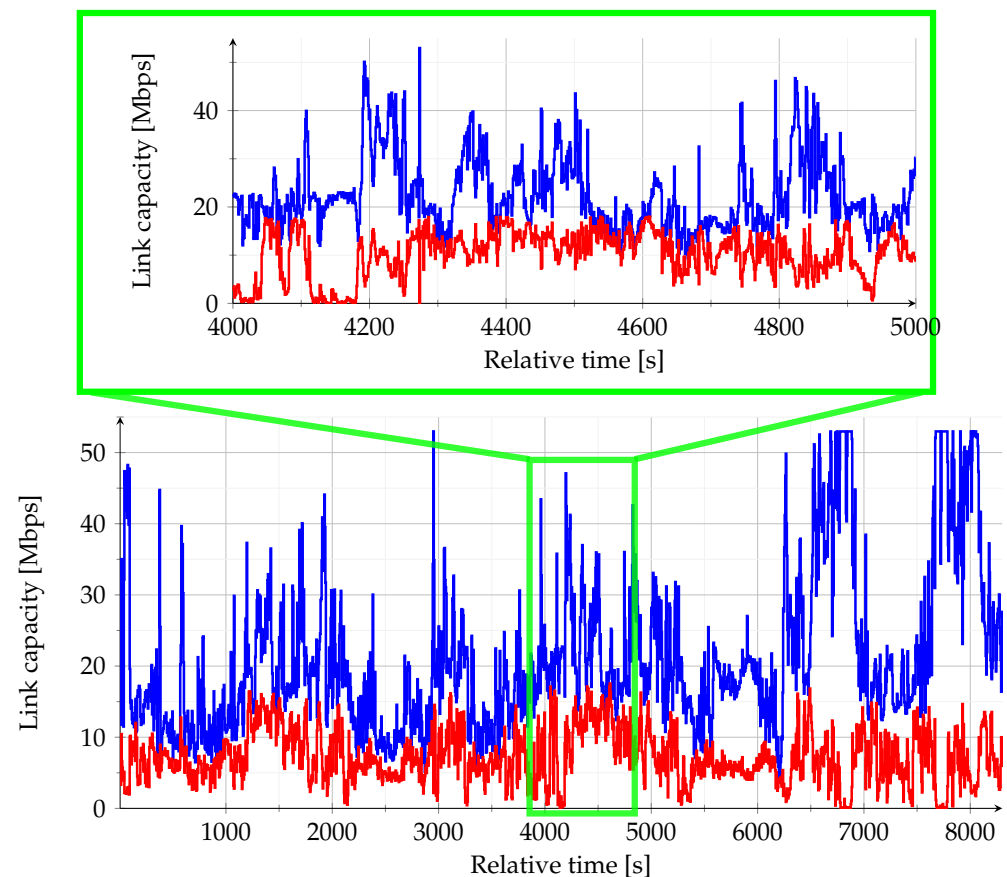
$$B = f \frac{C}{\ln 2} \min\left[\frac{c_{15} \ln 2}{C}, \ln(1 + \gamma SINR)\right] \quad (2)$$

where  $f$  is the bandwidth of the channel,  $C = 0.9449$  (the downscaling constant to the Shannon-formula),  $\gamma = 0.4852$  (constant) and  $c_{15} = 5.5547$  (the efficiency of the given CQI (Channel Quality Indicator) index). All variables and constants used in our calculations are also summarized in Table 1. Using multiple channels of the same carrier simultaneously would result in higher interference, thus we only consider maximum one visible cell per carrier as a potential serving cell.

**Table 1.** Symbols used in the calculations.

Notation	Interpretation	Value
$Q_L$	network load	10%
$N_{thermal}$	thermal noise	−132.24 dBm
$NF_{UE}$	noise figure of the UE	7 dB
$f$	channel bandwidth	10 MHz
$\gamma$	constant	0.4852
$c_{15}$	efficiency of the CQI index 15	5.5547
$C$	downscaling to the Shannon-formula	0.9449

Figure 1 shows the estimated downlink capacity over time for one day of the drive test, where the two different colors correspond to the two strongest cells in each second. The dynamic nature of LTE link capacities is illustrated on two different timescales. An important finding is that there is often a dominant cell with significantly higher capacity compared to the secondary cell. However, it is also visible that efficient aggregation with the secondary cell could yield considerable improvement in the available downlink capacity. From here on, we will assume a low, 10% load in order to study higher peak data rates in a dynamic network environment.



**Figure 1.** Evolution of the available downlink channel capacities of the primary and secondary cells over time: averaged to 1 s (top) and 5 s (bottom) time windows.

### 3.2. Available Cells

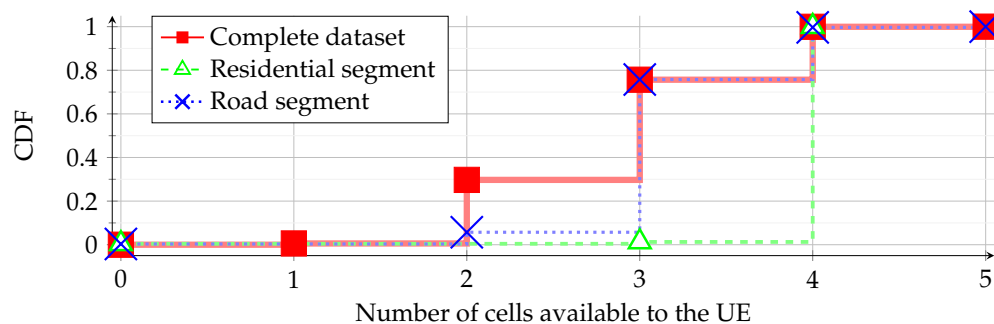
In this section, we present a statistical description of the dataset regarding the number of available cells for the UE.

