

TECHNICAL UNIVERSITY OF MUNICH

DEPARTMENT OF INFORMATICS

Master's Thesis in Informatics

**Evaluating Transport Protocols for Remote
Piloting of Aerial Vehicles**

Hendrik Leonard Cech

TECHNICAL UNIVERSITY OF MUNICH

DEPARTMENT OF INFORMATICS

Master's Thesis in Informatics

**Evaluating Transport Protocols for Remote
Piloting of Aerial Vehicles**

**Evaluation von Transportprotokollen für die
Fernsteuerung von Luftfahrzeugen**

Author: Hendrik Leonard Cech
Supervisor: Prof. Dr.-Ing. Jörg Ott
Advisors: Dr. Nitinder Mohan and Aygün Baltaci, M.Sc.
Submission Date: 2021-07-26

I confirm that this master's thesis is my own work and I have documented all sources and material used.

Munich, 2021-07-26

Hendrik Leonard Cech

Acknowledgments

First, I would like to thank my supervisors Aygün Baltacı and Dr. Nitinder Mohan who gave me more support than I could have ever asked for. Your passion for our work motivated me to give my best. The next mango sauce falafel will be on me!

For making the drone measurements possible by providing the necessary equipment, I would like to express my gratitude towards Airbus. For the same reason, I would like to thank the Chair of Astronautics at the Technical University of Munich and in particular Nicolas Zunhammer. Without your willingness to lend us a spare drone, this thesis would look a lot different.

I would like to thank Moritz Thalmayr for providing us with the opportunity to conduct flights close to Türkenfeld. I am very grateful for your help that ended our desperate search for a rural takeoff location.

The work on the LTE emulator would not have been possible without help from a number of people. First, I would like to thank Paul Emmerich and Brenton Walker for their work on MoonGen and their helpful responses to my questions. Second, I would like to thank the administrators of the Chair of Connected Mobility for providing the infrastructure that the emulator was executed on. In particular, I would like to thank Simon Zelenski, who patiently fulfilled my change requests over and over again, as I came up with new requirements.

Furthermore, I would like to thank Julian Rodemann for the relaxing runs that we shared, for your emotional support, and your help on statistical matters.

I would like to thank my parents for their support throughout my studies. You gave me the confidence to trust my intuition with regard to my career path and enabled me to focus on it, for which I am very grateful.

Finally, thank you to my girlfriend Laura, for all her love, support, and encouragement throughout my studies and the work on this thesis.

Abstract

Electrically-powered aerial vehicles can be used in application areas such as transportation, agriculture, and emergency assistance. A reliable and available communication infrastructure is required to orchestrate and control Beyond Visual Line of Sight (BVLoS) missions. This thesis investigates the suitability of cellular communication for remotely controlling aerial vehicles, such as Unmanned Aerial Vehicles (UAVs) and Electrical Vertical Take-off and Landing Vehicles (eVTOLs). Furthermore, the performance of the transport protocols TCP, UDP, and QUIC is evaluated with regard to the requirements of remote-piloting.

We approached those questions by instrumenting a drone with up to four LTE-connected single-board computers to conduct field tests at altitudes of up to 200 meters above the ground. The network tests were designed to mimic the delivery of control commands to the vehicle and the transmission of live video to the pilot. We conducted 33 flights in two different locations and 148 tests over the network of two different mobile providers.

Our analyses reveal an impact of LTE handovers and altitude on the transport layer latency. Furthermore, packet losses happen regularly and often in temporal proximity to other packet losses. The performance of live video delivery is mixed and would often not suffice to remote-control an aerial vehicle. As an outlook, we present the work on a DPDK-based LTE emulator.

Based on our results, TCP is a better choice than UDP for remotely controlling an aerial vehicle over LTE. We observe that the live transmission of a high-quality video is a challenging task for LTE if a low delay is required. Further investigations are necessary to realize the potential of cellular technology for the remote-controlling of aerial vehicles.

Zusammenfassung

Elektrische Luftfahrzeuge werden für den Transport, die Landwirtschaft und für Hilfe in Notfällen eingesetzt. Um diese Einsätze außerhalb des Sichtfeldes zu organisieren und durchzuführen, wird eine verlässliche und verfügbare Kommunikationsinfrastruktur benötigt.

Diese Arbeit erforscht die Eignung von der Mobilfunktechnologie für die Fernsteuerung von Luftfahrzeugen, wie zum Beispiel Unmanned Aerial Vehicles (UAVs, deutsch: unbemannte Luftfahrzeuge) und Electrical Vertical Take-off and Landing Vehicles (eVTOLs, deutsch: elektrisch senkrecht startende und landende Luftfahrzeuge). Darüber hinaus wird die Leistung der Transport Protokolle TCP, UDP, und QUIC im Hinblick auf die Anforderungen des Fernsteuerns bewertet.

Die Fragen wurden bearbeitet, indem Feldmessungen mit einer Drone durchgeführt wurden. Diese wurde mit bis zu vier LTE-vernetzten Einplatinencomputern ausgestattet, um Feldmessungen in Höhen von bis zu 200 Metern über dem Boden durchzuführen. Die Netzwerktests wurden so konzipiert, dass sie den Transport von Steuerungsbefehlen an das Fahrzeug und die Übertragung von Live-Video an den Piloten nachahmen. Insgesamt wurden 33 Flüge an zwei verschiedenen Standorten und 148 Tests über das Mobilfunknetz von zwei verschiedenen Anbietern durchgeführt.

Die Auswertung zeigt den Einfluss von LTE Verbindungsübergaben (englisch: handovers) und der Flughöhe auf die Latenz der Transportschicht. Paketverluste finden regelmäßig statt und stehen häufig in zeitlicher Nähe zu anderen Paketverlusten. Die Leistung der Live-Videoübertragung ist durchmisch und würde häufig nicht ausreichen, um ein Luftfahrzeug fernzusteuern. Als Ausblick wird die Arbeit an einem DPDK-basierten LTE Emulator vorgestellt.

Ausgehend von den Ergebnissen dieser Arbeit ist TCP eine bessere Wahl als UDP für die Fernsteuerung von Flugfahrzeugen über LTE. Die Übertragung eines Live-Videos ist eine anspruchsvolle Aufgabe für LTE, wenn eine geringe Verzögerung gefordert wird. Weitere Untersuchungen sind nötig, um das Potenzial der Mobilfunktechnik für die Fernsteuerung von Flugfahrzeugen umzusetzen.

Contents

Acknowledgments	v
Abstract	vii
Zusammenfassung	ix
1 Introduction	1
2 Background & Related Work	3
2.1 Long Term Evolution (LTE)	3
2.1.1 Network Architecture	3
2.1.2 User Plane Protocol Stack	5
2.1.3 Reliability & Error Control	6
2.1.4 Handovers	6
2.1.5 Wireless Propagation	9
2.2 Aerial Vehicles	9
2.2.1 Communication with Aerial Vehicles	10
2.2.2 Challenges with Cellular Communication in the Air	11
2.3 Transport Protocols	11
2.3.1 TCP	12
2.3.2 UDP	13
2.3.3 QUIC	13
2.3.4 RTP	14
2.4 Related Work	14
2.5 Summary	16
3 Measurement Setup	17
3.1 Design	17
3.1.1 Test Location	17
3.1.2 Flight Trajectories	19
3.1.3 Selection of Transport Protocols	20
3.1.4 Performance Tests	20
3.1.5 Test Execution	21
3.2 Realization	22
3.2.1 Hardware Configuration	22
3.2.2 Software Configuration	24
3.2.3 Constant Throughput Tests	26

3.2.4	Video Transmission Tests	26
3.3	Summary	27
4	Results	31
4.1	Handovers	32
4.1.1	Detection of Handovers	32
4.1.2	Overview	32
4.1.3	Handover Correlation with Altitude & Velocity	34
4.1.4	Handover Interruption Time (HIT)	37
4.1.5	Summary	37
4.2	Packet Loss	39
4.2.1	Computation of Packet Losses	39
4.2.2	Number of Packet Losses	40
4.2.3	Packet Losses over Time	40
4.2.4	Summary	42
4.3	Network Latency	44
4.3.1	Calculation of Latency	44
4.3.2	Influence of Protocols and Transfer Direction	45
4.3.3	Influence of Signal Strength	47
4.3.4	Influence of Altitude	48
4.3.5	Influence of Handovers on Latency	49
4.3.6	Summary	51
4.4	Throughput	51
4.4.1	Comparison of Transport Protocols	52
4.4.2	Influence of the Test Location	53
4.4.3	Performance Comparison of the Mobile Networks	53
4.4.4	Influence of the Flight Trajectory	55
4.4.5	Influence of Altitude and Velocity	56
4.4.6	Summary	59
4.5	Video Streaming	59
4.5.1	Ground Tests	59
4.5.2	Standalone RTP Tests	61
4.5.3	Coupled RTP Tests	62
4.5.4	Summary	62
4.6	LTE Emulation	62
4.6.1	Emulation Setup	63
4.6.2	Emulation of Handovers	64
4.6.3	Performance Evaluation	65
5	Discussion	67
5.1	Reliability	67
5.2	Video Streaming	68
5.3	Transport Protocols	69

6 Conclusion	71
6.1 Limitations & Future Work	72
Glossary	74
List of Figures	76
List of Tables	78
Bibliography	79

1 Introduction

Novel, electrically operated aerial vehicles can be used in a variety of commercial and governmental applications: Transpiration, delivery, agricultural services, critical infrastructure, environmental monitoring, and emergency assistance are only some of the envisioned uses [41, 60, 34]. Especially in urban settings, the use of small and relatively low-cost UAVs and eVTOLs is expected to increase in the coming years: The US company Archer targets a commercial launch of its eVTOL for passenger transportation in 2024 [10]. Commonly, applications such as the inspection of pipelines or exploration of a wildfire, require the operation of drones over a range of hundreds of kilometers. For these operations, that are executed beyond the pilot's Visual Line of Sight (VLoS), access to real-time high-quality video and flight information is required. The transmission needs to be highly reliable, especially if the drone is operated close to critical infrastructure or humans.

The radio control technology used by commercial drones can bridge distances up to a couple of kilometers between pilot and drone, but requires an unobstructed environment that is free of inferences. This communication is typically conducted on unlicensed 2.4 and 5.8 GHz frequencies using IEEE 802.11 or proprietary technologies [103, 39]. To scale the execution of beyond line-of-sight flights, the spatial position of pilots and vehicles must be decoupled. A communication technology should provide the necessary infrastructure to allow drone operations—at best—anywhere on earth.

One such option is Satellite Communication (SATCOM). While the baseline latency of Geosynchronous Equatorial Orbit (GEO) and Medium Earth Orbit (MEO) satellite constellations can not satisfy the requirements of remote piloting, Low Earth Orbit (LEO) broadband services could be sufficiently performant. World-wide availability, high throughput, and constant latency make this technology attractive for remote piloting [108, 90]. The high power requirements, size, and weight of satellite terminals, do however pose a challenge to light-weight UAVs and eVTOLs [13].

Another option for the BVLoS operation are cellular networks, whose modems are designed to be small and have modest energy needs, to work in mobile phones. Furthermore, today's cellular network are widely available: 4G technology is deployed in 240 countries and serves 62% of mobile users worldwide, while 5G is already available in 70 countries [33]. The evolution from 4G to 5G promises an additional increase in peak data rates (from 1 to 20 Gbps) and a drop in link layer latency (from 10 to 1 ms) [14]. Due to its availability and ongoing development, cellular technology is a promising approach to support the location-independent operation of aerial vehicles.

The performance of cellular-connected applications is influence by the chosen transport protocol. Research shows for example, that this choice influences the video stream-

ing performance in mobile settings [82], the loading times of websites [66], and the effect of LTE handovers [107]. It is therefore essential to explore the performance of different transport protocols for the use case of remotely controlling aerial vehicles.

The performance of LTE has already been measured in mobile settings on the ground [5] and in the air [63]. In both settings, it is able to provide service, but the operation at altitude increases the Radio Frequency (RF) channel interference levels at the flying vehicle and on the ground [69]. Considering those challenges, their performance impact on the transport-layer, which could, e.g., carry control commands, is intriguing. Sending live video from an aerial vehicle over an IEEE 802.11 link has been studied in field tests [64, 46]. The performance of a cellular link for this use case has not been thoroughly investigated. Furthermore, the impact of transport protocols on the reliability and latency for the delivery of time-sensitive control information, has not been investigated.

We contribute insights about the real-world performance of different transport protocols that are carried over LTE for the communication requirements of remotely-controlled drones. Our measurement study uses a drone to compare the transport protocols TCP, UDP, and QUIC [44] on two different provider networks in an urban and a rural test location at altitudes up to 200 m. We analyze the performance of the transport link and the tested transport protocols, with regard to (i) how rapidly and reliably the transport of critical control information can be conducted, as well as (ii) how well, in terms of playback latency and rate, live video captured at the drone can be transmitted to a pilot.

To answer those questions, Chapter 2 introduces the architecture and mechanisms of LTE that are relevant to this thesis. Furthermore, the requirements and challenges of cellular-connected aerial vehicles are discussed before the relevant transport protocols are introduced. Chapter 3 outlines our testing methodology and describes the configured hardware and software setup. Chapter 4 presents the data from our measurement campaign and derives results. The discussed aspects are handovers, packet loss, latency, throughput, and video streaming performance. Furthermore, an approach to emulating an LTE link is presented. Chapter 5 discusses the implications of the measurement results in view of the communication requirements of remotely operated aerial vehicles. Finally, Chapter 6 summarizes our findings and highlights opportunities for future research.

2 Background & Related Work

Our research is conducted at the intersection of multiple topics. In the following, we describe the concepts of the topics that are relevant to this thesis. Those topics are the wireless data transmission standard Long Term Evolution (LTE) (described in Section 2.1), the communication needs of aerial vehicles (described in Section 2.2), and transport protocols (described in Section 2.3).

2.1 Long Term Evolution (LTE)

Long Term Evolution (LTE) is a standard for radio communication that is widely deployed in countries around the world [33]. The technology was specified by the 3rd Generation Partnership Project (3GPP) that was established in 1998 by different telecommunications associations. The organization started working on a 3G technology specification that was realized by systems such as Universal Mobile Telecommunications System (UMTS). Motivated by the increasing demand for higher data rates and Quality-of-Service (QoS), the 3GPP started working on two parallel projects: LTE and System Architecture Evolution (SEA), which are together known as the Evolved Packet System (EPS); both are included in 3GPP Release 8 (2008) and have been extended in Release 9 (2009). The system does not fully comply with the requirements of 4G that is specified by the International Telecommunication Union's Radio Section (ITU-R). It is therefore informally known as 3.9G. The 4G-compliant standard LTE-Advanced was published in 3GPP Release 10 in 2011 [6].

LTE works over a multitude of frequency bands that are in between 410 and 5900 MHz. Different frequency bands are in use in different countries. The spectrum bands are grouped into Component Carriers (CCs) which are up to 20 MHz wide. Carrier aggregation is introduced by the LTE-Advanced standard where multiple Component Carriers (CCs) are used in tandem to operate on a bandwidth of up to 100 MHz. The CCs do not have to be contiguous but can be aggregated over different spectrum bands. Carrier aggregation's main goal is to improve the data rates that can be transported [6]. Complete information about the frequency bands and their aggregation capabilities are described in [2]. The 4G requirements published by the ITU-R state a peak speed requirement of 100 Mbps for mobility scenarios and 1 Gbps for stationary settings [42].

2.1.1 Network Architecture

The overall system is the EPS which is visualized by Figure 2.1. It consists of a network part, the Evolved Packet Core (EPC), and a radio network part called Evolved UMTS

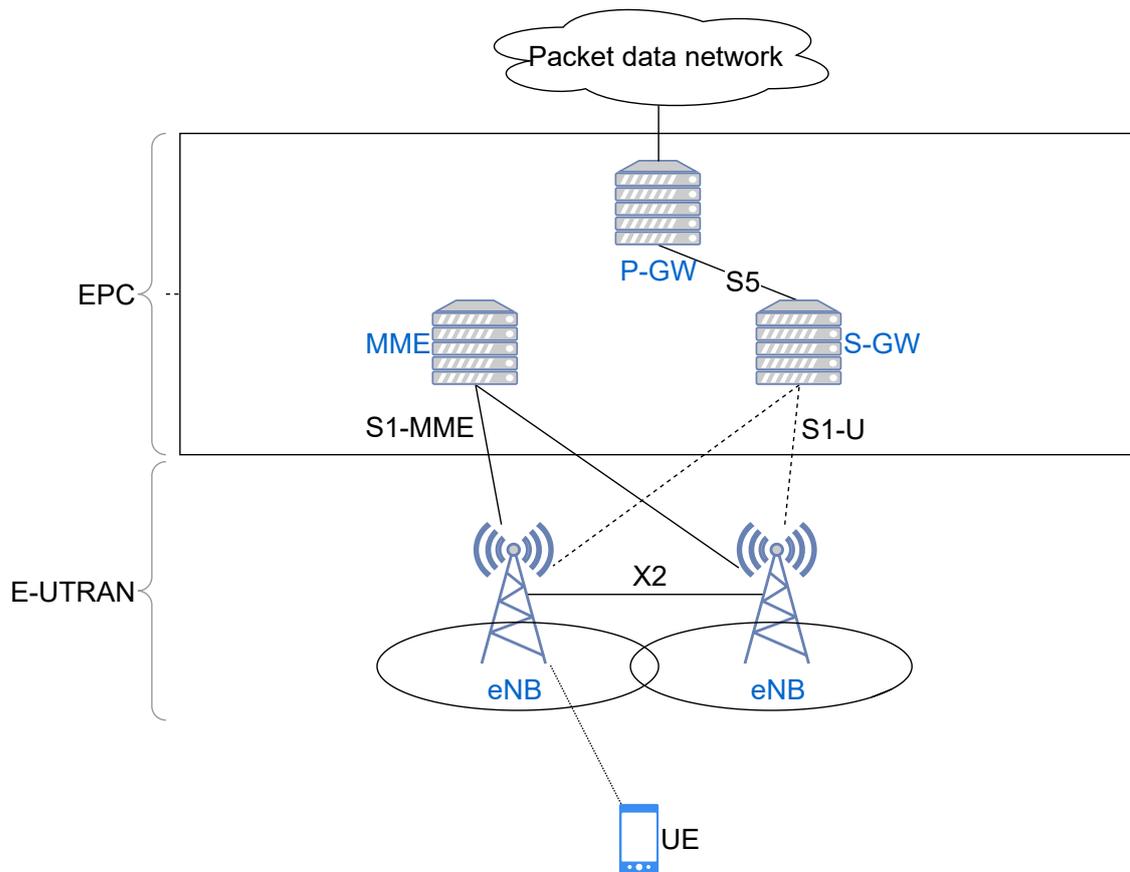


Figure 2.1: The LTE architecture (adapted from [53, 38, 6]).

Terrestrial Radio Access Network (E-UTRAN). The EPC serves a common infrastructure for multiple different 3GPP radio access technologies, e.g., UMTS, HSPA, HSPA+, and LTE. This is a key prerequisite for the seamless operation of different Radio Access Technologies (RATs) as it enables inter-3GPP handovers, i.e., handovers between different RATs. The EPC has a control plane that is constituted by the Mobility Management Entity (MME) and a user plane that encompasses the Serving Gateways (S-GWs) and the Packet-data network Gateways (P-GWs). The radio-access network consists of radio base stations, denoted as Enhanced Node Bs (eNBs), which communicate with the mobile terminals or User Equipments (UEs). The eNBs communicate with each other over the X2 interface and with the EPC over the S1 interface. The geographical area that is covered by an eNB is a *cell* [52].

While using the network, an UE is associated with an unique S-GW that handles local inter-eNB handover and inter-3GPP mobility. The P-GW assigns an IP address from the Packet-data network (PDN) to the UE. The communication between an UE and the user-plane entities occurs over an eNB [53, 6].

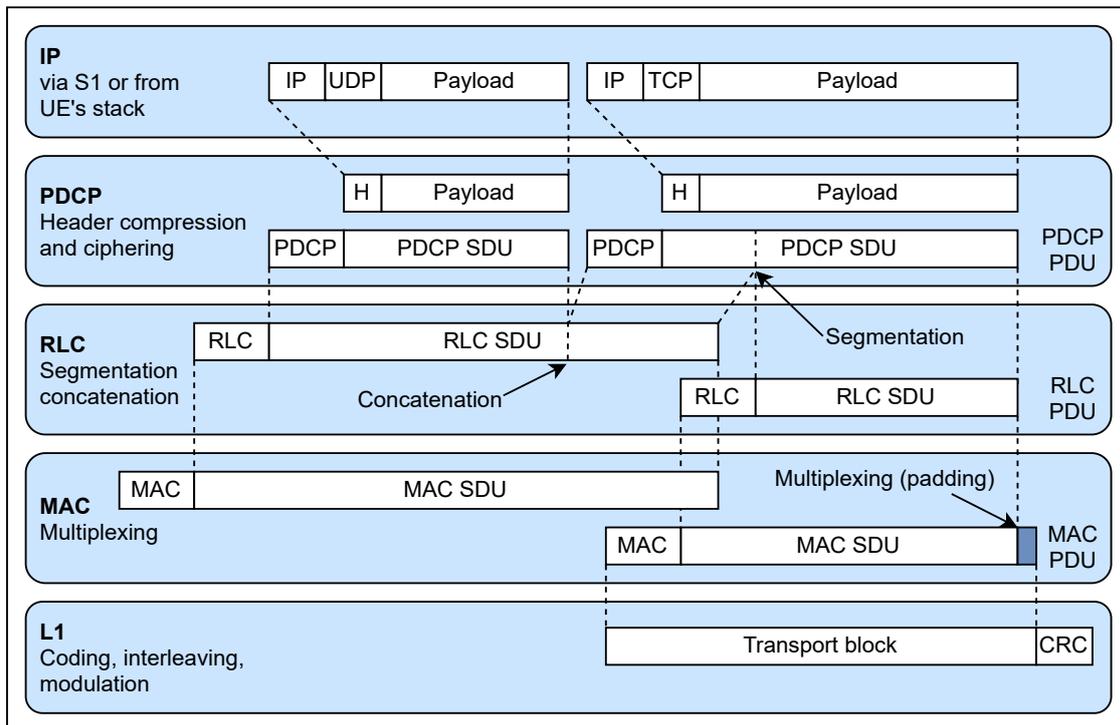


Figure 2.2: The lower layers of the LTE protocol stack (taken from [53, Figure 3]).

2.1.2 User Plane Protocol Stack

On a protocol level, the LTE stack is divided into the Access Stratum (AS) and the Non-Access Stratum (NAS). The AS is responsible for transporting data over the radio network. The NAS is a layer between the core network and the UE that manages control tasks such as authentication, registration and sessions. The messaging related to those tasks is conducted using the Radio Resource Control (RRC) protocol [6].

To improve the communication between an UE and an eNB, the link and physical layers of LTE are tasked with improving reliability, increasing security, maintaining integrity, and providing functionality for medium access. The layer is built to carry Internet Protocol (IP) packets to the EPC and receive them from the EPC. The transformation of messages is depicted in Figure 2.2.

On the lowest level, data is sent as a bit stream on the physical layer while being protected by a Cyclic Redundancy Check (CRC) and turbo-coding. The link-layer is divided into three sub-layers: the Packet Data Convergence Protocol (PDCP) layer, the Radio Link Control (RLC) layer, and the Media Access Control (MAC) layer. PDCP's main task is IP header compression. The subjacent RLC layer applies error correction using Automatic Repeat reQuest (ARQ) and supports data segmentation and concatenation. It also reorders Protocol Data Units (PDUs) based on their RLC sequence numbers. Finally, the MAC sublayer applies another level of error correction using Hybrid ARQ (HARQ) and provides functionality for medium access [53].

2.1.3 Reliability & Error Control

LTE was designed with the performance characteristics of reliable transport layer protocols in mind. The performance of the Transmission Control Protocol (TCP) quickly degrades for example with increasing packet loss probability; it only performs well with low packet loss (10^{-5} to 10^{-7} Packet Error Rate (PER)) [61, Figure 1] and its performance has an inverse proportional relationship with Round-Trip Time (RTT). Based on those requirements, LTE does not propagate bit errors to higher layers but instead drops and retransmits data. On the physical layer, a 24-bit CRC is used to detect transmission errors. The error correction is performed by a two-level scheme that aims to achieve reliability with low latency and low overhead. The majority of packet errors are corrected by a HARQ scheme on the MAC layer while an additional ARQ mechanism at the RLC layer is only invoked if HARQ fails.

The HARQ scheme on the MAC sublayer is fast and relatively light-weight. It uses multiple stop-and-wait HARQ processes instead of a single stop-and-wait scheme so that data can be transmitted continuously. Retransmissions are requested by a single HARQ bit instead of a sequence numbering scheme. Each PDU is positively or negatively acknowledged (ACK/NACK). The acknowledgments have a fixed-timing relation to the corresponding transmission attempt, i.e., under normal operation, the acknowledgments are received at a fixed time offset after the packet's transmission. The acknowledgment scheme is not foolproof and ACK/NACK misinterpretations lead to a residual packet loss rate of 10^{-4} to 10^{-3} . Additionally, errors in certain other control signaling mechanisms can lead to a failure of HARQ.

The ARQ scheme on the RLC sublayer complements the HARQ scheme on the MAC sublayer by providing reliability in case of HARQ failure. The RLC sublayer orders PDUs by their RLC sequence number. In case of a gap and after a reordering timer expires, the retransmission of a certain lost PDU is requested by specifying its RLC sequence number. The combination of HARQ and ARQ results in low latency and low overhead scheme that does not sacrifice reliability. It constructs a network link that actively tries to improve the performance of reliable transport protocols such as TCP [53].

2.1.4 Handovers

For the duration of their usage of the mobile network, UEs are always attached to one eNB. Either due to a UE location change or due to a management decision of the cellular provider, the currently selected cell might no longer be the best choice. The UE can change its cell association by performing an *handover*. LTE's handover mechanism follows the *break-before-make* paradigm where an UE is only ever connected to at most one base station. A brief period during which the UE is unable to send or receive data is therefore unavoidable [93]. This time period is called Handover Interruption Time (HIT). The handover procedure is depicted in Figure 2.3. It can be divided into three phases: Handover preparation, execution, and completion.

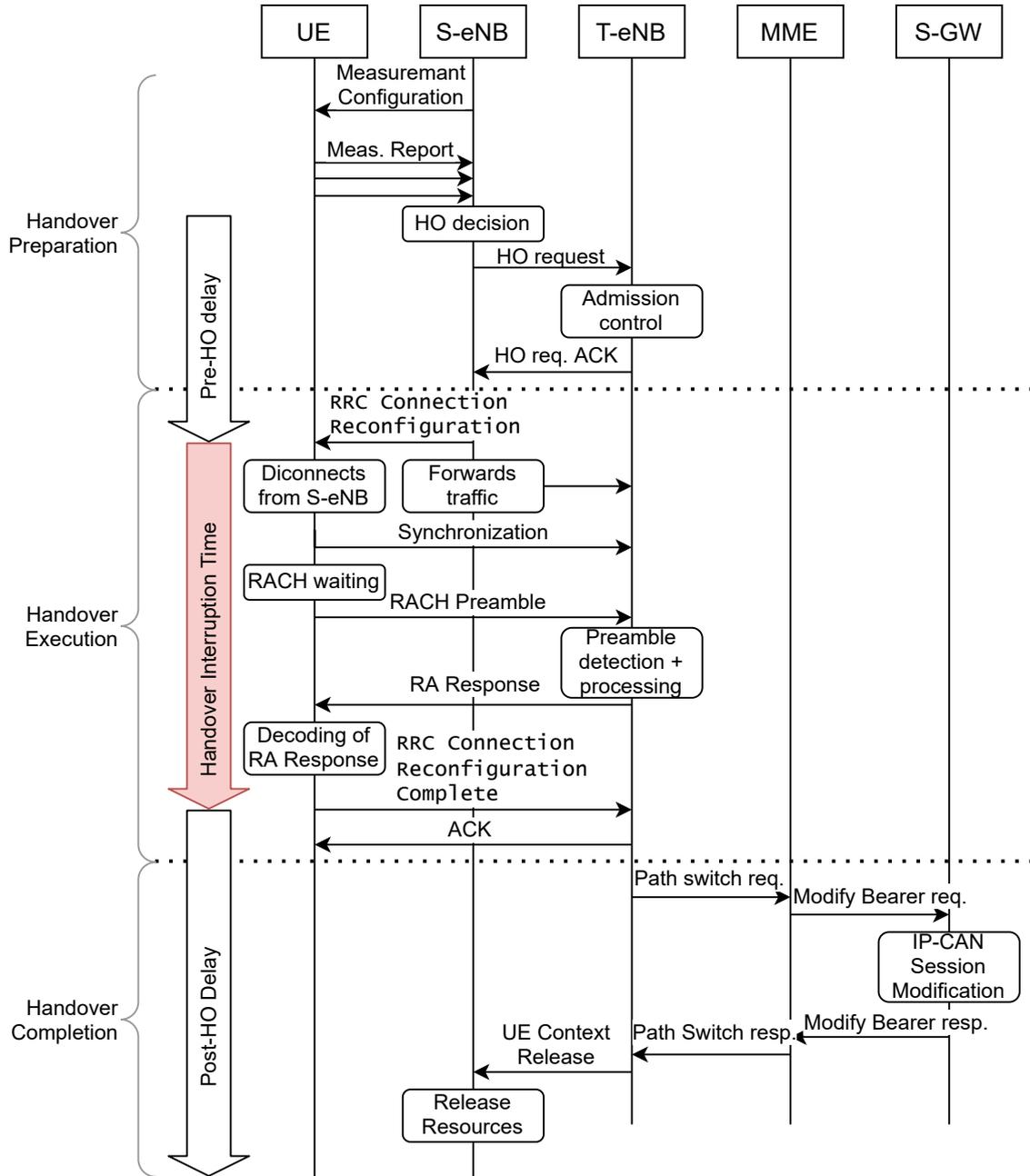


Figure 2.3: The LTE handover process. Adapted from [38][Figure 1] and [93][Figure 3].

Periodically, the UE performs downlink (DL) signal strength measurements of reference signals from its Serving Enhanced Node B (S-eNB) and neighboring eNBs to notice when an handover is necessary. The eNB prescribes under which conditions the UE shall send a notification with the signal strength of the surrounding eNBs using a RRC Connection Reconfiguration message. This notification is the *measurement report* which contains the signal strengths of the S-eNB and neighboring eNBs. The notification conditions are formalized by 3GPP as *events* [3, Section 5.5.4], where the A3 event plays a key role. The A3 event is triggered if a neighboring eNB's signal strength becomes higher than the serving eNB's signal strength including a hysteresis margin. The margin increases the stability and prevents false-positive events from being triggered frequently. If a configured event takes place, the UE compiles a measurement report containing the measurement results and sends it to its S-eNB to kick-off the handover preparation phase. Based on this information, the S-eNB decides if a handover to another eNB (the Target Enhanced Node B (T-eNB)) should be performed.

If a positive handover decision is made, the handover execution phase is started as the S-eNB sends a HO request to the T-eNB. This component performs *admission control* to allocate resources for the UE. A HO request acknowledgment is returned to the S-eNB which starts to forward UE-traffic to the T-eNB. Forwarding is done directly between the eNBs over the X2 interface. Additionally, the S-eNB sends a RRC Connection Reconfiguration message to the UE that contains the necessary information to perform the handover. The UE detaches from the old cell and performs the Random Access CHannel (RACH) procedure to synchronize with the T-eNB. This procedure results in timing information which the UE needs to schedule its uplink transmissions. Once the procedure is complete, the UE sends a RRC Connection Reconfiguration Complete message to the T-eNB to confirm the handover execution. The T-eNB starts to send buffered data to the UE which ends the HIT. The handover is completed from the view-point of the UE. The core network also finishes the handover procedure by registering the eNB relocation with the MME and S-GW if necessary [93].

The signal quality that the UE measures and compares to the thresholds sent by its S-eNB is derived from the measurement of specified reference signals. The results of signal level and quality are reported in different measures: The average measured received power without interference and noise components is the Reference Signal Received Power (RSRP) while the Received Signal Strength Indication (RSSI) is the average received power including interference and noise components. The Signal to Interference plus Noise Ratio (SINR) is calculated as the the strength of the desired signal divided by the noise. Han et al. measured the handover duration and found that HIT takes 21.08% (about 18 ms) of the total HO delay on average [38]. Most of the interruption delay is taken by Random Access (RA) even though resources are pre-allocated to the UE such that the process is free from contention [38].

Handovers are not always successful and can fail due to poor timing. If a handover happens too early, an immediate switch back to the previous serving eNB may happen as it's signal strength may appear to be higher greater again (a *ping-pong* handover).

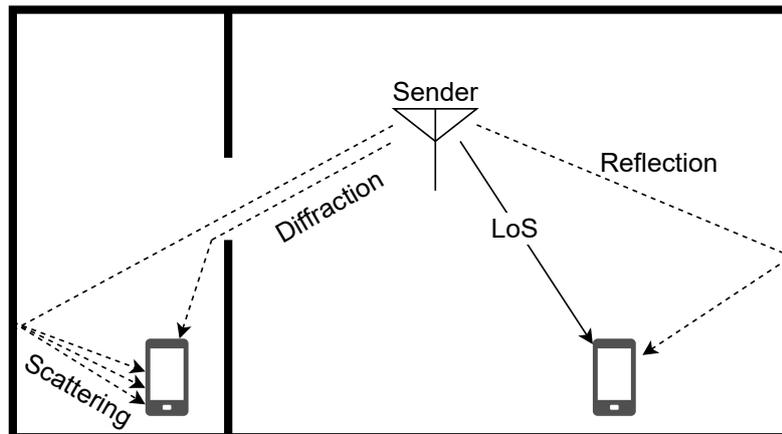


Figure 2.4: The propagation methods that are relevant for mobile broadband networks. Adapted from [40, Figure 5.30].

If a handover happens too late, the S-eNB channel may break before the handover is complete [93].

2.1.5 Wireless Propagation

A transmitted radio signal can reach the receiver by different propagation methods: Among those are Line of Sight (LoS), reflection, diffraction, and scattering that are depicted in Figure 2.4. Line of Sight (LoS) propagation requires that no obstructions are along the direct path from the sender to the receiver. The existence of the other three propagation methods allow a receiver that does not have a LoS to the sender, to receive the signal. A signal losses power along its path. The path length therefore determines the strength of the signal at the receiver and when it is receives. Among the propagation methods, the LoS signal has the shortest time delay and is therefore usually the strongest signal. If a signal reaches a point by multiple paths, we speak of *multipath* reception. The effect of multipath on the strength of a signal is environment-specific and hard to predict [40].

2.2 Aerial Vehicles

The advance of technology has enabled a new generation of battery-powered aerial vehicles. On the one hand, small-scale Unmanned Aerial Vehicles (UAVs) are being utilized for natural disaster monitoring, delivery of goods, and construction. Those vehicles have multiple rotors and are usually equipped with Inertial Measurement Units (IMUs), Global Positioning System (GPS) antennas, and other sensors [39]. The European Union provides unified regulations for drones with a maximum take-off weight of 25 kg [27]. On the other hand, larger Electrical Vertical Take-off and Landing Vehicles (eVTOLs) are designed to transport people over middle-long distances. The target

operation area of these “flying taxis” are cities. eVTOLs are designed to be controlled from the ground [28].

2.2.1 Communication with Aerial Vehicles

UAVs are available to consumers starting from a low price point. They are most often controlled by a hand-held remote controller that communicates directly with the drone over a point-to-point radio channel. While the most simple controller only requires unidirectional data flow to send control commands to the drone, the communication with most drones happens both-ways to relay status information such as altitude and battery level to the operator. In case the aerial vehicle is controlled beyond the pilot’s range of vision (BVLoS), the feedback channel becomes essential and a live video feed from the flying object’s point of view is usually required. The wireless technologies that are used to communicate with the aerial vehicles usually operate over the unlicensed 2.4 GHz and 5 GHz frequency bands. Drone manufacturers such as DJI developed proprietary transmission systems. The use of different standardized wireless technologies such as Bluetooth and Wi-Fi is compared in [39, Table IV].

Throughout this thesis, the communication channel from the aerial vehicle to the operator is the *uplink* (UL). In this direction, telemetry, payload data, and typically also live video data is sent from the UAV to the pilot. Telemetry typically includes flight altitude, position, and velocity. The reverse direction—from pilot to the controlled vehicle—is the *downlink* (DL) over which typically only control information is sent. This type of traffic is called Control and Non-Payload Communication (CNPC). The information are either flight instructions for manually controlled UAVs or waypoint updates for (semi-)autonomous UAVs [105].

Telecommunication industry organizations have published think pieces about the use of cellular networks for aerial vehicles. Two of those organizations are the Next Generation Mobile Networks Alliance (NGMN), which is a forum of mobile network operators [67], and the 3GPP, which is the standards organization that maintains telecommunication standards such as LTE [1]. Both organizations state that CNPC traffic must be transported over a highly available and reliable channel with low latency. The NGMN Alliance sets the bar at >99.9999% availability, >99% area coverage and <100 ms latency [68]. The 3GPP sets similar connectivity service requirements with a 50 ms latency target for CNPC and a maximum PER of $< 10^{-3}$ [4]. In addition, they allocate 60–100 Kbps for CNPC and up to 50 Mbps for uplink (UL) data communication. The real-world bandwidth usage of commercial drones have been analyzed by Baltaci et al. They recorded the traffic of three manually controlled consumer drones and found that up to 130 Kbps were served on the UL. Video and telemetry data occupied up to 3.5 Mbps on the DL [12].

2.2.2 Challenges with Cellular Communication in the Air

Terrestrial and aerial UEs are served by the same infrastructure but are not exposed to the same conditions because of their different relative distances to the ground. The first difference is that flying UEs operate outside of the optimized reception area of base stations. The antennas of most cells are down-tilted to best serve ground-based UEs. This design reduces the interference power level to other cells but also means that flying UEs are served by the base station's side lobes. Compared to the antenna's main lobe, the received signal power is lowered [58]. A second identified difference between terrestrial and aerial UEs is the Line of Sight (LoS) probability, i.e., how often a non-obstructed line exists between an UE and a base station. Radio waves are diverted by obstacles such as trees or buildings. The sender has to rely on other objects reflecting the waves so that they eventually reach the receiver (Non Line of Sight (NLoS) reception).