Table 2 summarizes the findings for the whole dataset, as well as two selected shorter periods (residential and road segment) which we used later for the emulation in our measurements. Our dataset is comprised of more than 33 h of drive test data and the results show that the average number of available cells for the measurement device is 2.97. As mentioned in Section 3.1, we only consider a maximum of one cell available per carrier. The number of visible cells in the dataset is higher than this value, i.e., 4.26 on average and higher than seven in 20% of the cases.

**Table 2.** Average number of available cells.

Environment	Average Number of Available Cells
Entire dataset	2.97
Residential segment	3.99
Road segment	3.19

The cumulative probability of the number of available cells can be seen in Figure 2. In the residential segment, there are exactly four available cells for nearly the entire time, while in the road segment, this is only true in less than 25% of the time. In the remaining, over 75% of the cases, the number of available cells is either two or three, with extremely brief periods of only one cells available. In the dataset, it is more common to have only two available cells compared to either selected segments.



**Figure 2.** Cumulative distribution of available cells.

The findings about the average number of available cells in our dataset reveal untapped potential for multipath transport protocols to utilize the aggregated capacity of multiple paths.

### 3.3. Capacity Statistics and the Potential of Multipath Aggregation

Table 3 contains a statistical evaluation of the available capacity over the entire dataset, assuming different levels of transport layer multipath aggregation. OP1 SP and OP2 SP represent single path reference cases for the first and the second operator, respectively. Best Single Path is a hypothetical, optimized single path transport layer solution that utilizes the best available path in each second, which can be selected from either operator. Finally, Multipath is the aggregated capacity of all available paths. A similar constraint to the one applied to the number of available cells has to be applied here also, we only aggregate the largest capacity for each carrier.

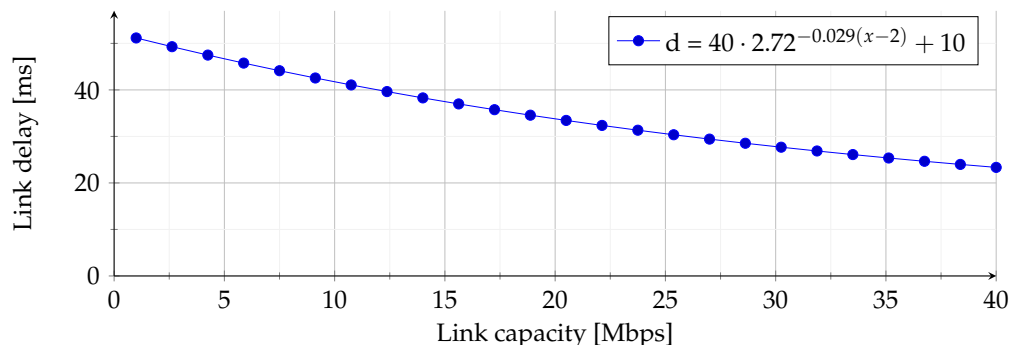
**Table 3.** Average capacity (in Mbps) over the whole dataset assuming different levels of aggregation.

Aggregation Level	Average	Median	Percentiles				
			20th	10th	5th	1st	0.1th
OP1 SP	25.74	23.24	9.05	4.70	2.12	0.43	0.03
OP2 SP	19.37	16.58	2.17	0.59	0.22	0.03	0.01
Best Single Path	36.57	34.63	25.08	22.69	20.56	17.59	15.48
Multipath	46.15	44.94	39.92	38.41	36.91	34.78	32.97

Regarding the average values, multipath aggregation increases the available capacity with 79% and 138% over the first and the second operator, respectively. Compared to the Best Single Path solution, the gain is significantly lower (26%). In order to get insights regarding the reliability provided by each aggregation type, we studied the cases with lower than average capacity. Even for the 20th percentile, the multipath capacity is more than 4 times larger than the single path capacity of the first operator, and more than 18 times larger than that of the second operator. The difference becomes even more extreme in the lower percentiles, which means that multipath aggregation in cellular networks has a significant potential to enhance performance in use-cases where reliability is crucial. One such use-case could be telemetry data in vehicular communications.

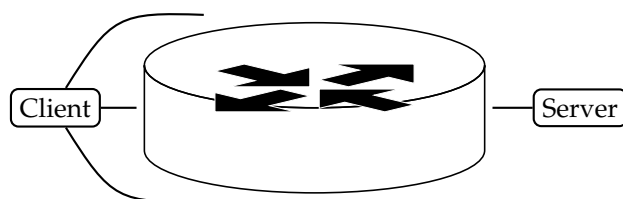
**4. Testbed Measurements and Results**

We evaluated the multipath QUIC implementation of Picoquic [31]. We extended the Mininet [32] based Minitopo [33] tool to precisely change link bandwidth and delay parameters during measurements. However, the measurement traces lack latency data, so the tool uses a simple delay estimation formula shown in Figure 3.



**Figure 3.** Propagation delay as a function of link capacity. The curve is fitted on two typical 5G values:  $d(50) = 20, d(200) = 100$ .

Our emulation sets up a test network where the client accesses a switch via different numbers of parallel paths and the traffic between the server and the switch is multiplexed to a single link, as illustrated in Figure 4.



**Figure 4.** Network setup.

The data-flow diagram in Figure 5 shows the steps taken by our emulation at a high level. First, bandwidth traces and Minitopo templates (for topology, source, and destination application setup) are processed by Minitopo-utils, our extension to Minitopo, creating



topology and application configuration files that can be processed by Minitopo and Mininet. These latter tools start the emulated network and pass command line arguments to the MP-QUIC (or MPTCP) applications running at the server and client hosts. During the emulation, log files are created that are processed by Minitopo-utils at the end of each test run to generate summary tables and charts. We also extended the Picoquic demo client to add and abandon paths following a predefined script because it was not capable to handle state changes of the network interfaces.

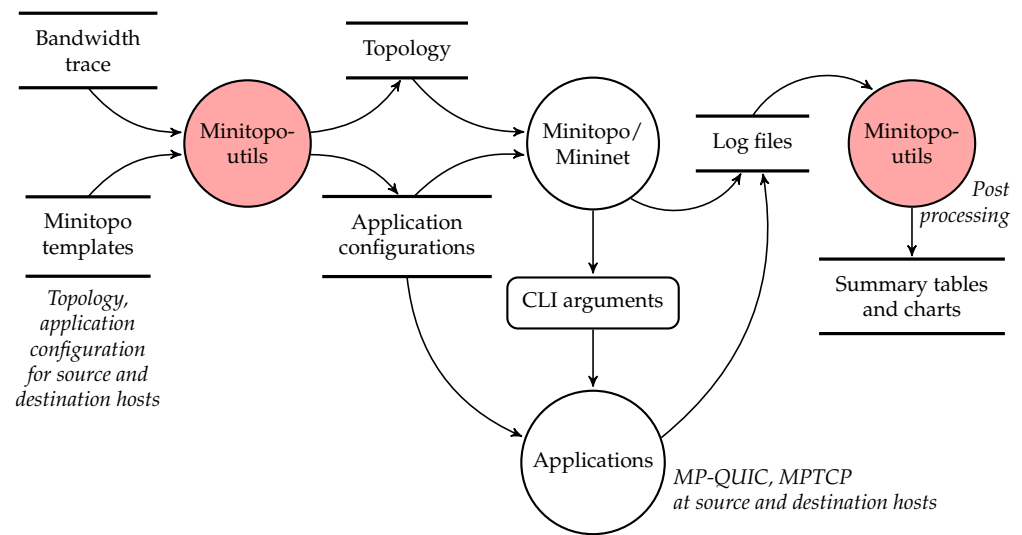


Figure 5. High-level internal operation of the emulation environment as a data-flow diagram.

Figure 6 demonstrates Picoquic capabilities in a simple scenario. The capacity of the first link is a constant 4 Mbps throughout the whole measurement, while the second link has a capacity of 8 Mbps, but the client can access it only between 2–4 s and 6–8 s. The green multipath goodput curve, which is the sum of the goodput values of individual QUIC paths, shows the protocol is able to follow the fluctuations of the overall available capacity. Note that paths and links are not in a one-to-one relationship: the initial (red) qpath terminates after 2 s and a new (brown) path takes its place achieving 4 Mbps until 6.25 s, when another (magenta) path replaces the brown path. This is in contrast with the (green) path of the fluctuating link, which does not transmit data between 4–6 s, but restarts after 6 s.

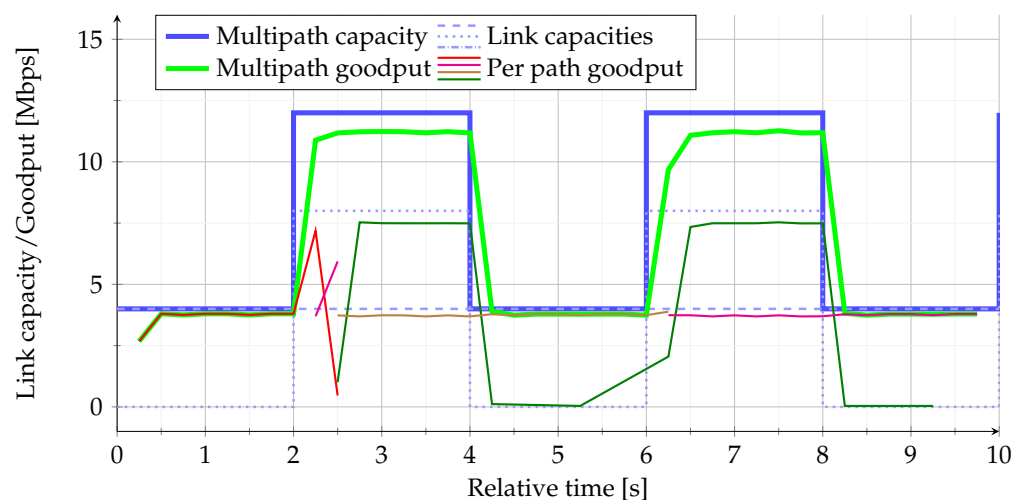


Figure 6. QUIC in an idealized environment.