Lin et al. analyzed the LoS probability in the context of a rural setting [58]. With rising UE altitude, the LoS probability increased for a given distance to the base station. The LoS probability decreased below one only for distances greater than one kilometer for UEs flying higher than 50 m. According to their simulation, any UE at least 50 m off the ground is very likely to have a LoS to a base station. The upside of a LoS is decreased path loss (free space propagation) which increases the received signal power. This can even offset the base stations antenna side lobe gain reduction [58].

The downside of the high LoS probability of aerial UEs is down- and uplink interference. A flying UE is likely to have a LoS to *multiple* base stations which might transmit on the same frequencies. The 3GPP performed measurements on an aerial UE flying at 50 m or above and registered a high level of DL interference by up to 16 cells [4]. Furthermore, aerial UEs are shown to degrade the throughput performance of terrestrial UEs by causing uplink interference. Due to them having LoS to more cells, their uplink also causes interference to more cells [4]. These findings were confirmed by Kovacs et al. who performed LTE measurements using an UAV at altitudes up to 120 m. Several interfering signals were detected by an UAV in the air. In addition, almost twice as many cells are interfered by the drone if it is flying compared to being on the ground [49].

2.3 Transport Protocols

One goal of this thesis is to investigate which transport protocols are most suited to support the operation of remotely controlled UAVs. We focus on the two most established and widely used transport protocols TCP and UDP. In addition, we include QUIC into our considerations which has received a lot of attention in recent years and has been standardized in 2021 [44]. Video transmission has been identified as a key requirement for remotely controlling of aerial vehicles. A long-standing and widely used technology for the purpose of transporting real-time media is the Real-Time Transport Protocol (RTP). Based on its availability and stability, we have selected RTP to study the performance of video transmission. Subsequently, the protocols that are relevant to this thesis are introduced.

2.3.1 TCP

The Transmission Control Protocol (TCP) was first formally defined in September 1981 [98] and was extended over time; RFC 4614 [25] provides a summary of all extending RFCs. It is built on top of the Internet Protocol (IP) which together form the *Internet protocol suite* (TCP/IP). The addressing scheme of IP, which identifies hosts, is extended with *ports* that match packets to applications. TCP's longevity has made it the dominant transport protocol on the Internet: its share is at least 87% in Internet traffic data sets from 1999–2007 [55].

TCP's goal is to provide applications with a reliable byte stream interface that allows communication of two parties over an unreliably network. The application is not aware of the size or boundaries of individual IP packets. To build that abstraction, TCP uses sequence numbers to bring messages in the right order and to detect lost packets. TCP does not use advanced error recovery mechanism such as Forward Error Correction (FEC) but instead retransmits lost or erroneous packets.

TCP takes precautions to avoid stressing the network and receiver beyond their capabilities. TCP's Congestion Control (CC) sets an upper bound on the number of concurrently unacknowledged data packets that is progressively increased until a congestion event (usually packet loss) is registered. In response, the CC algorithm shrinks its *congestion window* to reduce the number of in-flight packets. CC was retroactively added to TCP as a mean to improve the overall performance of a network [45]. One reason for packet loss are full queues of overloaded hosts along the network path. In this case, packet loss is an accurate signal of congestion. Packets can however also be lost during transmission between two hosts, e.g., due to interference on a WiFi channel, which causes an unnecessary hit to TCP's performance. Alternative signaling mechanisms such as Explicit Congestion Notification (ECN) exist that accurately indicate congestion. TCP's mechanism to respect the receiver's processing capabilities (*flow control*) is realized in form of the *receive window*. Analogous to the congestion window, the maximum unacknowledged data is limited. The receiver continuously updates this limit and the sender adapts its send rate to adhere to it.

Over time, TCP has been extend to adapt to higher data rates, new applications, and new physical transport mediums. Nevertheless, conditions under which TCP performs sub-optimally remain. TCP's connection setup takes two round-trips; in case the connection is encrypted by TLS, another 1–2 round-trips are required. Data can only be served after the connection setup has been completed. For short-lived connections, those round-trips can take a significant share of the total connection time. This issue can for example delay the display of a website as the HyperText Transfer Protocol (HTTP) works upon TCP.

For some use cases, the design of TCP's byte stream interface can cause unnecessary delays. If a packet is lost, TCP must wait for the retransmission of that packet to be able to deliver all packets in order; this is known as Head-of-Line (HoL) blocking. For applications that are only every interested in the latest packet, this mechanism causes undesirable variations in latency; examples are video games or phone calls [51], [92,

Chapter 6.5], [36][Chapter 2].

2.3.2 UDP

The User Datagram Protocol (UDP) is a connection-less and “fire-and-forget” protocol. No handshake and no similar mechanism is necessary to initiate a connection on the transport layer. The transport protocol is described in RFC 768 [76] and assumes that IP is used as the underlying protocol. Similar to TCP, UDP adds the notion of source and destination ports to the address carried by the IP layer. Furthermore, the UDP header includes a checksum and a 16-bit length field. The maximum size of an UDP packet is however limited to 65,515 bytes due to the overhead of the underlying IP layer. In contrast to TCP, UDP does not implement any flow or congestion control. If desired, the application needs to implement measures such as reordering and retransmission [92, Chapter 6.4].

Even though UDP provides weaker guarantees than TCP to the application, it may be a more suitable choice for some application areas. Fundamentally, TCP offers a stream abstraction to the application while UDP exposes the underlying packet-based transport. The application has more control over what data is sent, and when. UDP does also not suffer from TCP’s Head-of-Line (HoL) blocking issue as lost packets must not necessarily be resent. This makes UDP attractive for real-time applications where retransmitted data may already be irrelevant once it finally arrives. A consequence of UDP’s lack of built-in congestion control is that it exhibits no fairness towards other protocols over the same link [29].

2.3.3 QUIC

A only recently standardized addition to the transport protocol space is QUIC [44]. The protocol aims to support similar use cases as TCP while improving on its shortcomings: It is also a connection-oriented protocol that provides a byte stream interface to applications by handling message reordering and retransmissions under the hood.

One targeted problem area is TCP’s initial three-way handshake that delays the time until data can be transmitted. If the connection is encrypted using Transport Layer Security (TLS), the connection establishment phase requires 3–4 RTTs [36, Chapter 4]. QUIC only supports encrypted connections and integrates the exchange of secrets into its handshake. Doing so, it trims the establishment phase to one round-trip in case of a fresh connection but furthermore supports immediate data transmission (*0-RTT*) for repetitive connections. Another feature tries to improve on TCP’s HoL blocking issue: QUIC supports the creation of *streams* which are individually flow and congestion controlled. Problems on the transmission of one stream do not interfere with the progression of other streams. Furthermore, QUIC connections are identified by a persistent connection ID instead of the quadruple of socket identifiers (IP addresses and port numbers). This difference is the base for QUIC’s *connection migration* support: If a smartphone for

example switches from a cellular to a Wi-Fi connection, an existing QUIC connection can be maintained and must not be reconstructed [50].

QUIC is built on top of UDP to avoid the problems caused by *transport layer ossification*. Ossification refers to the issues that prevent the adoption of new transport protocols, such as the deployment of middleboxes by network operators that only process TCP and UDP packets [72]. Work on a multitude of extensions to QUIC is ongoing [48]. Similar to Multipath TCP (MPTCP), a multipath mode is proposed: Multipath QUIC (MPQUIC) [20]. Another example is a datagram mode that would disable some of QUIC's features, such as packet retransmission and reordering, while keeping other features enabled, such as encryption [74]. Measurements that compare QUIC and TCP are inconclusive with regard to throughput, but attest improvements when looking at latency. The protocols performance seems to be most impacted by the choice of congestion algorithm and the robustness of the implementation [104].

2.3.4 RTP

The Real-Time Transport Protocol (RTP) is defined in RFC 3550 [84] in conjunction with the RTP Control Protocol (RTCP). It is used for many multimedia and real-time applications, such as telephony and video delivery. RTP typically runs on top of UDP and IP but can also be transported by TCP (as standardized in RFC 4571 [54]). RTP adds meta-information to each data chunk, most notably the type of encoding, a sequence number and a timestamp. The sequence number allows the receiver to detect packet loss and recover from packet reordering. The timestamp "reflects the sampling instant of the first byte in the RTP data packet" [51, p. 626]. It allows the receiver to handle network jitter and synchronize different RTP media streams, e.g., a video and an audio stream. RTP does however not implement any QoS measures, such as congestion control.

Even though retransmissions are specified for RTP [57], those might not arrive in time to be used in case of low-delay streams. Applications may therefore decide to handle packet losses by skipping a frame or by interpolating the media data [92, p.539]

2.4 Related Work

The performance of LTE on the ground in combination with different transport protocols has been well analyzed. One of the earlier studies was performed in a field trail in 2010 to showcase the performance capabilities of LTE. The authors performed measurements and used TCP and UDP. They found that both uplink and downlink were able to perform up to their theoretical limits (50 Mbps DL, 25 Mbps UL). The performance of TCP and UDP was on par. No handover failures were detected and the average duration of HIT was 21 ms [101]. In 2017, Akselrod et al. performed LTE measurements in a stationary setting, on a highway route, and on a city route. They reduced some of the observed performance differences to the width of the allocated frequency bands: 20 MHz cells perform better than 10 MHz cells. Furthermore, a correlation between

signal strength and throughput was shown, as high data rates are more often achieved with a higher signal strength. The comparison of TCP and UDP in transfers of four seconds revealed no throughput difference after TCP had left its slow-start phase. The authors did however not analyze other transport layer metrics such as latency or packet loss [5]. In addition to TCP and UDP, Awang Nor et al. evaluated the performance of two additional transport protocols—Stream Control Transmission Protocol (SCTP) and Datagram Congestion Control Protocol (DCCP)—over a simulated LTE connection. The results of a ns-3-based evaluation show that DCCP performs best while comparing throughput, packet loss, delay, and jitter [11].

With the roll-out of 5G, first measurement results of test deployments are published. Muzaffar et al. gathered preliminary results to compare the performance of 4G and 5G in the air with regard to throughput and handovers. The 5G-results show a significantly increased downlink throughput (up to 700 Mbps) and a similar uplink throughput compared to 4G [63]. Most production deployments of 5G do currently however operate in a non standalone mode which does not provide a performance advantage over 4G technology [70].

Based on real-world measurements, researchers have developed models that approximate different aspects of LTE. Albaladejo et al. performed stationary and mobile real-world measurements in an urban setting and modeled LTE RTT and bandwidth with a mixture of Gaussians [7]. The authors found that the model generalized to the measurements of two different operators. The distribution parameters are calculated in dependence on the signal strength. An increase in signal strength was related to lower RTTs and higher bandwidths [7]. Han et al. focused on LTE handovers and modeled handover delay and HIT. They dissected the handover process and found that HIT takes around 18 ms on average which is 21.08% of the total HO delay. As RA takes the most time during HIT, they found that the duration of HIT scales with the number of people/cell/minute. With this parameter, a normal distribution is appropriate to model the total HO delay and HIT. Stratmann et al. built an LTE emulator that was shown to accurately reproduce the results of fields measurements in terms of bandwidth, latency, and packet loss [89].

The suitability of existing network technologies for remotely controlling aerial vehicles has been discussed in multiple surveys [39, 9]. Others focus solely on the use of cellular connectivity for UAVs [32, 62]. Similarly, Zeng et al. argue why they see cellular connectivity as the most promising communication platform compared to satellite links, direct ground links, and ad-hoc mesh networks [106, 105]. Nguyen et al. performed a measurement study using UAVs in a rural area with a focus on LTE's suitability as the link for CNPC traffic. They find that the Signal to Interference plus Noise Ratio (SINR) falls below a certain threshold ever more often with increasing altitude (from 4.2% at 1.5 m to 51.7% at 120 m under full network load) as more cells are detected (from 5.1 cells at 1.5 m to 16.9 cells at 120 m) and interference levels increase. Referring to these results, they highlight reliability as the main challenge for remotely controlled UAVs. Both operators and UEs can contribute to the reduction of interference levels, e.g., using

closed-loop control systems [17, 102].

While a lot of work has been invested in playing live video on mobile devices, the published research on streaming live video from a mobile device to the Internet—in particular in the air—has received much less attention. Kacianka et al. have developed a rate-adaptive video streaming algorithm that aims to prevent network congestion. They evaluated their work in UAV field tests over an IEEE 802.11 link and measured a significant RTT reduction [46]. Another IEEE 802.11-based approach that uses a custom adaptive streaming system was tested and described in [64]. The depth of the literature is more shallow with regard to streaming live video over cellular links. A research group has investigated how LTE-connected drones can be utilized for video surveillance and used a network simulator to assess their work [65, 77]. We could not find further related work that covers real-time video transmission over cellular networks in the air.

2.5 Summary

We have presented the concepts of the topics that are relevant to this thesis. First, we introduced the architecture of LTE, described the structure of LTE's data link layer and the processes that increase its reliability, and explained LTE's break-before-make handover mechanism. Second, we explained the communication requirements and challenges of cellular-connected UAVs, before we introduced the relevant transport protocols. The related work section provides an overview about the relevant literature, e.g., about LTE's performance on the ground and in the air, and about the approaches to live video streaming from an UE.

3 Measurement Setup

The goal of this thesis is to evaluate the suitability of different transport protocols for the remote piloting of aerial vehicles over LTE. Important criteria are reliability as well as performance in terms of latency and throughput. First, we selected the most promising transport protocols for our use case and developed measurement strategies to compare their performance. These processes are described in Section 3.1 before we dive into the final hardware and software configurations in Section 3.2. Finally, Section 3.3 and Table 3.2 summarize the executed tests in terms of location, hardware, and software configuration.

3.1 Design

To characterize the performance of the selected transport protocols in the envisioned use case, we sought an approach to perform LTE measurements in the air. We designed a test strategy that closely resembles the conditions under which, e.g., UAVs and eVTOLs, operate: Using an UAV, we carry measurement equipment to different altitudes during which it performs tests over a public LTE network. Within the limits of the regulatory constraints, we can vary altitude and choose the location freely. The design of our drone tests is described in the following sections.

3.1.1 Test Location

Future aerial vehicles will take on different tasks that will be performed in different environments: Inspections of wind turbines will have to be carried out in rural areas while eVTOLs will taxi people in urban locations. It was therefore important to us, to conduct tests in both types of environments to notice similarities and differences that might necessitate specialized solutions. Urban and rural locations differ in terms of properties that influence the radio frequency channel and in terms of the available LTE infrastructure. In a rural area, one's view might extend to a few kilometers while it is often limited to a couple of meters by a building in a city. Similarly, the height of objects (e.g., trees in the country-side and buildings in the city) can differ significantly in addition to the density and configuration of cellular towers due to different population densities. Cellular providers have to allocate more resources to crowded places to deliver a consistent service level. On the flip-side, rural areas can have spotty coverage as large areas are only served by a few cells.

We conducted UAV flights on the campus of the Technical University of Munich that is close to the city's center. The university buildings enclose a courtyard in which we



(a) The flight corridor.



(b) A cell tower is close to the campus.

Figure 3.1: The urban test location close to Munich's city center. The courtyard is enclosed by 40–60 m tall buildings.



(a)



(b)

Figure 3.2: The rural flight location, approximately 36 km outside of Munich. Only few houses and natural obstacles such as trees shape the landscape.

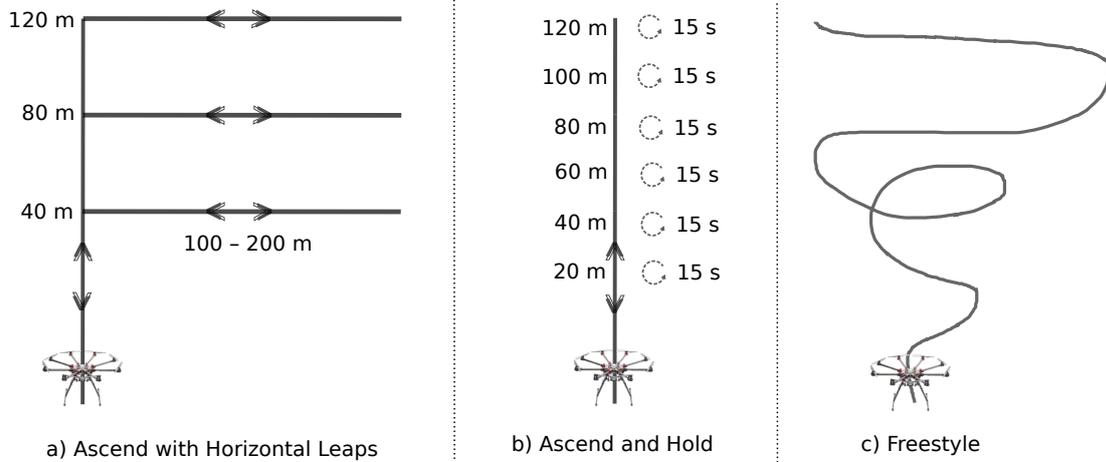


Figure 3.3: The flight trajectories that were conducted.

were allowed to perform flights in an area of approximately 200×25 m. The area is bounded to the east by the no-flight zone of a helipad that is operated by a hospital. The surrounding university buildings have an approximate height of 40–60 m. In the surrounding area, no buildings are significantly higher. As shown in Figure 3.1(b), a tower with a large number of antennas exists close to our urban flight area. Only from certain angles and below approximately 20 m, no LoS exists from the drone to this tower.

As a companion to our urban location, we managed to find a location in a rural area, where flying was legal and where the property owner granted us takeoff and landing permissions. The area is close to Türkenfeld, a municipality with 3710 inhabitants [15] that is approximately 36 km from the urban test location in Munich’s city center (distance measured directly). The surroundings and conditions on flight day are shown in Figure 3.2. The flight area is enclosed by a line of trees at its northern border but open towards all other directions. Only a few houses are in the vicinity of the property.

3.1.2 Flight Trajectories

To compare the impact of the drone’s movement, we developed a standardized flight trajectory that was to be repeated in different flights. Figure 3.3(a) visualizes the plan: The drone ascends step-wise to 40, 80, and 120 m. At each step, a loop is performed, where the drone flies parallel to the ground to the other end of the test area and returns on the same path. Figure 4.1(a) traces the path of a flight that was executed according to the standardized trajectory. Real-world flights might also follow a trajectory that separates vertical and horizontal movement to ease coordination with other aerial vehicles. To gather sufficient data that can be compared to each other, the majority of flights should follow the standardized trajectory. In addition, we planned “altitude tests” where the drone ascends in increments of 20 m and holds that position for a short while (see Figure 3.3(b)). Those tests should give us additional data about the influence of

altitude on the transport protocol’s performance.

Finally, we planned to conduct a series of flights that do not follow a pre-set path and include frequent accelerations and decelerations. During those flights, the drone should repeatedly change its orientation (rotate around its center axis) and ascend and descend in altitude. These “*freestyle*” flights (see Figures 3.3(c) and 4.1(b)) are supposed to model applications such as emergency rescues where flight paths are not pre-planned. Although we were given the opportunity to fly higher than 120 m in the urban area during the first set of flights, we later had to limit our flight altitude to 120 m for the remaining flights.