In the next set of measurements, the client downloads a 40 MB-sized file, and the characteristics of the parallel paths are set according to the pre-recorded trace file of Section 3. Tables 4 and 5 summarize the results. Each line shows the statistics of 18 measurements started from different points of the trace files. The multipath capacity columns correspond to the sum of link capacities during the download just like in Figure 6; the single-path capacity columns are calculated by selecting the highest link capacity at a given time; the goodput and completion columns show the client's achieved performance metrics.

**Table 4.** Measurement results of Picoquic and MPTCP downloading a 40 MB-sized file in the *residential* environment.

idx	CC alg.	Case	Goodput [Mbps]		MP Capacity [Mbps]		SP Capacity [Mbps]		Completion Time [s]	
			avg	SD	avg	SD	avg	SD	avg	SD
1	bbr	sp	11.68	3.30	13.41	3.78	13.41	3.78	28.17	5.85
2	cubic	sp	11.35	2.76	13.23	3.37	13.23	3.37	28.22	5.13
3	fast	sp	8.63	1.56	12.86	2.34	12.86	2.34	36.50	7.74
4	reno	sp	9.90	3.69	13.73	3.90	13.73	3.90	34.29	8.76
5	bbr	1	6.90	2.76	12.69	2.54	12.68	2.54	41.27	19.50
6	cubic	1	6.05	2.86	12.55	2.39	12.54	2.39	47.10	25.65
7	fast	1	6.23	2.73	12.88	2.35	12.87	2.35	56.84	27.61
8	reno	1	3.70	1.86	12.35	2.20	12.34	2.20	83.03	40.28
9	bbr	2	13.66	5.17	20.95	4.07	13.94	4.55	20.92	5.23
10	cubic	2	13.88	3.12	20.91	3.18	13.90	3.84	25.57	9.30
11	fast	2	12.32	5.09	21.91	5.53	14.88	5.91	34.98	42.25
12	reno	2	10.57	3.47	19.76	2.18	12.87	2.57	33.14	10.23
13	bbr	3	13.53	5.35	23.63	2.34	12.82	2.75	24.51	12.64
14	cubic	3	15.71	3.24	23.67	2.13	12.97	2.57	23.13	5.86
15	fast	3	15.92	4.43	24.06	3.79	13.62	4.47	26.35	29.71
16	reno	3	13.05	2.84	23.99	2.35	13.28	3.08	26.25	8.24
17	bbr	4	10.05	4.64	25.21	2.49	13.62	3.38	38.13	13.58
18	cubic	4	14.81	3.56	25.36	2.77	13.72	3.41	28.43	11.27
19	fast	4	18.12	2.67	25.94	4.11	14.46	5.26	19.96	4.19
20	reno	4	13.28	3.49	25.58	2.87	13.92	3.81	29.24	7.29
21	mptcp	sp	12.31	7.12	13.85	8.33	13.85	8.33	27.72	5.24
22	mptcp	1	7.93	7.33	13.27	6.75	13.26	6.75	62.17	31.98
23	mptcp	2	7.12	8.73	18.90	6.83	11.93	6.58	81.28	51.02
24	mptcp	3	8.13	9.22	21.11	9.98	12.68	7.17	101.35	108.53
25	mptcp	4	11.35	13.76	24.79	7.47	12.94	8.09	38.59	15.85

There are different cases presented in these Tables denoted by the column *topo* as follows. Case sp shows an idealized, baseline case when the client can only use a single channel at a time, but it can instantly perform seamless handovers to the channel with the highest link capacity. Observe that the achieved goodput is slightly below the available capacity because Picoquic cannot follow link-capacity fluctuations perfectly. In some cases, for example, with the reno congestion control algorithm (line 3) the goodput is well below the SP capacity.

In case 1, the client is still not allowed to use parallel links, but it has to perform more realistic handovers. We would expect comparable performance to the baseline just by looking at the simple demonstrative Figure 6. However, the actual performance is significantly reduced compared to the baseline case. For example, Picoquic with BBR congestion control plummets to 60% goodput (lines 1 and 5 of Table 5). Note that this handover strategy is a bit simplistic as the user equipment does not immediately switch away from radio stations with temporal drops in channel capacity.