3.1.3 Selection of Transport Protocols

The overhead of tests and the need to collect a large enough amount of data to derive meaningful results, forced us to limit the number of different transport protocols to include in this study. We settled on including TCP and UDP, which are the two most popular transport protocols on the Internet by a long way, and QUIC as a “next-generation” technology in our study. The design and capabilities of these protocols are described in Section 2.3.

TCP has proven itself in various application areas: It serves in large-scale deployments on the World Wide Web while also connecting low-powered Internet-of-Things devices [35]. Since TCP alleviates the problems that are part of every network (e.g., packet losses), it eases the development of new applications. We therefore think that it is desirable to use TCP for the remote piloting of aerial vehicles and likely that it will adapt to this use case.

UDP is only a thin layer on top of IP and allows the application to precisely control the sending of packets. It is often used for real-time applications such as video and audio delivery or online gaming. We see a similarity with our use case where control orders and information needs to be exchanged with a low latency between operators and vehicles. UDP was included in our measurements as the second protocol for this reason.

QUIC aims to improve performance in use cases that are currently served by TCP. As it was only recently standardized [44], the development of implementations is still on-going and its performance characteristics are not fully understood [104]. We see potential in features such as *streams* that can be used to avoid TCP’s HoL-blocking issue and improve performance for real-time applications [36][Chapter 2]. To explore its potential, QUIC was included in our analysis.

3.1.4 Performance Tests

We were looking for a testing methodology that sheds light on all relevant performance aspects of the protocols under test. To meet this end, we designed several tests.

The first type of test should saturate the link with traffic of each tested transport protocols to derive insights about the achieved throughput, the emerged latency, and

Table 3.1: The test configurations of the test flights. Each test was always executed over both providers. “RTP-TCP” and “RTP-UDP” denote video transfers while “TCP”, “UDP”, and “QUIC” mark tests that transmit generic data. If “(N Mbps)” is noted in brackets after a protocol name, the data rate was limited to N Mbps; if not, the rate was not limited. The downlink UDP rate of 100 Mbps is higher than the maximum rate of the LTE link in our tests.

Test	Pi 1 (P2) & Pi 3 (P1)		Pi 2 (P2) & Pi 4 (P1)	
	Uplink	Downlink	Uplink	Downlink
1	RTP-TCP	TCP (5 Mbps)	RTP-UDP	UDP (5 Mbps)
2	RTP-TCP	UDP (5 Mbps)	RTP-UDP	TCP (5 Mbps)
3	TCP		UDP (100 Mbps)	UDP (100 Mbps)
4	TCP		QUIC	
5	RTP-TCP		RTP-UDP	
6		TCP		UDP (100 Mbps)
7	TCP (Pi 1)	QUIC (Pi 2)	UDP (100 Mbps)	UDP (100 Mbps)
8	QUIC	UDP (5 Mbps)	TCP	UDP (5 Mbps)

the probability of packet losses. Data should be both served *uplink* (to the server) and *downlink* (to the drone) to gather information about the influence of the direction. The second type of test should generate traffic that is close to the communication patterns of remotely controlled aerial vehicles. One type of traffic is control information which can be approximated by sending a constant low-bitrate stream from the server to the drone. Measurements of real-world drones have shown that their control traffic rate is below 200 Kbps [12]. We have chosen to model this traffic using a constant 5 Mbps data stream for two reasons: First, we expect that the amount of control data will rise for more demanding applications such as passenger transport. Second, any performance differences of the tested transport protocols are likely exaggerated if they are tasked to uphold a higher data rate.

Another type of traffic is video that is generated on-the-fly by the drone and sent to the remote operator. In our case, video should be served by the drone such that a remote server can render it and record QoS metrics such as frames per second and playback latency. The last type of traffic is telemetry, that is sent by the drone to the server. As this is also a constant low-bitrate stream, its effect is minimal and can be considered part of the video traffic that we generate.

3.1.5 Test Execution

To make the test results from repeated flights comparable, we aim to control as many external factors as possible. However, some are not under our control: Especially the state of the LTE network is not observable by us and its performance may change

between subsequent flights. Other variables are the weather and the data center’s load. We therefore decided to conduct multiple concurrent tests during each flight. This way, most of the external influences will affect the tests equally which will make the comparison between them fairer. To be able to identify carrier-related behavior, we scheduled each test during each flight twice, each over a different carrier network. We used the networks of two large German mobile network operators, *P1* and *P2* (provider one and provider two), to conduct the measurements. The operators advertise downlink rates of up to 300 Mbps (*P1*)/500 Mbps (*P2*) and uplink rates of up to 50 Mbps (*P1* and *P2*). Lastly, having settled on the drone’s trajectories, the protocols under test, and the type of tests those should undergo, we setup a detailed flight and test plan. Table 3.1 shows the different types of flights that we performed. Each flight can either follow the fixed or free trajectory and can be performed and repeated in an arbitrary order.

Test 5 gathers data to compare video transmission performance if it is transported over TCP or UDP. During tests 1 and 2, control traffic is in addition concurrently sent on the downlink. The two tests differ in their video and control traffic protocol combination: During test 1, the protocols are paired “homogeneously” (e.g., RTP-TCP with TCP), while the pairing is “heterogeneous” (e.g., RTP-TCP with UDP) during test 2. These three tests will help us analyze the protocols’ performance for video streaming and in addition measure the impact of a concurrent data stream. Tests 3 and 4 measure the protocols’ performance on the uplink channel. To collect more data, we tested UDP downlink in addition to UDP uplink during test 3. Preliminary trials showed that the streams do not affect each others performance. Downlink behavior was measured by test 6 (TCP and UDP) and test 7 (QUIC). Test 7 was the only test, where the test configurations were not mirrored between the operators.

Test 8 was an attempt at simulating video streaming over QUIC as actually transmitting video over QUIC was not doable (see Subsection 3.2.2). In contrast to video streams which exhibit a “bursty” traffic pattern, test 8 produced constant-rate traffic. A TCP uplink transfer was run in parallel to have a comparison point for QUIC. All QUIC tests were initially planned as uplink tests. In the post analysis, we discovered that the QUIC connections actually transferred data downlink.

3.2 Realization

The planned flights and tests were executed using a consumer drone, an array of single-board computers, and a server, as depicted in Figure 3.4. In the following, we describe the hardware that we used, along with the software configuration.

3.2.1 Hardware Configuration

Our measurements were conducted by flying a drone carrying a payload of multiple Raspberry Pi computers and batteries as shown in Figure 3.5. A DJI Spreading Wings S1000+ was used to conduct a first test flight in an urban setting. The drone is specified

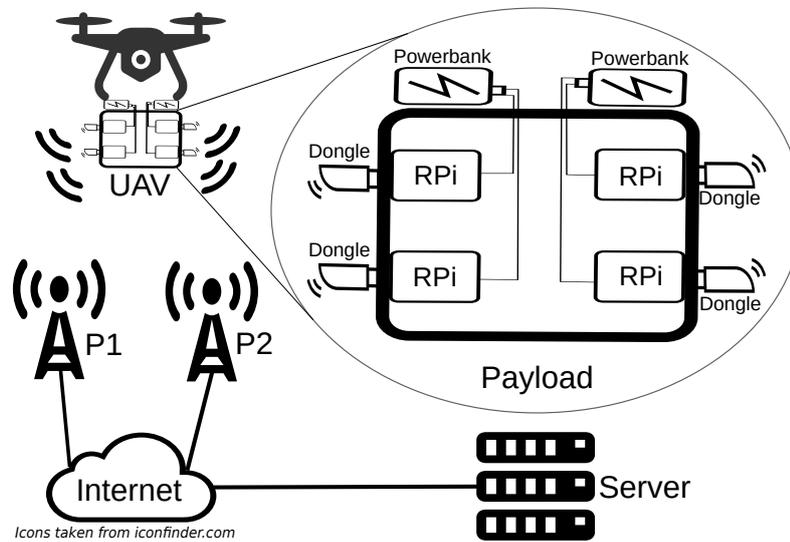


Figure 3.4: A schematic depiction of our measurement setup.



(a) The DJI Matrice 600 PRO drone along with the payload that is attached underneath.



(b) Two of the four Raspberry Pi computers are visible, the other two are similarly attached on the opposite side of the wooden box. Two batteries that are fixed on top of the case power the single-board computers.

Figure 3.5: Our measurement setup consists of a drone and a payload, that conducted the network tests.

with a total weight of 4.2 kg and a lift capacity of 11 Kg while its maximum hover time is reported at 15 minutes [23]. DJI is a Chinese company and the largest producer of consumer drones worldwide (76.1% market share in the United States between 2018 and 2020 [83]). The second and third test days were conducted with a DJI Matrice 600 PRO (or “M600”; the drone is shown in all pictures, e.g., in Figure 3.2). The drone and batteries weighed 10 kg which leaves 5.5 kg for payload. Depending on the total weight, the hovering time is specified to be between 18–38 minutes. The drone is engineered to reach a velocity of 18 m/s while the maximum speed of ascending and descending is 5 m/s and 3 m/s, respectively [22]. Both drones were adequate for our use case. A failure of the S1000+ after the first test day forced the switch to the M600 for subsequent flights.

We attached Raspberry Pis equipped with USB LTE dongles to the drone. The modems are of type *TeleWell TW-LTE/4G/3G Cat 4* [94] and *D-Link DWM-222* [21]. During the first test flight, two Raspberry Pi 2 Model Bs [79] were used that each orchestrated tests over two modems. For the subsequent two flights, four Raspberry Pi 3 Model Bs [80] were used that each used one modem. The changes were motivated by performance issues that became apparent during the analysis of the first flight’s measurement results. The computers were powered through their Micro-USB port by two lithium-ion batteries.

The other party in the bandwidth and video tests was a Linux server that was hosted on Amazon Web Services (AWS) in its London region. The machine was scaled to 8 CPU cores and 16 GB of memory. Amazon advertises a network performance of “up to 10 Gbps” for the selected *c5.2xlarge* instance type [8]. The server was accessible at a known and static IP address and served as a gateway for the communication with the Raspberry Pis.

3.2.2 Software Configuration

The Raspberry Pis ran the Debian-based Linux distribution that is specifically made for the product line: Raspberry Pi OS [81]. A single image was written to the SD cards of all Pis; only a few configuration option changes differentiated their final disk contents.

The system setup process was performed by custom systemd [91] services. First, the *modem_setup.service* waited until an USB LTE modem becomes available. The modems that we used, initially register themselves as a removable disk which contains information such as installation instructions. The service used the *usb_modeswitch* program to switch the operating mode of the device; the disk availability is swapped for a *wwan* network interface. The successful completion of the *modem_setup.service* triggered the execution of the *lte_connect.service*. It monitored the execution of the *wvdial* program which is a Point-to-Point Protocol (PPP) dialer responsible for dialing the LTE modem and connecting to the Internet. The service configures Linux routing, logs the assigned IP address, and restarts *wvdial* if necessary.

Once the Internet connection was established, two systemd services were started as part of the boot process that monitored and recorded information about the state of the attached modem. Service *modem_log.service* used the *AT command interface* to repeatedly request information about the signal strength (*AT_CSQ*), associated cell

(AT+CREG?), channel number, and reference signal strength (AT+QCRSRP?). The commands were separated by brief pauses of 100 ms but due to the modem's processing delay, each command was only executed about once per second. A second `modem_pcap.service` monitored the attached modem using QCSuper [71]. As both modems are based on the Qualcomm MDM9225 chipset, the tool is able to use the Qualcomm Diagnostic Monitor (QCDM) protocol to capture LTE control plane messages. This communication is stored in the *pcap* format, which can be analyzed by Wireshark [100].

To be able to reach each Raspberry Pi at a known address, we setup a Wireguard [24] VPN with the AWS server at its center. Briefly after start-up, once the LTE connection has been established, we were able to access the Raspberry Pis using SSH from a laptop. All traffic was routed through the AWS server. The cloud-hosted server was based on Linux and ran Ubuntu 20.04. We installed the applications that are necessary to conduct the planned tests as detailed in Sections 3.2.3 and 3.2.4. On the server, we disabled the offloading capabilities of the Linux kernel related to packet processing to be able to receive the packets as they were sent by the Raspberry Pis. These options include Generic Receive Offload, Generic Segmentation Offload, and the corresponding TCP-specific options. These features are aimed at improving performance but disabling them has no impact at the relatively slow rates that we produce.

During the tests, *tcpdump* [97] ran on all Raspberry Pis and on the server. The program records the network traffic on a given interface, which can be analyzed retroactively. Tools such as Wireshark [100] allow an intuitive analysis by providing a graphical user interface. Alternatively, selected fields can be exported to a more approachable data format to be analyzed by other means. We managed to stop *tcpdump* from dropping packets during recording by reducing its load and relaxing processing time boundaries. The `-s <snaplen>` option was used to only store the first 88 byte of each packet; this was enough to decode the header of most protocols. In addition, we increased the capture buffer size to 16 MB to prevent a temporary slow-down of processing to lead to dropped and non-recorded packets.

An additional source of information about TCP connections was the output of the *ss* CLI tool. It accesses the kernel's information about open sockets and allowed us to observe the RTT estimation, the congestion window size, and the retransmission timeout of each connection. This information was requested and stored at a frequency of 10 Hz.

DJI drones capture information about their state and environment during flight at a high resolution. We exported the log in CSV format sampled at a frequency of 10 Hz. While resolutions up to 1000 Hz are available, the chosen export settings proved to be a good middle ground between precision and cost in terms of processing time and storage requirements. Even though no official documentation describes the various data fields, the rather descriptive column names allow the inference of their meaning. Most importantly, it gave us access to the drone's (GPS) location, altitude, and velocity for each point during the flight.

3.2.3 Constant Throughput Tests

Constant throughput tests were conducted with all three protocols. *iPerf3* [43] was used to benchmark the performance of TCP and UDP while *qperf* [19] was used to transmit QUIC traffic. The program builds upon *quicly* [78], a Golang QUIC implementation of Internet Engineering Task Force (IETF) draft 28. To take one variable out of the comparison of the TCP and QUIC, we configured both to use the same Congestion Control (CC) algorithm. As *qperf* only supports *cubic*, we had no choice in which CC algorithm to use.

The *iPerf* tests were orchestrated by a wrapper scripts that was executed on each Raspberry Pi and started the corresponding *iPerf* process on the server through an SSH connection. The port number of the *iPerf* server process was deterministically generated based on the unique Raspberry Pi ID. As the Raspberry Pis had no public IP address, connections could not be initiated by the server. While this did not obstruct the uplink tests, which were initiated by the Raspberry Pis, we had to rely on *iPerf*'s "reverse" feature for downlink tests: The client opens the connection and the server generates traffic.

In case of QUIC, we could not rely on a similar feature as the *qperf* tool did not have support for a "reverse-mode". Because its client/server semantics are different, we were only able to test QUIC's downlink performance: A *qperf* server waits for connections and starts sending data once a client connects instead of receiving it as an *iPerf* server does. Alternative approaches, such as NAT hole punching or routing the traffic through the Wireguard tunnel, were not workable. NAT hole punching did not work reliable and the cause for a significantly degraded performance, while routing traffic through Wireguard (most likely related to IP fragmentation), could not be determined.

In case of UDP, if no throughput ceiling is set, the maximum send rate is only limited by the sender's hardware capabilities. The process will try to keep the operating system and Network Interface Controller (NIC) queues filled. On a Raspberry Pi, the bottleneck will be the LTE link so that in practice at most 100 Mbps will be transmitted. On the server-side however, due to the powerful network infrastructure, 10 Gbps or more will be generated. Almost all of that will be dropped by queues in the LTE infrastructure. To limit the unnecessary load on the carrier network and to ease our analysis, we set a throughput target of 100 Mbps, whenever we ran the measurements on downlink at full capacity.

3.2.4 Video Transmission Tests

The goal of the video transmission tests was to simulate the capturing of live video on a Raspberry Pi and its transmission to a remote operator. In our test setup, the video source was a pre-recorded file and the playback happened on the cloud server. We decided against actually recording live images with a camera to improve reproducibility and to avoid further complexity.

We used the GStreamer [37] framework to develop an application that generates

and transmits RTP video traffic. The multimedia framework has an extensive plugin ecosystem for different codecs and other features and is available under the LGPL open source license. The application is written in C and consists of less than 300 lines of source code. On the sender side, the framework's C API was used to read in a video file, select the video content, and enclose the content piece-by-piece in RTP packets. Finally, the packets are sent to the listening server process. On arrival, the RTP packets were processed by GStreamer's pre-built `rtpbin` element which automatically handles RTP traffic according to best practices. Most notably, it inserts data into a `rtpjitterbuffer` element which reorders packets and smoothens their arrival rate over a configurable time interval. Finally, the video data is reassembled from consecutive RTP packets and passed to the playback sink. We used the `fpsdisplaysink` element that measures the current playback rate.

The foundation for later analysis of the video streams was formed by data from custom log files. Our GStreamer application logged the current state of the video stream at a frequency of 10 Hz on both sender and receiver. The streamer (the Raspberry Pi attached to the drone) recorded the sequence number and RTP timestamp of the last sent RTP packet, and the video's current run time. The player (the server) retrieved playback information from just before the RTP packets are reconstructed to video frames but after they have been handled and buffered by the `rtpjitterbuffer` GStreamer element. At a frequency of 10 Hz, the latest packet's RTP sequence number, timestamp, and run time is logged. In addition, the current playback rate in frames per second (FPS) is read from the playback sink and stored in the log file.

The transmitted video was H.264-encoded with variable frame rate (VFR) and constant quality settings. H.264/AVC video codec is widely supported and today's most popular video codec [18]. The encoding settings are recommended for real-world usages. The content had a frame rate of 60 frames per second and a resolution of 3840×2160 pixels ("4K") to simulate a high-quality video stream. Given a medium-high quality settings, the resulting video file had an average bitrate of 20 Mbps.

3.3 Summary

We identified the communication patterns of remotely controlled aerial vehicles—delivering control commands and video—and picked transport protocols that appear suitable to support that use case: TCP, UDP, and QUIC. To examine their performance, we setup Raspberry Pi computers that automatically connect to an LTE network and collect information about the connection. The tests that simulate the communication patterns of remotely controlled vehicles were realized using `iPerf`, `qperf`, and GStreamer.

Four instances of the LTE-connected Raspberry Pis along with batteries were integrated with a consumer drone. To be able to reason about the impact of the mobile network and the location, we scheduled tests on two different mobile networks and in two different areas. One of the locations is close to the center of a large city, the other one is situated in a rural area. We collected data with this measurement setup on three

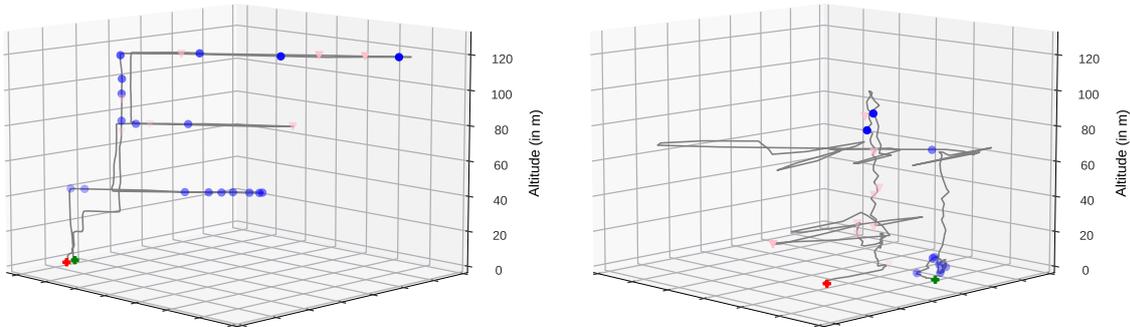
separate days which are summarized in Table 3.2 in terms of location and hardware configuration. Since fundamental changes were necessary after the first test day, we were only able to use data from the last two test days for our analysis. The results of our evaluation are presented in the following chapter.

Table 3.2: An overview of the conducted test days. “Pi N” denotes the Nth Raspberry Pi computer that was attached to the drone to carry our LTE measurements.

	Flight 1	Flight 2	Flight 3
Location ¹	Urban university campus (Arcisstraße 21, 80333 Munich, Germany). Area size: 200×25m. Surrounded by 40–50 m tall buildings.  <small>Imagery ©2021 Google, Map data ©2021</small>		Open field in a rural area. Close to Türkenfeld, Germany (Coordinates: 48.1108, 11.0724), 170×70m  <small>Imagery ©2021 GeoBasis-DE/BKG, GeoContent, Maxar Technologies, Map data ©2021 GeoBasis-DE/BKG (©2009)</small>
Maximum relative height	220 m	120 m	
UAV model	DJI S1000+	DJI Matrice 600 PRO	
Pi 1	RPi 2B, TeleWell modem with P2 SIM, D-Link modem with P1 SIM	Raspberry Pi 3 Model B, D-Link modem, P1	
Pi 2	RPi 2B, TeleWell modem with P2 SIM, D-Link modem with P1 SIM	RPi 3B, TeleWell modem, P2	
Pi 3	–	RPi 3B, D-Link modem, P1	
Pi 4	–	RPi 3B, TeleWell modem, P2	

4 Results

We setup multiple single-board computers that connected to an LTE network and attached them to a drone. Using this measurement setup, we conducted 33 flights over the course of three days in two different locations. In total, the drone spent more than two hours in the air, conducting 148 tests using the network of two different mobile operators. More than 90 million packets were received, whose log files took about 66 GB of disk space. Figures 4.1(a) and 4.1(b) show the paths of two conducted flights. One of them followed the pre-planned trajectory while the other one was performed “freestyle” and contained randomized motions.



(a) An instance of the standardized flight trajectory.

(b) A freestyle flight where the drone did not follow a standardized trajectory.

Figure 4.1: The start point of each flight paths is marked with a green cross while the end point is marked with a red cross. Pink triangles and blue circles show the occurrence of handovers on P1’s and P2’s networks, respectively.

Unfortunately, not all data that we gathered could be used for every aspect of our analysis. Especially the data recorded on our first test day revealed several errors, which necessitated technical changes to the test execution. Most of the collected results from the first test day are not included in the final analysis, to avoid accidentally introducing errors. The results are however used for the examination of handovers. Moreover, the conducted QUIC transfers could not be analyzed as thoroughly as planned. Even though we stored the cryptographic secrets that are necessary to decrypt QUIC packets, we could unfortunately not dissect its traffic. We instructed tcpdump to only store the first 88 byte of each packet with the intention of reducing the load of the single-board computers. All required information about TCP and UDP packets can be extracted from this prefix data; in case of QUIC however, this practice prevented us from decoding these packets. We were able to analyze the QUIC traffic on the UDP-layer and retrieve

information from the qperf log files. This way, we were able to compute its throughput and packet loss. The transport-layer latency during QUIC transfers could unfortunately not be analyzed.

The following sections are ordered so that lower layers are described first, before moving to upper layer protocols. Each section goes from general to more specific topics and analyses. Section 4.1 analyses the encountered handovers by studying correlations and their duration. Section 4.2 studies the amount of packet loss that each protocol experienced. Section 4.3 investigates the transport-layer latencies and looks at various influencing parameters. Section 4.4 presents and discusses the throughput results and looks at them in view of the various external influences. Finally, Section 4.5 analyses the video streaming performance of the tested protocols and the impact of coupled traffic. As an outlook, Section 4.6 describes our work on developing an emulator that can be used to reproduce the measured results.

4.1 Handovers

The LTE link layer performs a lot of work to offer a reliable platform for the upper protocol layers as described in Subsection 2.1.3. Mechanisms such as link-layer retransmissions are completely transparent to higher layers. Similarly, the process of switching from one cell to another is not announced. It entails, however, a period during which the UE is not connected to the LTE network and can neither transmit nor receive packets. In the following, we present our analysis of the extend of handovers, their effect on the higher protocol layers, and the influence of external factors.

4.1.1 Detection of Handovers

Parts of the communication between the LTE modem and the cellular infrastructure is continuously recorded (see Subsection 3.2.2). Handovers can be detected by looking for the occurrence of two messages (see Figure 2.3): The UE receives information along with the order to switch to a different cell in a RRC Connection Reconfiguration message that contains a Target Physical Cell ID. In response to this message, the UE terminates the connection to its current base station and synchronizes to the target cell. On completion, the UE sends a RRC Reconfiguration Complete message to its new cell.

These two messages do therefore denote the start and end times of handovers from the UE's point of view. Since the UE can neither receive nor send user traffic in-between those message, this duration is called "Handover Interruption Time (HIT)". The same approach to calculate HIT was taken by [101, 38].

4.1.2 Overview

Table 4.1 lists the number of handovers that the modems performed on each measurement day. In addition, the number of distinct cells which served the modems is shown. Most surprising is that the modems connected to P1 performed no handovers in the

Table 4.1: Total number of handovers split by the location at which they were recorded. The columns “M1” and “M2” contain the values for modem 1 and modem 2 of the respective provider. The column “Total” contains the sums of “M1” and “M2”. The columns “Distinct Cells” contain the number of different cells that each modem connected to at least once.

Location	Handovers P2			Handovers P1			Distinct Cells P2		Distinct Cells P1	
	M1	M2	Total	M1	M2	Total	M1	M2	M1	M2
Urban	221	210	431	92	120	212	15	12	17	17
Rural	489	500	989	0	0	0	10	17	1	1
Total			1420			120				

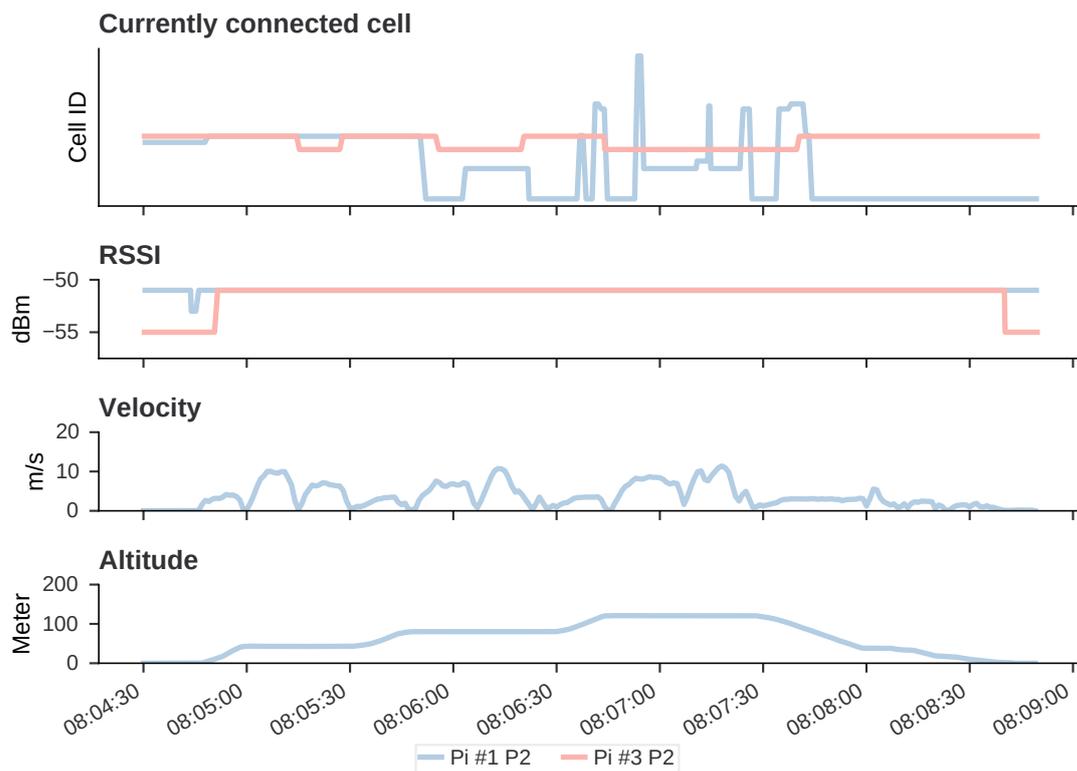


Figure 4.2: The handovers of the two modems that are both connected to P2’s network and the conditions during a flight in the urban area. The x-axis shows the wall clock time. The first row visualizes which cell each modem is attached to. A value change indicates an handover at that point in time.

rural test area. The reported Received Signal Strength Indication (RSSI) was at a good average level of -62 dBm but occasionally dropped to -73 dBm. The most likely reason for the omission of handovers is that no other fitting cell tower is available in proximity of the test location. This behavior is in stark contrast to the handover behavior observed at the two modems that were connected to P2's network. They switched cells very frequently (about 500 handovers per modem) and connected to 10 and 17 different cells respectively. In the urban environment, the number of handovers in P2's network is about half as big (about 215 handovers per modem) while P1's modems only performed 110 handovers on average. The number of distinct cells is in a similar range in the two test locations (rural: 10–17, urban: 12–17).

Figure 4.2 shows the conditions during one flight in the urban area. The RSSI and the cell attachment of both modems that were connected to P2's network are plotted. A cell ID change indicates an handover at that point in time. A few interesting observations can be made. First, even though both modems were connected to the same network and were physically only a few centimeters apart, they spent most of their time connected to different cells. Similarly, the number of handovers performed by each modem is very different. Second, both modems reported a very strong signal during flight and only a slightly reduced reception quality on the ground. Finally, two "ping-pong" handovers are performed in quick succession by Pi #1's modem: The first handover switches to a different cell and the second handover switches back to the initial cell.

These findings suggest that the handover management differs substantially between different locations and also between operators in the same location. Our impression from the detailed look at the individual flight is that the handover behavior also differs between modems that are very close to each other and connected to the same provider network.

4.1.3 Handover Correlation with Altitude & Velocity

To find a correlation between altitude, velocity, and the occurrence of handovers, we analyzed the flights in the urban and the rural environments that followed the standardized trajectory. The recorded handovers were grouped by the altitude or the velocity at which they occurred. In each bucket, the absolute number of handovers was divided by the time that the modem spent at that altitude or velocity. The normalized result has the unit *handovers per second*. The influence of altitude is shown in Figure 4.3.

On first sight, a strong unifying trend between the handovers in the three test days is missing. This is even more surprising in the case of the two urban tests, which were conducted in the very same location. If the large number of handovers on ground during day Urban #1 is treated as an outlier, the clearest common behavior between the three days is that handovers are more frequent above 20 m. Above that, the highest handover rates are encountered at different altitude ranges: For Urban #1, handover rates do not vary significantly between 20 and 119 m. Above that (up to 200 m), significantly less handovers were performed. For Urban #2, the handover rates are highest in the 20–39 m and 100–139 m buckets but the relative rate is consistently higher than during Urban #1.

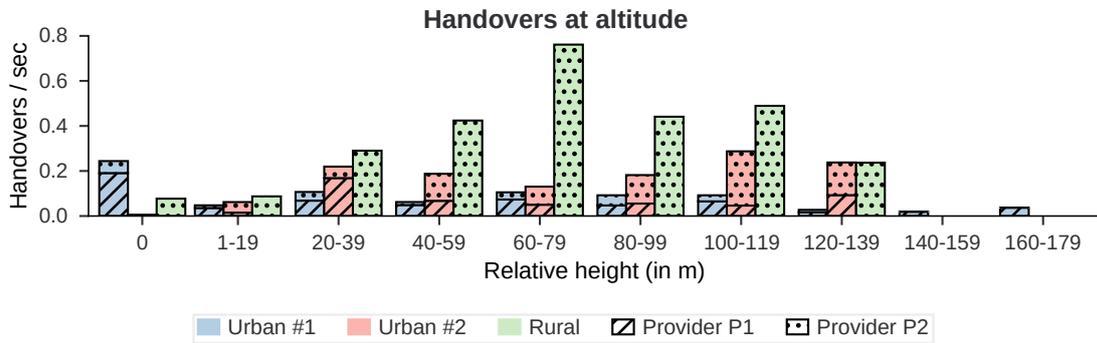


Figure 4.3: Number of handovers at different altitudes normalized by the time spent at that altitude. The three test days—two at that urban, one in the rural location—are shown. No grand unifying trend is perceptible.

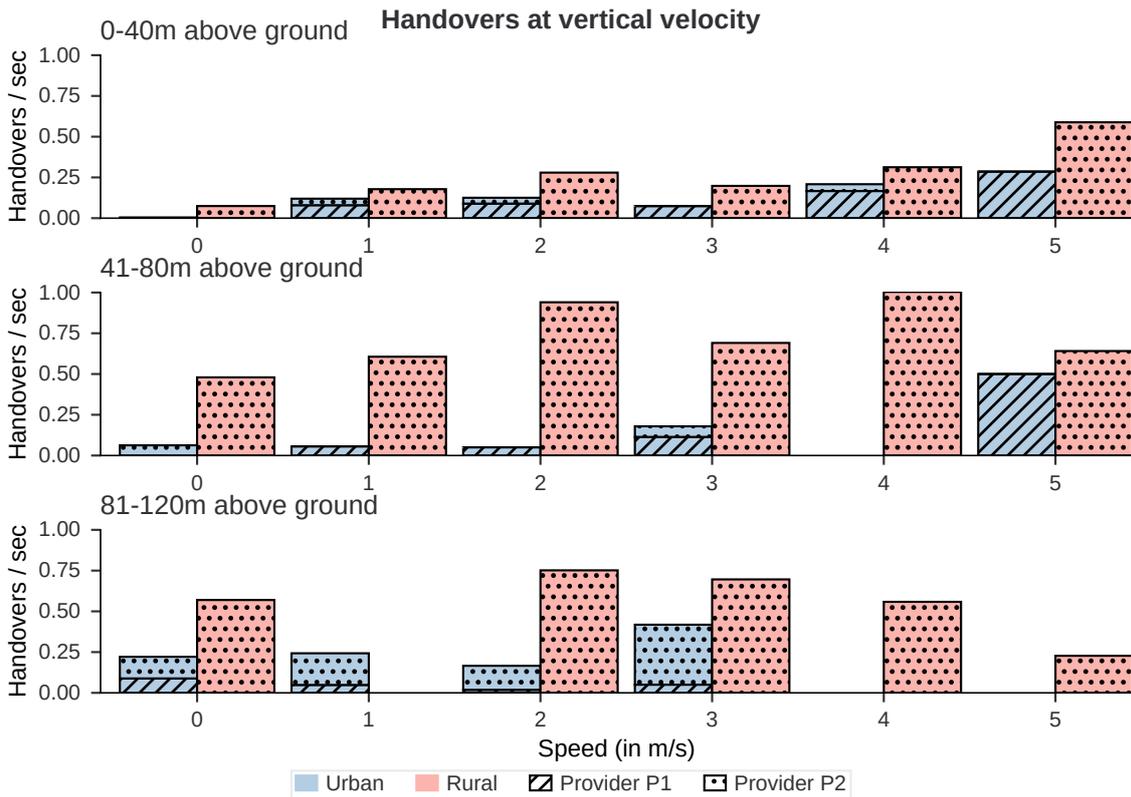


Figure 4.4: Number of handovers at different vertical velocities normalized by time. Only handovers occurring at <0.3 m/s (<1 km/h) horizontal speed are included.

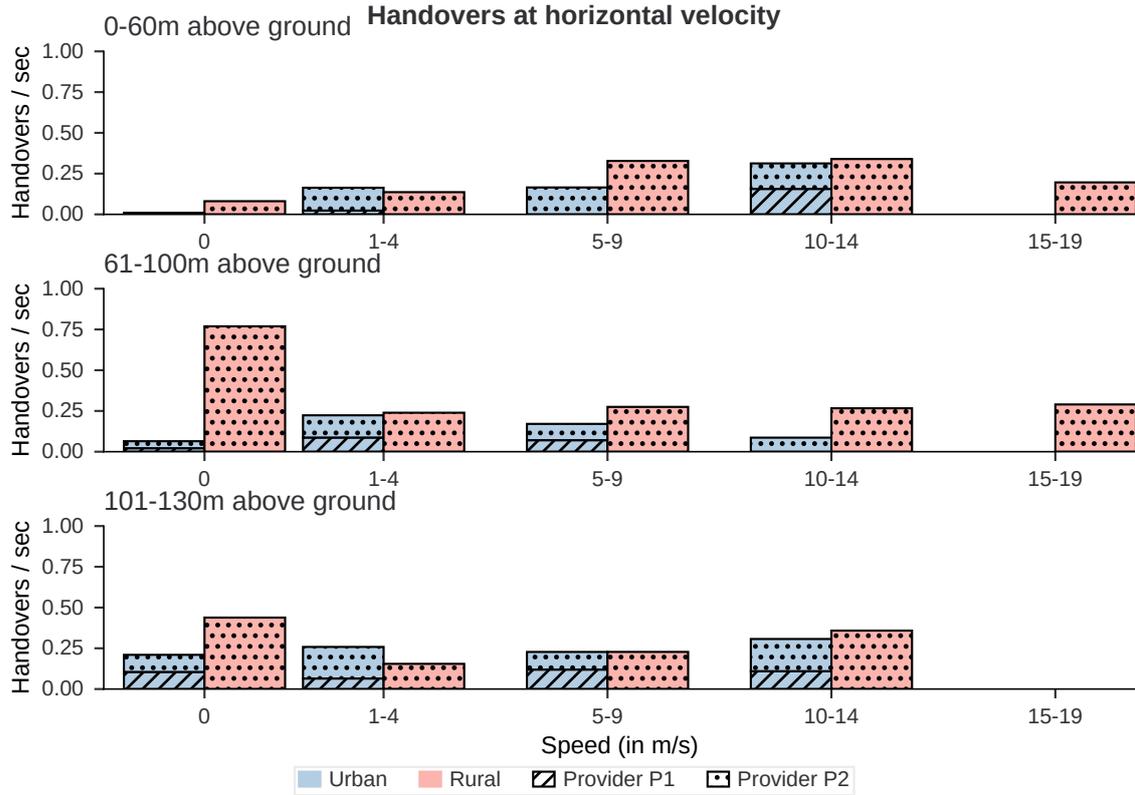


Figure 4.5: Analogue to Figure 4.4 but analyzing the handovers correlation with horizontal velocity.

The interpretation of the urban results is difficult due to the disparate behavior. A possibly influential phenomenon are radio wave reflections that could influence measurements and trigger handovers. Those would likely be most pronounced while the drone is close to the ground and flying next to the 50 m tall buildings. As the drone increases its altitude, more cells are in LoS and offer handover opportunities. The handover rate above 100 m remains high, which suggests that even at those altitudes, a LoS to further cells is obtained.