**Table 5.** Measurement results of Picoquic and MPTCP downloading a 40 MB-sized file in the *road* environment.

idx	CC alg.	topo	Goodput [Mbps]		MP Capacity [Mbps]		SP Capacity [Mbps]		Completion Time [s]	
			avg	SD	avg	SD	avg	SD	avg	SD
1	bbr	sp	15.67	3.90	18.27	4.76	18.27	4.76	20.34	3.66
2	cubic	sp	15.12	2.87	18.19	4.57	18.19	4.57	20.67	3.29
3	fast	sp	10.50	2.56	17.83	3.26	17.83	3.26	33.30	10.91
4	reno	sp	12.92	5.95	18.76	4.60	18.76	4.60	26.08	8.92
5	bbr	1	9.21	6.67	19.10	3.72	19.09	3.72	33.20	13.89
6	cubic	1	10.03	6.57	19.33	4.13	19.32	4.13	39.26	15.76
7	fast	1	4.24	2.95	18.11	1.46	18.10	1.46	78.90	44.59
8	reno	1	6.47	6.56	19.15	3.60	19.14	3.60	73.74	36.61
9	bbr	2	16.33	6.54	27.12	4.99	18.33	4.93	21.19	11.81
10	cubic	2	18.70	5.93	26.37	5.06	18.01	4.99	16.38	4.83
11	fast	2	14.08	5.43	26.45	4.15	17.60	3.70	28.04	27.21
12	reno	2	11.81	5.36	26.83	3.09	17.33	2.48	31.06	16.24
13	bbr	3	13.63	6.40	30.55	3.17	18.97	4.43	29.72	13.02
14	cubic	3	21.19	5.85	28.91	3.97	18.32	4.49	19.20	12.57
15	fast	3	18.05	3.49	29.25	4.59	18.12	4.58	18.28	4.17
16	reno	3	14.41	5.54	30.41	3.04	18.19	2.86	27.22	10.51
17	bbr	4	12.17	7.20	30.05	3.68	17.34	2.76	36.64	29.14
18	cubic	4	20.30	5.59	30.71	3.48	18.64	4.18	19.14	5.92
19	fast	4	17.77	4.99	30.11	4.33	18.31	4.26	20.95	12.78
20	reno	4	15.08	6.76	30.96	2.81	18.34	3.28	26.02	11.03
21	mptcp	sp	17.07	7.54	18.89	6.75	18.89	6.75	19.89	3.80
22	mptcp	1	8.00	8.98	18.43	7.57	18.42	7.57	67.67	35.54
23	mptcp	2	9.08	10.67	25.88	11.41	18.63	7.38	76.11	48.78
24	mptcp	3	9.46	9.97	30.56	9.73	18.21	7.16	82.29	94.97
25	mptcp	4	11.89	11.96	32.00	9.78	18.18	7.43	50.79	34.58

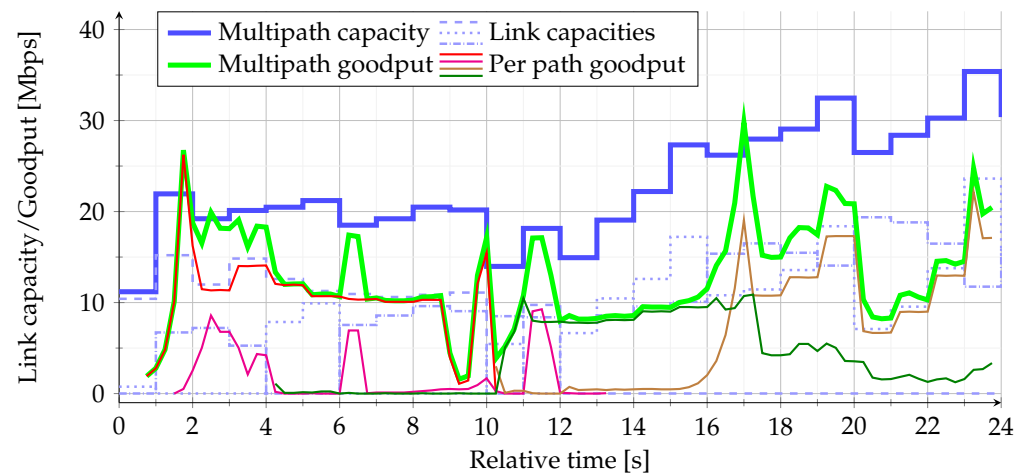
Case 2 opens up the possibility of parallel transmission. The client has two active interfaces at a given time, but it has to perform handovers when a higher capacity channel appears in the trace file. We can see from the tables that the goodput values are higher than in case 1 or case sp, they are close to the single path capacity. In one case (line 10 of Table 5) the average achieved goodput for Cubic is even higher than the corresponding single-path capacity. It is apparent that when having two paths for parallel transmission, MP-QUIC performs better than when only a single, best path is available.

A sample measurement is shown in Figure 7 shining light on some reasons why the goodput lags behind the multipath capacity. A handover happens at 4 s, but the new (green) path cannot achieve notable goodput. Similarly, it takes 3 s to the brown path to start increasing its goodput after the handover event at 12.

In cases 3 and 4 the client can access the best 3, 4 channels, respectively. The multipath capacity columns show that the extra available capacity is less and less with each additional channel. Cubic congestion control achieves higher goodput than the sp-capacity (line 14 of Tables 4 and 5). Moreover, a bit counterintuitively, providing more (low quality) links does not result in better performance, that is, case 4 exhibits worse performances than case 3.

All in all, multipath transmission outperforms the single path one, but it is not worth communicating on all the available links at least when using the default MP-QUIC settings. Also, the goodput values are far from the theoretical maximum (multipath capacity), so there's room for improvement. Slow links might impede fast links without large enough flow control windows resulting in head-of-line blocking. The performance can be improved with more sophisticated path selection methods for the initial packets (e.g., round-robin or min-RTT [34], acknowledgments (e.g., fastest-path acknowledgment [35]), and retransmis-

sions (e.g., FastCoRe [36]). Besides, changing the congestion control policy to not reset at access link changes could also be a way to improve the throughput.



**Figure 7.** A measurement for Case 2, BBR CC. algorithm (line 9 of Table 5).

The MPTCP experiments were performed using the same Mininet-based emulation environment applied in the case of Picoquic. Disappearing links were emulated by updating the state of the respective bottleneck links to down. We executed the emulation in a virtual machine having 1 vCPU and 3072 MB of memory, utilizing version 1 of the protocol (RFC 8684 [37]) implemented within the mainline Linux kernel of Fedora 34 with kernel version 5.15.5-100.fc34.x86\_64. Measurements were made with default (MP)TCP configurations and setting up iperf tests with fixed file sizes. TCP window sizes supplied to the iperf command were 85.3 kB (default) at the server and 5 MB (double of the default) at the client-side. The default TCP connection between the iperf endpoints was hijacked and converted to MPTCP using a systemtap script as per chapter 30 of the guide to configuring and managing networking in Red Hat Enterprise Linux 8 [38]. MPTCP endpoints were configured using the `mptcp` command of the `iproute2` package for setting up one subflow for each path between the client and the server in the tested network (i.e., 3 in the case of the client in the network shown in Figure 4). Application-level goodput was calculated using the `mptcp analyzer` tool [39].

Our MPTCP measurement results show that the best goodput can be reached by using the idealized single path (i.e., idx 21 in Tables 4 and 5), and adding multiple paths having quick changes in link capacities impedes the protocol to fully leverage the total capacity offered by the paths. In the residential environment, MPTCP can only approach the idealized single path performance when it has access to three paths (idx 25) while in the road environment, it is still far off from the sp performance. Comparing MPTCP's performance to that of Picoquic, we can observe that in both environments, MPTCP achieves similar goodput to Picoquic in the sp and 1 cases, while in all subsequent cases, it is significantly outperformed by Picoquic. We note that this behavior is due to slow adaption to available link capacities which might be improved upon by changing the interface state at the client instead of at the bottleneck link, and using the `mptcpd` [40] interface auto-configuration tool to configure the MPTCP interfaces. However, these aspects are not explored in this work due to the limitations of the measurement environment.

## 5. Conclusions

Multipath transport is a topic that has seen rapid evolution and has been studied extensively over the years. Most of the existing research is, however, focused on WiFi-cellular aggregation or handovers. In this paper, we study the potential of multi-operator cellular network aggregation in the transport layer. First, after a statistical analysis of LTE drive test data, we show that the average number of available cells motivates such an approach,

and also show the significant difference in the aggregated multipath capacity compared to single-path capacities over the different networks. Then we present a comprehensive performance evaluation of MP-QUIC using testbed measurements with a trace-based approach. Our results prove that MP-QUIC is able to outperform single-path solutions even over volatile cellular paths, however, we also highlight that further optimizations are needed to reach the theoretical limit.

**Author Contributions:** Conceptualization, Z.K., F.N., A.M., S.M. and G.P.; methodology, Z.K., F.N.; software, F.N., I.P.; formal analysis, Z.K.; investigation, Z.K., F.N., I.P., D.S.; resources, A.M., G.P.; writing—original draft preparation, Z.K., F.N., A.M., I.P.; writing—review and editing, S.M., G.P.; visualization, I.P.; supervision, A.M., S.M.; project administration, A.M., S.M. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by the Ministry of Innovation and Technology of Hungary from the National Research, Development and Innovation Fund through project no. 135074 under the FK\_20 funding scheme.

**Acknowledgments:** The authors would like to thank László Hévízi from Ericsson for his kind help.

**Conflicts of Interest:** The authors declare no conflict of interest

## Abbreviations

The following abbreviations are used in this manuscript:

TCP	Transmission Control Protocol
IETF	Internet Engineering Task Force
ATSSS	Access Traffic Steering, Switching, and Splitting
MP-QUIC	Multipath QUIC
WiFi	Wireless Fidelity
3GPP	3rd Generation Partnership Project
LTE	Long Term Evolution
BBR	Bottleneck Bandwidth and RTT
RTT	Round-trip Time
UE	User Equipment
eNB	Evolved Node B
LRF	Lowest RTT First
SOCKS	Socket Secure
DCCP	Datagram Congestion Control Protocol
RTO	Retransmission Timeout
OLIA	Opportunistic Linked Increase Algorithm
RSRP	Reference Signal Received Power
SINR	Signal to Interference plus Noise Ratio
CQI	Channel Quality Indicator
SP	Single Path

## References

- Langley, A.; Riddoch, A.; Wilk, A.; Vicente, A.; Krasic, C.; Zhang, D.; Yang, F.; Kouranov, F.; Swett, I.; Iyengar, J.; et al. The quic transport protocol: Design and internet-scale deployment. In Proceedings of the Conference of the ACM Special Interest Group on Data Communication, Los Angeles, CA, USA, 21–25 August 2017; pp. 183–196.
- Li, M.; Lukyanenko, A.; Ou, Z.; Ylä-Jääski, A.; Tarkoma, S.; Coudron, M.; Secci, S. Multipath transmission for the internet: A survey. *IEEE Commun. Surv. Tutor.* **2016**, *18*, 2887–2925. [[CrossRef](#)]
- Wu, H.; Ferlin, S.; Caso, G.; Alay, Ö.; Brunstrom, A. A Survey on Multipath Transport Protocols Towards 5G Access Traffic Steering, Switching and Splitting. *IEEE Access* **2021**, *9*, 164417–164439. [[CrossRef](#)]
- Merz, R.; Wenger, D.; Scanferla, D.; Mauron, S. Performance of LTE in a high-velocity environment: A measurement study. In Proceedings of the 4th Workshop on All Things Cellular: Operations, Applications, & Challenges, Chicago, IL, USA, 22 August 2014; pp. 47–52.
- Li, F.; Chung, J.W.; Jiang, X. Driving tcp congestion control algorithms on highway. *Proc. Netdev* **2017**, *2*, Article #7.
- Cardwell, N.; Cheng, Y.; Gunn, C.S.; Yeganeh, S.H.; Jacobson, V. BBR: Congestion-based congestion control. *Commun. ACM* **2017**, *60*, 58–66. [[CrossRef](#)]