The handovers that occurred in the rural area are centered around the altitude of 60–80 m and their number remains above average up to 120 m. It is to be expected that the number of base stations with LoS increases along with altitude, which could lead to more handovers. This trend is, however, not apparent in the rural, which could be due to the interplay of ground reflections and the dominance of a single LoS signal in the rural area. At lower altitudes, the RF channel is strongly influenced by reflections from the ground. With increasing altitude, the influence of reflections decrease and LoS signals become dominant. As the density of cell towers is comparatively low in the rural area, it is likely that their distances to the drone are different. Due to path loss, the drone will likely receive distinct signal powers from each base station. Thus, as the

altitude increased above 80 m, it became easier for the modems to detect the best base station due to the dominant LoS signal and the weak influence of reflections.

Figure 4.4 splits the number of handovers based on the altitude and subsequently on the speed at which they occur. To distill the impact of vertical velocity, only handovers happening while the drone was not moving horizontally (< 1 km/h) were considered. Looking at the urban handovers, a weak trend of a rising handover rate with increasing velocity is apparent. A coherent behavior is not perceptible from the rural handovers.

Figure 4.5 repeats the analysis for horizontal velocity, i.e., movement parallel to the ground. The handover rate does only barely change between the different speeds. Comparing Figures 4.4 and 4.5, one can observe that handovers tend to occur more often during vertical rather than horizontal movement.

4.1.4 Handover Interruption Time (HIT)

Figure 4.6 shows the HIT distribution in the urban and rural test locations. The median duration is 1.9 ms and 75% of the data points are smaller than 2.41 ms but a long tail raises the mean duration to 20.01 ms (SD: 195.13 ms). The handover durations are clustered around 1.9 ms, 3.8 ms, and 200 ms as shown in Figures 4.6(b) and 4.6(c). Only 4.6% of urban handovers and 3.5% of rural handovers take longer than 10 ms. Five handovers in each location, with interruption times longer than 250 ms, are not shown; these lasted up to 4,480 ms.

The “black-box” characteristics of the LTE infrastructure and in particular its handover management prevent us from drawing definite conclusion from these results. From our viewpoint, we are not able to answer why the distribution of HIT is as it is. Comparative data was gathered by Han et al. in urban areas such as public transport stations [38]. In their measurements, HIT varies between 17.1 and 19.45 ms on average, which matches our results. They did however not encounter any outliers as indicated by a low standard deviation of 0.516–1.202 ms. This led them to conclude that a normal distribution fits their HIT measurements best. In our case, a multimodal distribution or mixture model [85] is likely needed to describe the data.

4.1.5 Summary

Our analysis of handovers is limited to describing under which conditions they are more likely to occur. Sometimes, we can take educated guesses to justify our findings, but the lack of insight into the LTE infrastructure sets boundaries to the depth of our analysis. For this reason, the different occurrence patterns of handovers in different locations, on different operator networks, and even across modems that are attached to the same mobile operator network in the same location, can hardly be explained. We see that not only signal quality changes trigger handovers, but also decisions made by the network operator to, e.g., balance the load between neighbor base stations.

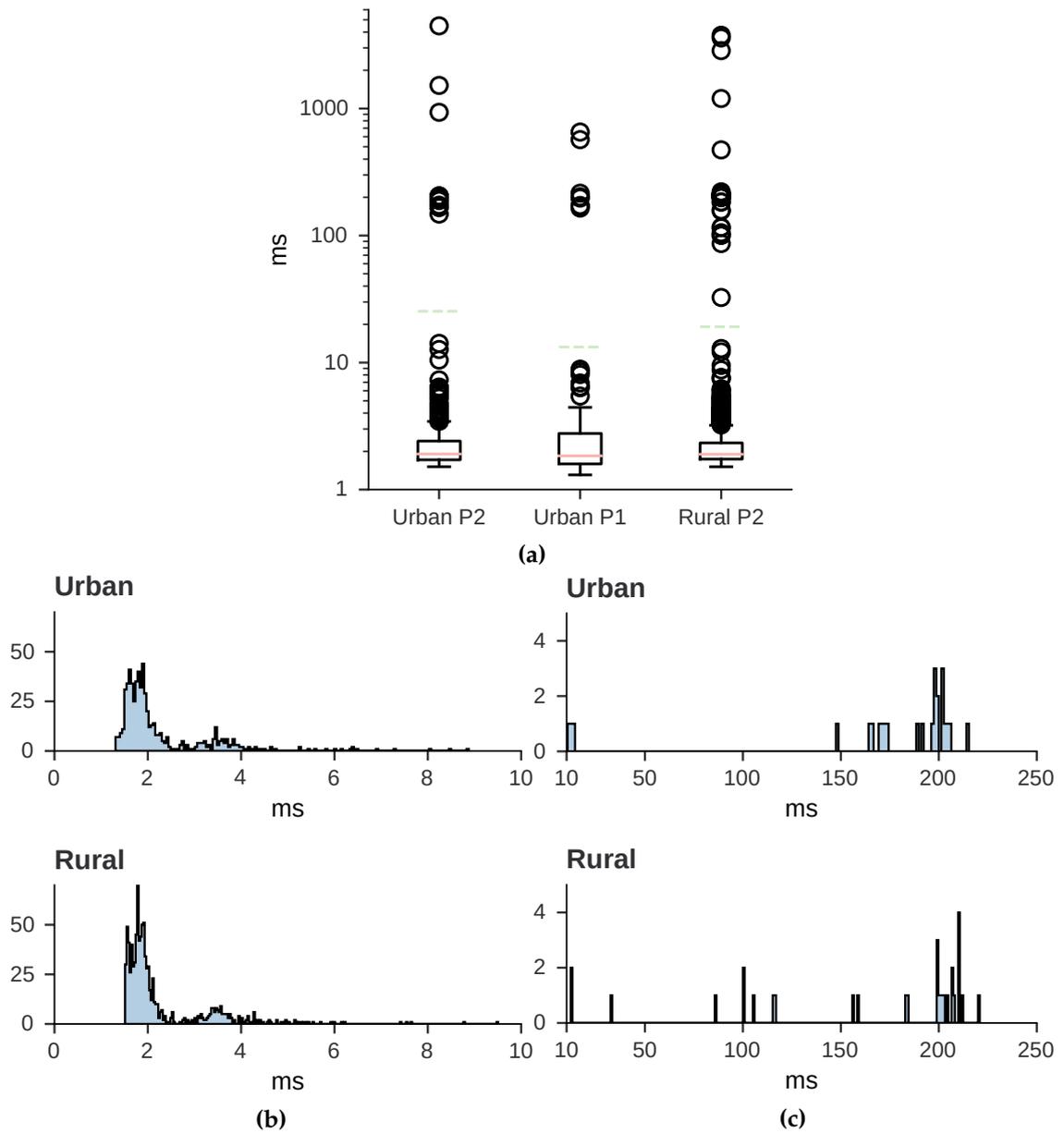


Figure 4.6: The distribution of the Handover Interruption Time (HIT) from all handovers that occurred in the urban and rural test locations.

The HIT is very brief on average which means that the performance of higher level protocols should not be impacted much by handovers in the normal case. However, handovers that include a HIT around 200 ms and the occasional outliers that last up to a couple of seconds, will definitely be noticeable. Especially those edge-cases are relevant to real-time applications that require high availability and reliability such as the remote controlling of aerial vehicles.

4.2 Packet Loss

Network packets can be lost at different points while traveling from the generating application at the sender to the consuming application at the receiver. Throughout this section, we will analyze the losses of packets on the transport layer. In other words, we only mark TCP and UDP packets as lost that have left the sender, but did not arrive at the receiver.

Reasons for packet loss can be software bugs, buffer overflows, transmission bit errors, etc. Compared to wired transmission methods such as Ethernet, wireless transmission using radio waves has a high loss probability due to fluctuating RF channel conditions. As described in Subsection 2.1.3, LTE has multiple error control mechanisms to increase link reliability in face of the challenges that the medium poses. Packets can however also be dropped on the sender, receiver, and along the path, if packet queues fill up and space needs to be freed. This is most prevalent at the sender if the application generates a higher rate than the network interface can transmit. In the case of LTE, once the UE has managed to transmit a packet to the base station and the LTE backbone network, the packet is subjected to the same loss probability as traffic arriving from other sources. On the return path, queues are expected to be overflowed if the rate is higher than what can be delivered to the UE, resulting in bufferbloat [95].

4.2.1 Computation of Packet Losses

In case of TCP, the packet loss was computed by grouping packets based on their sequence numbers across the sender's and the receiver's pcap. By comparing the size of these groups, the number of lost packets can be calculated: for group size $N > 1$, we can infer that $N - 2$ packets have been lost, which triggered TCP retransmissions. If the group has a size of 2, the TCP segment has been sent and received without problems. If the group size equals 3, a TCP segment with that group's sequence number must have been sent twice but only has been received once.

Note that all TCP connections transferred less than 4 GB so that the sequence numbers did not wrap around and did not cross the connection's initial sequence number. In other words, it never happened that the sequence number of a connection (i) started at N , (ii) wrapped around from $2^{32}-1$ to 0, and (iii) grew beyond $N - 1$ again. If this would have happened, the described computation of packet losses could have not been applied.

In case of UDP, we utilized the strictly increasing iPerf ID that is included in each iPerf UDP packet. For each iPerf packet in the sender's pcap, we searched for the matching packet in the receiver's pcap by matching packets on their iPerf IDs. All sent packets that have no partner in the receiver's pcap are packets that have been lost during transfer.

4.2.2 Number of Packet Losses

To evaluate the loss probability of the LTE connection, we computed the packet losses in all TCP transfers, the uplink UDP tests, and the downlink QUIC tests. Downlink UDP tests were not included as its data rate aimed at a higher value than the receiver's LTE link could ever support. A high fraction of sent packets are therefore consistently lost.

Tables 4.2, 4.3, 4.4, 4.5 and 4.6 show packet losses split by transfer direction and type. They list the absolute number of packet losses and the fraction of lost packets compared to the total number of sent packets. We find that the PER of the TCP and UDP transfers (both constant throughput and RTP tests) is between 0.0143–0.0617%, which is considerably lower than QUIC's downlink PER of 0.8498%. Small differences are however also apparent within the PER of TCP and UDP. They hint at a higher loss probability of the downlink compared to the uplink (TCP iPerf DL: 0.0592%, UL: 0.0204%) and at an impact of the traffic shape: The loss probability of the bursty RTP uplink traffic (0.0617%) is almost three times as large as the loss probability of the constant iPerf uplink traffic.

The QUIC downlink transfers had the greatest number packet losses both on an relative and absolute basis (0.8498% / 34,4000). The loss probability differences between carriers does not seem to be consistent between the different categories.

4.2.3 Packet Losses over Time

To gauge the impact of packet losses, their distribution over time needs to be investigated: Are packet losses spread out evenly over the course of a connection or are they clustered around a few occurrences? We analyzed their distribution by dividing each connection into one second buckets and computing the share of lost packets per buckets. Figure 4.7 shows the loss percentages for all seconds with non-zero loss.

The median losses per second range between 0.1% and 11% but higher loss percentages are common. Especially the downlink transfers contain a lot of outliers, where a large fraction of all the packets that were sent in one second did not arrive at the receiver. We can therefore conclude that packet losses most often occur in temporal proximity to other packet losses. We can only speculate about the reasons that cause this behavior: Possibly, packets are arriving in bursts at a full queue in the LTE network and need to be dropped.

Table 4.2: Loss during iPerf TCP downlink connections.

	Urban		Rural		Aggregated
	P1	P2	P1	P2	
Lost Packets	685	463	456	2,924	4,528
Total Packets	1,738,280	1,855,748	1,129,587	2,921,763	7,645,378
Loss %	0.0394%	0.0249%	0.0404%	0.1001%	0.0592%

Table 4.3: Loss during iPerf TCP uplink connections.

	Urban		Rural		Aggregated
	P1	P2	P1	P2	
Lost Packets	1,631	1,962	10	1,284	4,887
Total Packets	6,022,931	8,114,764	2,177,580	7,611,950	23,927,225
Loss %	0.0271%	0.0242%	0.0005%	0.0169%	0.0204%

Table 4.4: Loss during RTP-TCP connections (uplink).

	Urban		Rural		Aggregated
	P1	P2	P1	P2	
Lost Packets	2,303	1,298	1,749	1,471	6,821
Total Packets	3,185,178	3,564,452	1,244,738	3,060,638	11,055,006
Loss %	0.0723%	0.0364%	0.1405%	0.0481%	0.0617%

Table 4.5: Loss during iPerf UDP uplink transfers.

	Urban		Rural		Aggregated
	P1	P2	P1	P2	
Lost Packets	0	1,003	0	189	1,192
Total Packets	2,987,039	2,468,863	977,108	1,930,318	8,363,328
Loss %	0.0000%	0.0406%	0.0000%	0.0098%	0.0143%

Table 4.6: Loss during QUIC downlink transfers.

	Urban		Rural		Aggregated
	P1	P2	P1	P2	
Lost Packets	406	12,174	291	21,529	34,400
Total Packets	1,253,677	1,514,152	682,669	597,520	4,048,018
Loss %	0.0324%	0.8040%	0.0426%	3.6031%	0.8498%

4.2.4 Summary

The presented results shed light on the extent of packet loss. They show that TCP and UDP have a very similar PER and that only QUIC exhibits a considerable higher loss probability. The packet loss analysis on the basis of “losses per second” reveals that rarely only a single packet is lost but that their occurrences correlate. Depending on the application and protocol, continuous packet losses can have a more severe impact than separate packet losses. Other parameter, such as altitude or velocity, do not seem to be correlated with the occurrence of losses.

We discussed the requirements on the communication link for aerial communication links in Subsection 2.2.1. The 3GPP set a PER of 0.001 as the upper limit which is exceeded by all tested link/protocol combinations [4]. This signals the need to optimize the cellular networks for aerial users with the goal of reducing packet loss. Furthermore, an investigation of the impact of packet losses on the actual communication streams of remotely controlled aerial vehicles would help gauging the relevancy of this issue for their deployment.

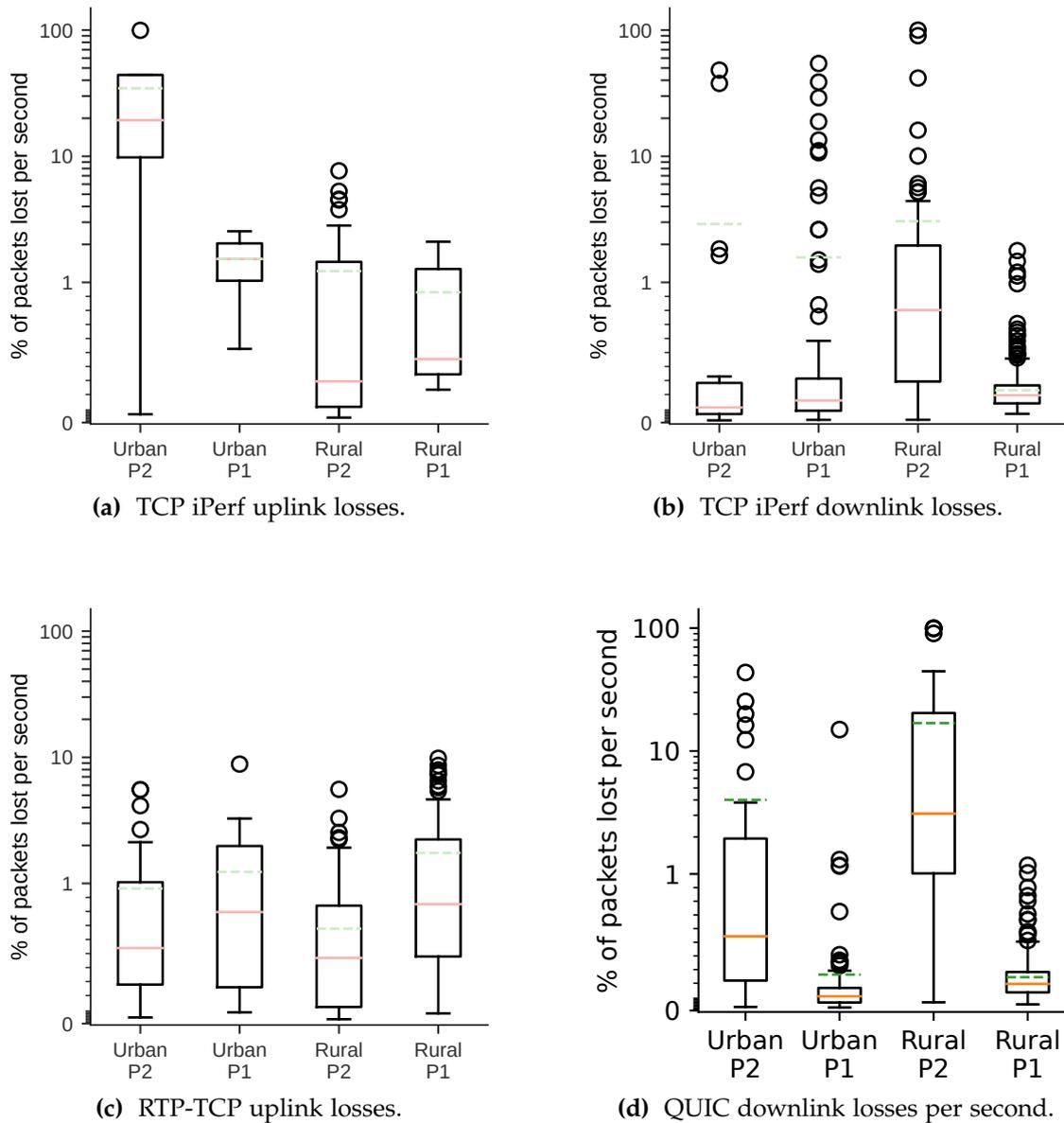


Figure 4.7: For all seconds that contained losses, the percentage of lost packets compared to the total number of packets sent in that second, is plotted. Each series' median is indicated by a orange line; the mean is indicated by a dashed green line.

4.3 Network Latency

The time that it takes a transport layer packet to travel from the sender to the receiver is analyzed throughout this section. We call this duration *One-Way Delay (OWD)* or *latency* and differentiate it from the *Round-Trip Time (RTT)*, which in addition includes the time that it takes a packet to return from the receiver the sender.

A variety of factors influence the latency: The physical length of the path from sender to receiver, link layer mechanisms such as ARQ, the processing delays at intermediate nodes, etc. Transport protocols can however also contribute to an increased latency by congesting buffers on the packet's path. We calculated the latency during iPerf transfers and analyzed the influence of test location, transport protocol, transfer direction, and throughput target. The results are listed in Table 4.7.

Table 4.7: The transport layer latencies split by test location, protocol, and transfer direction/type. Each item's mean, standard deviation, and median is given in milliseconds.