7. Midoglu, C.; Kousias, K.; Alay, Ö.; Lutu, A.; Argyriou, A.; Riegler, M.; Griwodz, C. Large scale “speedtest” experimentation in Mobile Broadband Networks. *Comput. Netw.* **2020**, *184*, 107629. [[CrossRef](#)]
8. Zhang, M.; Polese, M.; Mezzavilla, M.; Zhu, J.; Rangan, S.; Panwar, S.; Zorzi, M. Will TCP work in mmWave 5G cellular networks? *IEEE Commun. Mag.* **2019**, *57*, 65–71. [[CrossRef](#)]
9. Raiciu, C.; Paasch, C.; Barre, S.; Ford, A.; Honda, M.; Duchene, F.; Bonaventure, O.; Handley, M. How hard can it be? Designing and implementing a deployable multipath TCP. In Proceedings of the 9th USENIX Symposium on Networked Systems Design and Implementation (NSDI 12), San Jose, CA, USA, 25–27 April 2012; pp. 399–412.
10. Nikravesh, A.; Guo, Y.; Qian, F.; Mao, Z.M.; Sen, S. An in-depth understanding of multipath TCP on mobile devices: Measurement and system design. In Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking, New York, NY, USA, 3–7 October 2016; pp. 189–201.
11. Polese, M.; Jana, R.; Zorzi, M. TCP in 5G mmWave networks: Link level retransmissions and MP-TCP. In Proceedings of the 2017 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Atlanta, GA, USA, 1–4 May 2017; pp. 343–348.
12. De Coninck, Q.; Bonaventure, O. Multipath quic: Design and evaluation. In Proceedings of the 13th International Conference on Emerging Networking Experiments and Technologies, Incheon, South Korea, 12–15 December 2017; pp. 160–166.
13. Viernickel, T.; Froemmgen, A.; Rizk, A.; Koldehofe, B.; Steinmetz, R. Multipath QUIC: A deployable multipath transport protocol. In Proceedings of the 2018 IEEE International Conference on Communications (ICC), Kansas City, MO, USA, 16–20 May 2018; pp. 1–7.
14. Khalili, R.; Gast, N.; Popovic, M.; Upadhyay, U.; Le Boudec, J.Y. MPTCP is not pareto-optimal: Performance issues and a possible solution. In Proceedings of the 8th International Conference on Emerging Networking Experiments and Technologies, Nice, France, 10–13 December 2012; pp. 1–12.
15. Hurtig, P.; Grinnemo, K.J.; Brunstrom, A.; Ferlin, S.; Alay, Ö.; Kuhn, N. Low-latency scheduling in MPTCP. *IEEE/ACM Trans. Netw.* **2018**, *27*, 302–315. [[CrossRef](#)]
16. Wu, H.; Alay, Ö.; Brunstrom, A.; Ferlin, S.; Caso, G. Peekaboo: Learning-based Multipath Scheduling for Dynamic Heterogeneous Environments. *IEEE J. Sel. Areas Commun.* **2020**, *38*, 2295–2310. [[CrossRef](#)]
17. Raiciu, C.; Niculescu, D.; Bagnulo, M.; Handley, M.J. Opportunistic mobility with multipath TCP. In Proceedings of the 6th International Workshop on MobiArch, Washington, DC, USA, 28 June 2011; pp. 7–12.
18. Hampel, G.; Rana, A.; Klein, T. Seamless TCP mobility using lightweight MPTCP proxy. In Proceedings of the 11th ACM International Symposium on Mobility Management and Wireless Access, Barcelona, Spain, 3–8 November 2013; pp. 139–146.
19. Kanagarathinam, M.R.; Singh, S.; Jayaseelan, S.R.; Maheshwari, M.K.; Choudhary, G.K.; Sinha, G. QSOCKS: 0-RIT Proxification Design of SOCKS Protocol for QUIC. *IEEE Access* **2020**, *8*, 145862–145870. [[CrossRef](#)]
20. Amend, M.; Bogenfeld, E.; Cvjetkovic, M.; Rakocevic, V.; Pieska, M.; Kassler, A.; Brunstrom, A. A Framework for Multiaccess Support for Unreliable Internet Traffic using Multipath DCCP. In Proceedings of the 2019 IEEE 44th Conference on Local Computer Networks (LCN), Osnabrück, Germany, 14–17 October 2019; pp. 316–323.
21. Paasch, C.; Detal, G.; Duchene, F.; Raiciu, C.; Bonaventure, O. Exploring mobile/WiFi handover with multipath TCP. In Proceedings of the 2012 ACM SIGCOMM Workshop on Cellular Networks: Operations, Challenges, and Future Design, Helsinki, Finland, 13 August 2012; pp. 31–36.
22. Fejes, F.; Rác, S.; Szabó, G. Application agnostic QoE triggered multipath switching for Android devices. In Proceedings of the 2017 IEEE International Conference on Communications (ICC), Paris, France, 21–25 May 2017; pp. 1–7.
23. De Coninck, Q.; Bonaventure, O. MultipathTester: Comparing MPTCP and MPQUIC in Mobile Environments. In Proceedings of the 2019 Network Traffic Measurement and Analysis Conference (TMA), Paris, France, 19–21 June 2019; pp. 221–226.
24. Sinky, H.; Hamdaoui, B.; Guizani, M. Proactive Multipath TCP for Seamless Handoff in Heterogeneous Wireless Access Networks. *IEEE Trans. Wirel. Commun.* **2016**, *15*, 4754–4764. [[CrossRef](#)]
25. Liao, B.; Zhang, G.; Wu, Q.; Li, Z.; Xie, G. Cross-layer Path Selection in Multi-path Transport Protocol for Mobile Devices. *arXiv* **2020**, arXiv:2007.01536.
26. Sathyanarayana, S.D.; Lee, J.; Lee, J.; Grunwald, D.; Ha, S. Exploiting Client Inference in Multipath TCP Over Multiple Cellular Networks. *IEEE Commun. Mag.* **2021**, *59*, 58–64. [[CrossRef](#)]
27. Alay, Ö.; Lutu, A.; Peón-Quirós, M.; Mancuso, V.; Hirsch, T.; Evensen, K.; Hansen, A.; Alfredsson, S.; Karlsson, J.; Brunstrom, A.; et al. Experience: An open platform for experimentation with commercial mobile broadband networks. In Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking, Snowbird, UT, USA, 16–20 October 2017; pp. 70–78.
28. Safari Khatouni, A.; Trevisan, M.; Giordano, D.; Rajiullah, M.; Alfredsson, S.; Brunstrom, A.; Midoglu, C.; Alay, Ö. An Open Dataset of Operational Mobile Networks. In Proceedings of the 18th ACM Symposium on Mobility Management and Wireless Access, Alicante, Spain, 16–20 November 2020; pp. 83–90.
29. Tomić, I.A.; Davidović, M.S.; Bjeković, S.M. On the downlink capacity of LTE cell. In Proceedings of the 2015 23rd Telecommunications Forum Telfor (TELFOR), Belgrade, Serbia, 24–25 November 2015; pp. 181–185.
30. Østerbø, O. Scheduling and capacity estimation in LTE. In Proceedings of the 2011 23rd International Teletraffic Congress (ITC), San Francisco, CA, USA, 6–9 September 2011; pp. 63–70.