Variables	mean (ms)	SD (ms)	median (ms)
Urban			
TCP			
DL 5 Mbps	95.05	44.01	90.00
DL unlimited	57.13	98.13	50.01
UL unlimited	79.19	39.82	79.02
UDP			
DL 5 Mbps	301.98	792.77	56.00
DL unlimited	2915.43	2143.56	2427.99
UL unlimited	112.80	43.83	132.00
Rural			
TCP			
DL 5 Mbps	72.88	53.87	58.02
DL unlimited	117.13	98.38	85.01
UL unlimited	139.53	89.66	118.00
UDP			
DL 5 Mbps	423.20	745.94	126.01
DL unlimited	4064.23	2415.79	3564.01
UL unlimited	215.19	104.36	209.01

4.3.1 Calculation of Latency

The traffic generated during all tests was recorded using tcpdump on both sender and receiver. Theoretically, the actual duration that each packet was in the network can be calculated by identifying the same packets in both sender and receiver traffic log files and subtracting the arrival from the departure time. In practice, we noticed that

according to those log files, the departure of a particular packet was sometimes later than its arrival. The magnitude of this difference was not stable and much larger than what could be explained by real-time clock inconsistencies; occasionally, the “negative latency” dipped to -5 seconds. In total, 2.58%/0.55% of UDP packets that were generated by iPerf in the urban/rural test location were affected.

Since we had no alternative method at hand for computing the latency of UDP transfers, we aimed to purge all UDP tests that yield implausible latency estimates. That was achieved by discarding all UDP transfer that included a latency estimate of less than the theoretical minimal latency of 3.0688 ms¹. The test was failed by 17 UDP transfers which left 35 UDP transfer that were further analyzed. The same problem was revealed while matching TCP packets based on their sequence numbers. Fortunately, the RTT during a TCP transmission can be estimated even if only sender-side timing information is available. By finding the corresponding data packet for each incoming acknowledgment, the RTT can be estimated as the difference between the timestamps of those two packets. Wireshark exposes this information as “ack_rtt”.

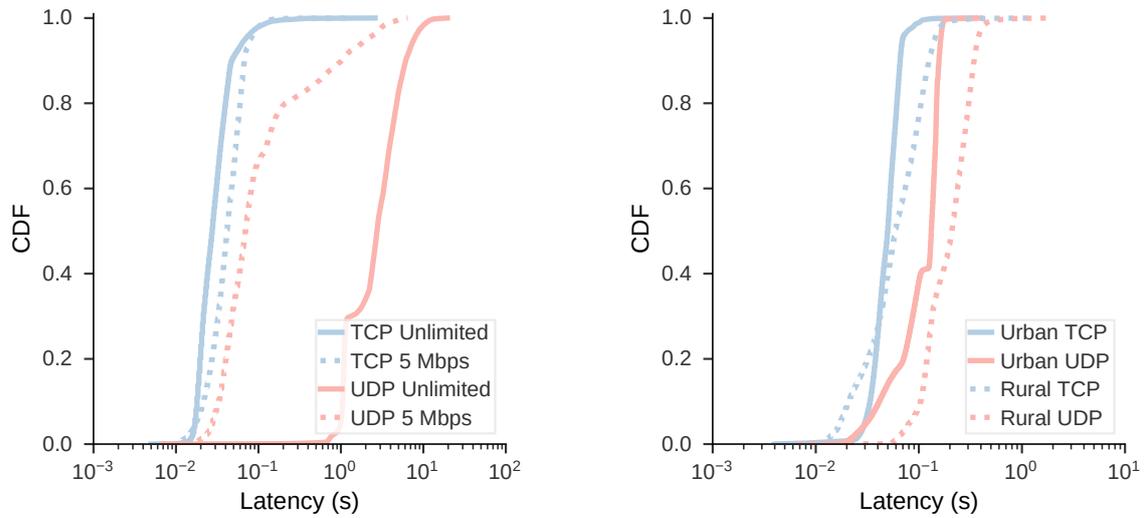
In principle, we were more convinced by the widely used RTT calculation done by Wireshark compared to the latency computed by the problematic packet log correlation. A visual comparison of the results of the log correlation and the ack_rtt calculation shows that latency trends match even if the log correlation yields negative timestamps. Moreover, in case the log correlation produces plausible values, the absolute network delay estimations match, too. Furthermore, the conversion of Wireshark’s RTT to One-Way Delay (OWD) relies on the assumption that the latencies are symmetric. Pathak et al. investigated if this assumption holds in the Internet by measuring latency and RTT in research and commercial networks [73]. They found that asymmetry (the ratio of latency and RTT) is prevalent in commercial networks and mostly ranges between 0.4 and 0.6. This means that some variation between the two latency estimation methods is to be expected. Furthermore, it means that the packet log correlation yields more reliable results if the packet log’s timestamps are correct.

Ultimately, we decided against using Wireshark’s RTT estimation for calculating TCP’s latency in favor of using the same calculation approach as for UDP. In addition to the mentioned advantages, this approach has the additional benefit that any error coming from flaws in the calculation equally affects UDP and TCP latencies. While any such error would distort the absolute values, it would most likely not warp the relative difference between the protocols.

4.3.2 Influence of Protocols and Transfer Direction

Figure 4.8(a) compares the latency during TCP and UDP iPerf transfers and the impact of the throughput target: either, iPerf was ordered to saturate the link or it was given

¹The direct distance between the test locations (Munich) and the AWS server zone (London) is approximately 920 km. Divided by the speed of light (299,792,458 m/s), we arrive at the estimated latency bound of 3.0688 ms. Due to processing delays, the actual lower bound on latency may be as high as 10 ms.



(a) Comparing the latency during TCP downlink connections that aimed to saturate the link and transfers that constantly generated 5 Mbps. The data of the two test locations was combined.

(b) Data from iPerf uplink connections was separated by test location and transport protocol. The throughput of uplink transfers was not throttled.

Figure 4.8: The impact of link saturation and test location on TCP’s and UDP’s latency is depicted.

a goal of constantly transferring 5 Mbps. We see that TCP packets travel faster from sender to receiver than UDP packets, independent of throughput. Unexpectedly, fully saturated TCP transfers are associated with a lower latency than 5 Mbps TCP transfers. This result may be a consequence of our test design: While unlimited DL TCP transfers were only tested without any other traffic on the UL, limited DL TCP transfers were only tested with RTP traffic on the UL. The utilization of the UL during those TCP DL tests may have slowed down data transfer.

The two DL UDP test types—unlimited and 5 Mbps transfers—shown in Figure 4.8(a) exhibited very different latencies. This is most likely the consequence of queues that build up at the cellular base station. The server sends data with a rate that is higher than the LTE DL capacity. The data is either dropped or queued at the base station until it can be transferred to the UE. The stopover becomes perceptible in the latency.

It is surprising that UDP performs worse than TCP in case of transfers with limited throughput (see Figure 4.8(a)). An explanation could be that the network operators apply different traffic policies to TCP and UDP traffic which could slow down the processing of UDP packets. Kakhki et al. have found that mobile network operators do indeed employ traffic shaping techniques and handle packets differently based on which transport protocols is used [47].

Figure 4.8(b) visualizes the latencies of UL iPerf transfers which were always executed with unthrottled throughput. We see that TCP exhibits a lower latency than UDP and

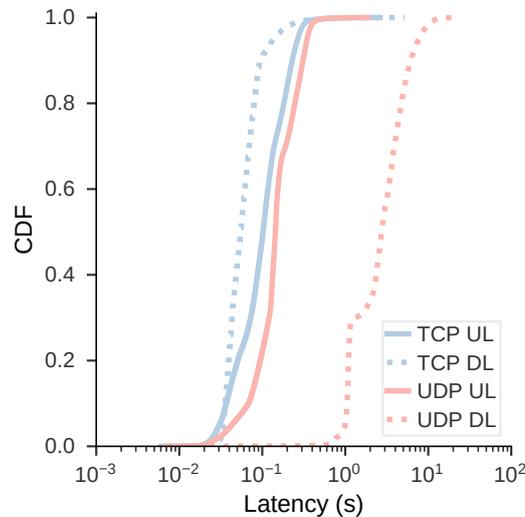


Figure 4.9: Comparing the impact of the transfer direction on the latency of TCP and UDP transfers. The data is taken from all unlimited throughput tests.

that the latencies in the rural location were generally higher than in the urban test area.

Figure 4.9 compares the latency performance of TCP and UDP on the up- and downlink channels if unlimited throughput tests are executed. As already discussed, TCP had a lower latency than UDP, both on the up- and on the downlink channel. Comparing the transfer direction, TCP's latency is noticeably higher on the UL compared to the DL: the median latency of TCP in unlimited throughput tests is 43 ms on the DL and 95 ms on the UL.

The difference between transfer directions is however far greater in case of UDP. Here, UDP's latency on the UL is much lower than on the DL. When data is sent on the UL to the server, the throughput is limited by the transmission capabilities of the UE. Packet queues build up in the sending machine's network stack, which throttle iPerf's packet generation engine. Once a packet has left the sender, it is delivered to its destination without significant delays. The packets on the DL however queue up at the base station while they wait for their transmission to the UE.

4.3.3 Influence of Signal Strength

Figure 4.10 examines the influence of the modem's reported signal strength on the latency of the transport layer. The lowest measured RSSI was -73 dBm in the analyzed tests while the highest signal strength was -49 dBm. We see a clear negative influence of signal strength on the latency. The lowest measured latency in the worst RSSI bucket (-73 to -69 dBm) is more than five times higher than in the buckets that represent a stronger signal. The influence of signal strength is however still noticeable when comparing higher RSSI values: About 70% of the latency data points in the second best bucket (-69

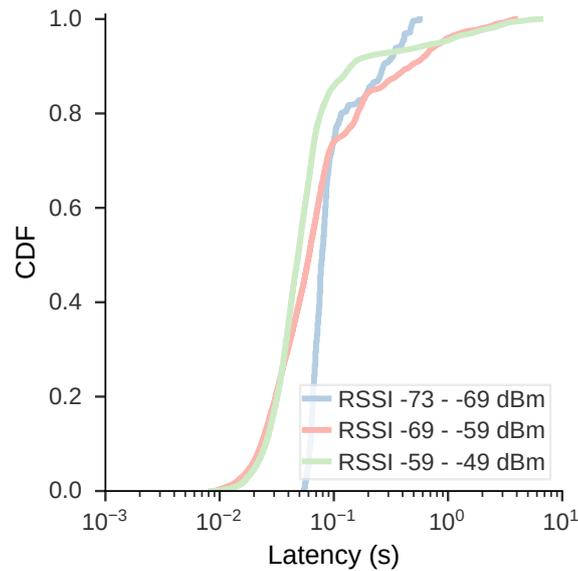


Figure 4.10: The data points were split based on the modem’s RSSI at that point in time to see the signal strength’s influence. A weaker signal seems to be correlated with a higher latency. The shown data is aggregated from all bandwidth-limited iPerf downlink connections.

to -59 dBm) are higher than in the bucket representing the best signal strength.

A lower signal strength is associated with more link-layer errors which necessitate HARQ/ARQ retransmissions (see Subsection 2.1.3). Those are opaque from the perspective of higher layers but they have an impact on the latency at the transport layer. We compared the results from the bandwidth-limited iPerf downlink tests that are shown in Figure 4.10 to the other analyzed iPerf test configurations. All of them show the same influence of RSSI.

4.3.4 Influence of Altitude

Finally, we investigated if an effect of altitude on transport-layer latency can be detected. Figure 4.11 shows the performance of TCP and UDP split into altitude ranges in each test location. The latency of TCP does not seem to diverge much depending on the altitude at which it was recorded. If it does however, lower altitudes are associated with a lower latency. The same is true for the UDP connections with the difference that the behavior is more pronounced—especially in the rural test location. These results show that TCP’s advanced mechanisms allow it to better adapt to different scenarios than UDP.

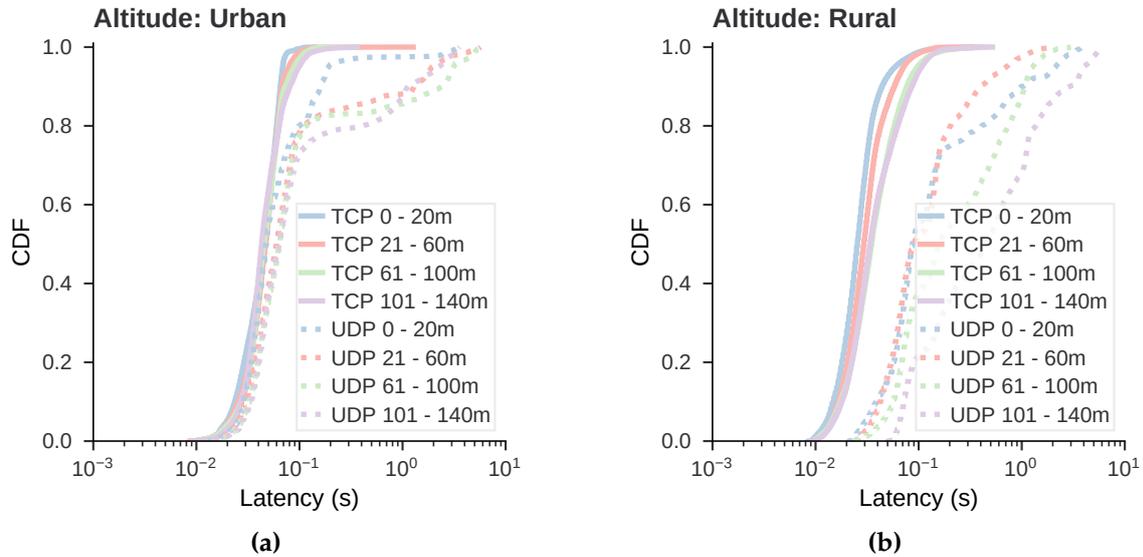


Figure 4.11: The latency of TCP and UDP iPerf connections from bandwidth-limited downlink tests is split by altitude. The difference of altitude on transport-layer latency is blurry.

4.3.5 Influence of Handovers on Latency

As discussed in Subsection 2.1.4, handovers cause the UE to be briefly disconnected from the LTE infrastructure while it is accessing its new channel. While a handover causes a link to be unavailable for only 20 ms on average in our measurements (see Subsection 4.1.4), we noticed that latency spikes sometimes coincide with handovers. Those spikes did not always occur immediately after an handover but often also up to 300 ms before it. Figure 4.12 shows two examples where this phenomenon occurs.

We observed that not all latency spikes are accompanied by an handover and similarly, not all handovers cause a spike in latency. To answer the question if the latency around handovers is actually systematically elevated, we conducted a structured analysis. We approached this problem by comparing the latency that immediately precedes and follows an handover with the latency before the handover occurred. These time frames are highlighted in Figure 4.13. The difference of the average latencies in those time intervals gives us the answer to our question: if $\text{latency}_{\text{around handover}} - \text{latency}_{\text{before handover}}$ is greater than 0, the latency close to the handover is elevated.

This analysis approach was repeated for all tests by iterating over all handover occurrences. For each handover, the difference was calculated as described and stored. We explored the parameter space and settled on taking the mean of the 50 ms immediately before and after an handover, and the mean of the second before the handover occurred. The analysis of all flights from the urban and rural environments resulted in an average difference of 75.23 ms (SD: 355.76 ms). While the mean clearly suggests that the latency around the occurrence of handovers is greater than during the preceding second, the

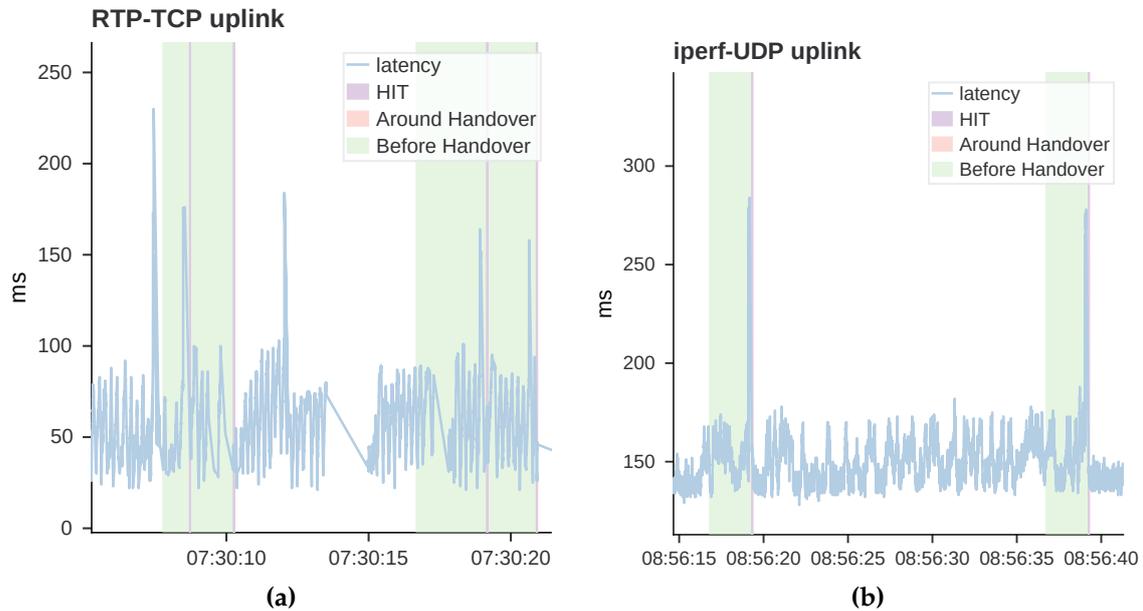


Figure 4.12: The occurrence of handovers (purple vertical lines) is sometimes accompanied by spikes in latency. The latency line is not smoothed but connects each measured latency data point (one data point per packet). The x-axis is annotated with the wall clock time.

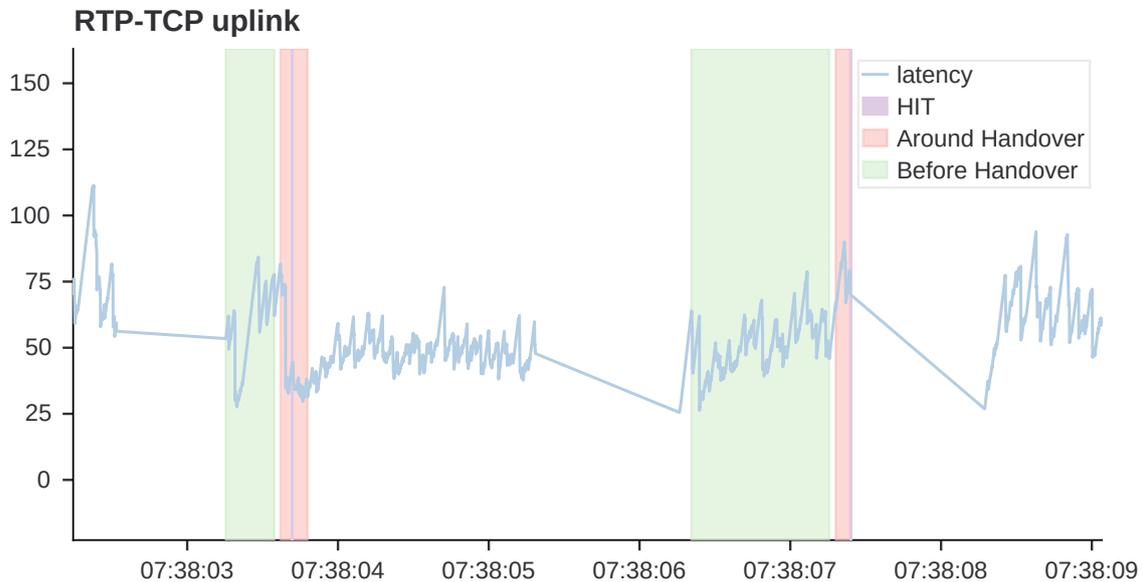


Figure 4.13: Two handover occurrences and the transport layer latency are depicted. The latency drops shortly before the handover interrupts the connection for the first time. The second time, the latency rises in the second before the UE switches cells.

large standard deviation indicates a large variety in the results.