31. Huitema, C.; Köher, B.; La Goutte, A.; Lubashev, I.; Wu, P.; Ferrieux, A.; Nardi, I.; Stewart, V.; Joshi, A.; Duke, M. et al. Picoquic, Version 48525d89ef4d. Available online: <https://github.com/private-octopus/picoquic> (accessed on 3 January 2022).
32. Lantz, B.; Heller, B.; McKeown, N. A network in a laptop: Rapid prototyping for software-defined networks. In Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks, Monterey, CA, USA, 20–21 October 2010; pp. 1–6.
33. De Coninck, Q. Minitopo: Scripts to Perform Easy Mininet Testing with Multipath Protocols; Version 7faea31e8ce0. 2020. Available online: <https://github.com/qdeconinck/minitopo> (accessed on 3 March 2022).
34. De Coninck, Q.; Michel, F.; Piraux, M.; Rochet, F.; Given-Wilson, T.; Legay, A.; Pereira, O.; Bonaventure, O. Pluginizing QUIC. In Proceedings of the ACM Special Interest Group on Data Communication (SIGCOMM '19), Beijing, China, 19–24 August 2019; Association for Computing Machinery: New York, NY, USA, 2019; pp. 59–74. [[CrossRef](#)]
35. Zheng, Z.; Ma, Y.; Liu, Y.; Yang, F.; Li, Z.; Zhang, Y.; Zhang, J.; Shi, W.; Chen, W.; Li, D.; et al. XLINK: QoE-Driven Multi-Path QUIC Transport in Large-Scale Video Services. In Proceedings of the 2021 ACM SIGCOMM 2021 Conference (SIGCOMM '21), Online, 23–27 August 2021; Association for Computing Machinery: New York, NY, USA, 2021; pp. 418–432. [[CrossRef](#)]
36. Hwang, J.; Walid, A.; Yoo, J. Fast Coupled Retransmission for Multipath TCP in Data Center Networks. *IEEE Syst. J.* **2018**, *12*, 1056–1059. [[CrossRef](#)]
37. Ford, A.; Raiciu, C.; Handley, M.; Bonaventure, O.; Paasch, C. TCP Extensions for Multipath Operation with Multiple Addresses, RFC 8684. 2020. Available online: <https://www.rfc-editor.org/info/rfc8684> (accessed on 3 March 2022). [[CrossRef](#)]
38. Red Hat, Inc. Getting Started with Multipath TCP. In *Red Hat Enterprise Linux 8 Configuring and Managing Networking*; Red Hat, Inc.: Raleigh, NC, USA, 2022; Chapter 30, pp. 193–198.
39. Coudron, M. Passive Analysis for Multipath TCP. In Proceedings of the Asian Internet Engineering Conference (AINTEC '19), Phuket, Thailand, 7–9 August 2019; Association for Computing Machinery: New York, NY, USA, 2019; pp. 25–32. [[CrossRef](#)]
40. Intel Corporation. Multipath TCP Daemon. 2022. Available online: <https://github.com/intel/mptcpd> (accessed on 3 March 2022).