To assess the significance of this result, we applied a statistical hypothesis test. Since the Shapiro-Wilk test [86] claims that the differences are not normally distributed, we can not conduct a paired sample t-test. A reputable alternative is the non-parametric Wilcoxon signed-rank test [99] that has no requirements on the samples' distribution. This test assesses whether the mean of the handover and the mean of the non-handover latency values differs significantly. The Wilcoxon signed-rank test rejects ($p=8e-09$) the null hypothesis that both series come from the same distribution. The test was repeated with the alternative hypothesis that the difference median is greater than 0. Again, the null hypothesis ($p=4e-09$) that the median is negative is rejected in favor of the alternative hypothesis, which states that the latency during handovers is higher than during non-handover periods.

The statistical analysis supports our impression that handovers coincide with spikes in latency. This result holds for a variety of parameter configurations as long as the time period around handovers is not larger than 250 ms.

4.3.6 Summary

The transport-layer latency is a central metric for many applications and especially important for our use case if the channel carries control information to the drone. Our measurements consistently show that TCP transfers have a lower latency than UDP transfers. This is of course most pronounced if the UDP stream saturates the link; in this case, a latency of several seconds can be expected. These findings highlight the need for flow and congestion control for latency-sensitive applications if the data rate is non-negligible. Another finding is the disparate performance in terms of latency of the up- and downlink channels: For TCP, the DL performs better than the UL. Due to the queuing behavior at the base station, the UL performs noticeably better than the UL for UDP. Furthermore, both lower signal strengths and higher altitudes correlate with higher latencies. Finally, our analysis showed that the latency can spike unpredictably due to handovers.

4.4 Throughput

The amount of data that can be transferred in a given time is the central performance criterion for a lot of applications. For example in case of bulk data transfers, the throughput directly determines the completion time. For other applications, such as video streaming, it determines the quality with which an video can be transferred to be played back in real-time.

In this section, we analyze the throughput that the tested transport protocols were able to generate. If not otherwise described, the data from the unlimited up- and downlink tests has been used to analyze the influence of various parameters on the throughput. Due to errors in experiment configuration, we only recorded QUIC uplink transfers

from flight type 4 (see Table 3.1). Including the QUIC tests from the flights of type 7 would introduce a bias as those tests were only executed on carrier P1.

4.4.1 Comparison of Transport Protocols

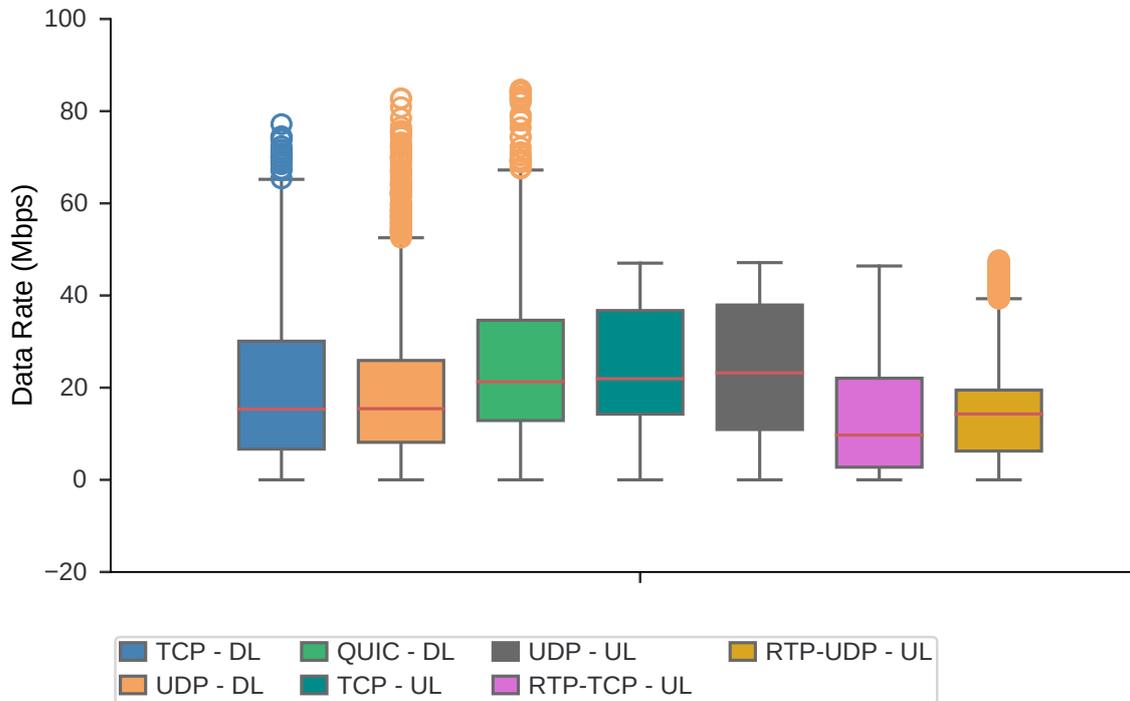


Figure 4.14: The throughput of all protocols is compared while the data is split by the transmission direction (up- or downlink). QUIC’s performance was only tested on the uplink. RTP-TCP and RTP-UDP show the throughput during video transmissions where the source material’s data rate was 20 Mbps.

Figure 4.14 compares the up- and downlink performance of all tested protocols. The data is taken from all flights where the maximum possible throughput was tested or video was transmitted. The analysis shows that, on the DL, UDP and TCP performed almost equally well, while QUIC shows significantly better performance. However, as QUIC’s throughput results only come from type 4 flights (see Table 3.1), we believe that its performance is overestimated in comparison to the other analyzed protocols that are shown in Figure 4.14. Comparing TCP-UL’s overall throughput (median: 22 Mbps) and its throughput in all flights of type 4 (median: 35 Mbps), we notice that throughput was a lot higher during flights of type 4 (see Figure 4.15). Apparently, other flight’s throughput was much worse and pulled down the overall median value.

We now compare the performance of TCP to UDP on the UL. The median throughput of the two protocols is almost equal with UDP having a small edge over TCP. Overall, the UL performs noticeably better than the DL: TCP’s and UDP’s throughput is about

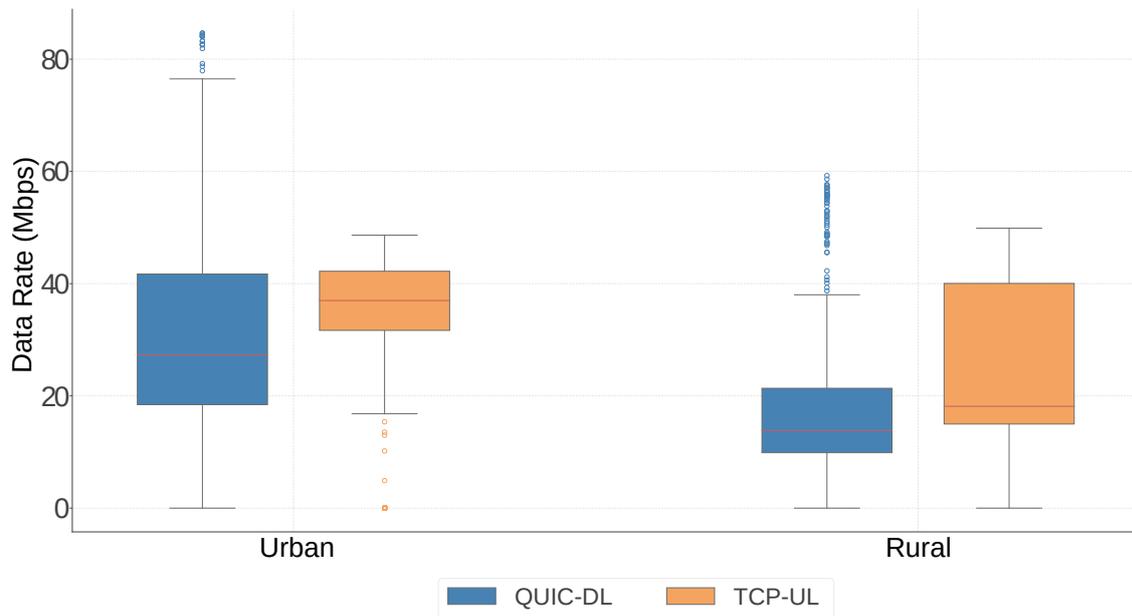


Figure 4.15: The throughput of QUIC and TCP during all flights of type 4 (see Table 3.1) is shown to highlight that the throughput during these tests is much higher than during other tests.

6 Mbps higher. This is surprising as it contradicts the operator’s declarations: They advertise DL rates of up to 300 Mbps (P1) /500 Mbps (P2) and UL rates of up to 50 Mbps (P1 and P2).

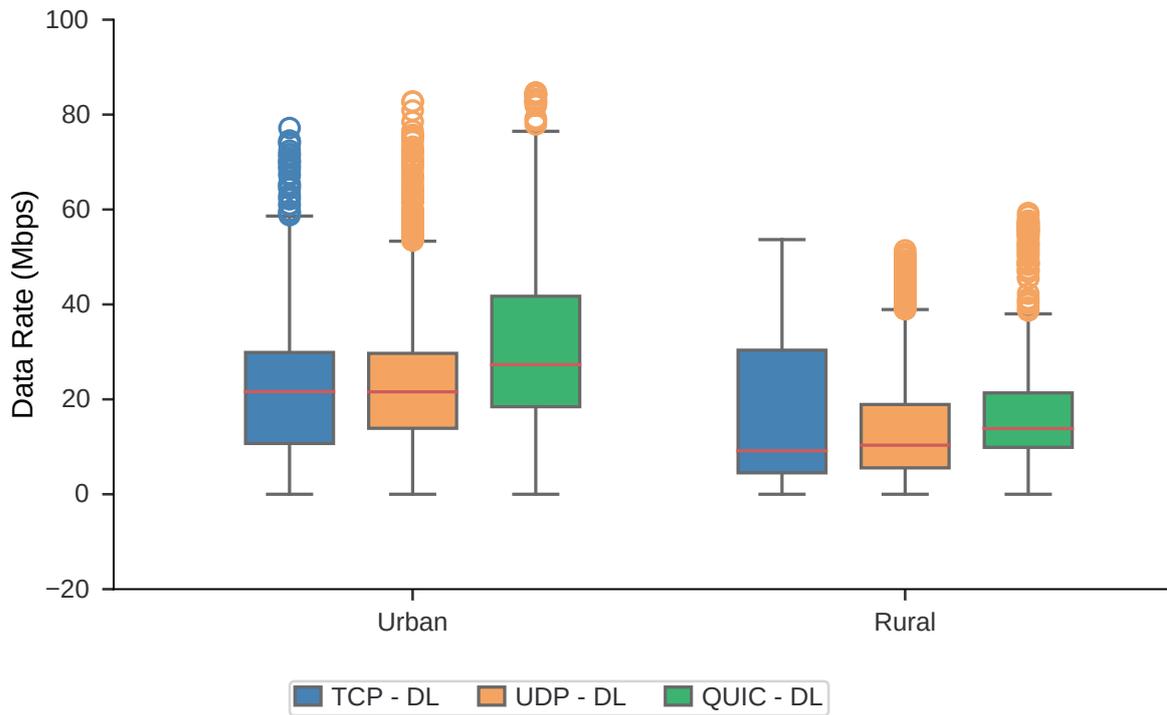
The video streams were transported by RTP that itself builds upon TCP and UDP, which handle the actual data transfer. As the transferred video had an average data rate of 20 Mbps, a similar TCP and UDP throughput was expected. In practice, the median throughput of TCP was close to 10 Mbps while UDP maintained close to 15 Mbps. The lower and upper quantiles show that RTP-TCP’s throughput varied more than RTP-UDP’s.

4.4.2 Influence of the Test Location

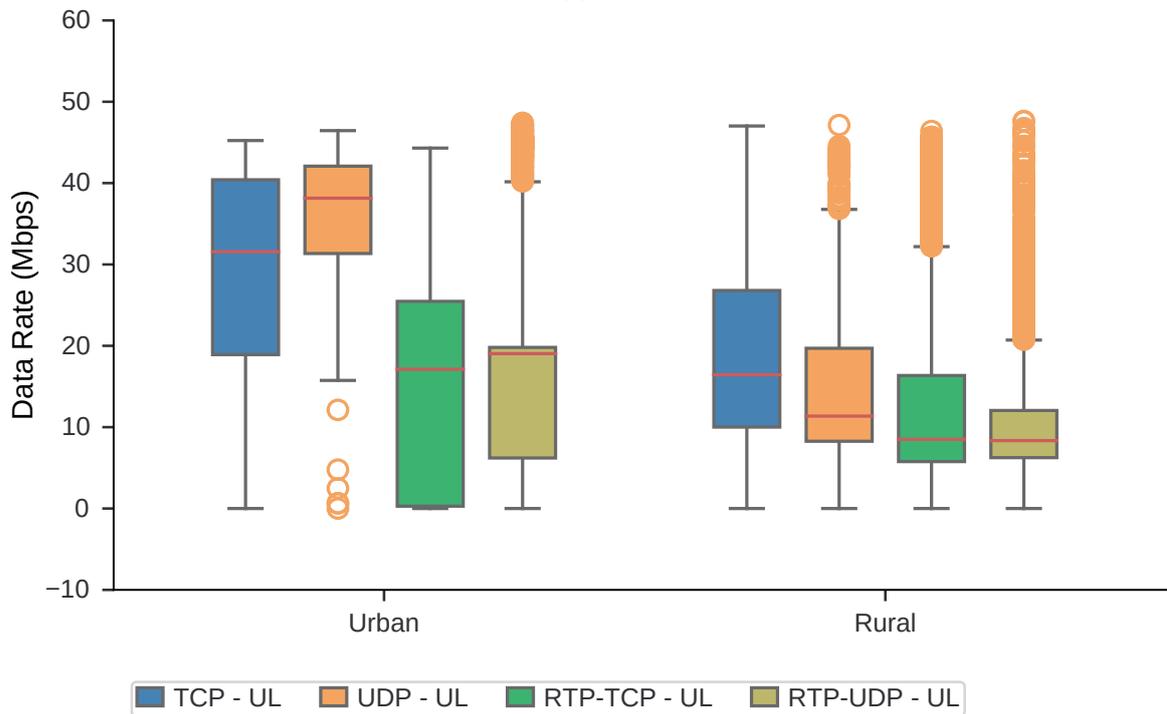
Figure 4.16(a) and Figure 4.16(b) show the DL and UL performance of TCP, UDP, and QUIC split by the test location. It is obvious that the protocols were able to perform much better in the urban area as all of them transferred more data per second on average. The context in which the QUIC results must be evaluated was described in Subsection 4.4.1 and still applies.

4.4.3 Performance Comparison of the Mobile Networks

Figure 4.17 shows the throughput of all unthrottled iPerf transfers split by the operator’s network over which they were performed. In the urban test location, no carrier



(a)



(b) The throughput of uplink connections split by the test

Figure 4.16: The throughput of (a) downlink and (b) uplink connections split into urban and rural tests. The throughput in both directions was generally higher in the urban location.

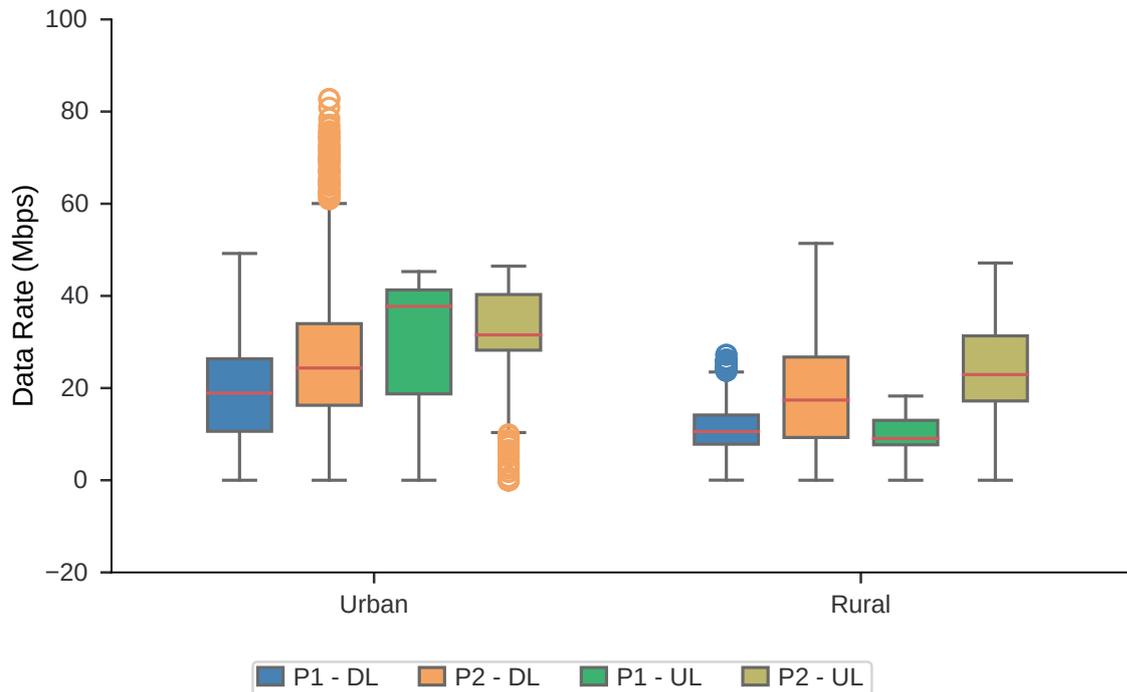


Figure 4.17: The up- and downlink throughput of the throughput tests split by the operator network over which they were conducted.

performed better than the other in both up- and downlink categories. The differences between carriers are less than 10 Mbps but it is interesting to see that P2 occasionally carried data at rates over 80 Mbps. The performance difference between carriers is clearer in the rural area where P2 outperformed P1 in both up- and downlink capacities. The median up- and downlink throughput over P1 is only around 10 Mbps, while P2 is able to support around 20 Mbps.

4.4.4 Influence of the Flight Trajectory

The throughput measurements were taken in flights that either followed a fixed trajectory or where the drone was manually controlled and altitude, heading, and velocity were tried to be randomized.

Figure 4.18 compares different connection types. TCP UL, TCP DL, and UDP DL perform significantly better in the freestyle tests. The throughput of RTP streams shows no major differences between the flight types and only QUIC performs worse in the freestyle flights. These results are surprising and any explanation for the performance improvement is hard to justify. As analyzed in Subsection 4.4.5, the altitude and velocity at which the UAV flies does not seem to have an impact on the throughput. It must be noted, that the orientation of the drone remained the same during the pre-planned trajectory flights. In contrast, repeated rotations were part of the freestyle flights, which

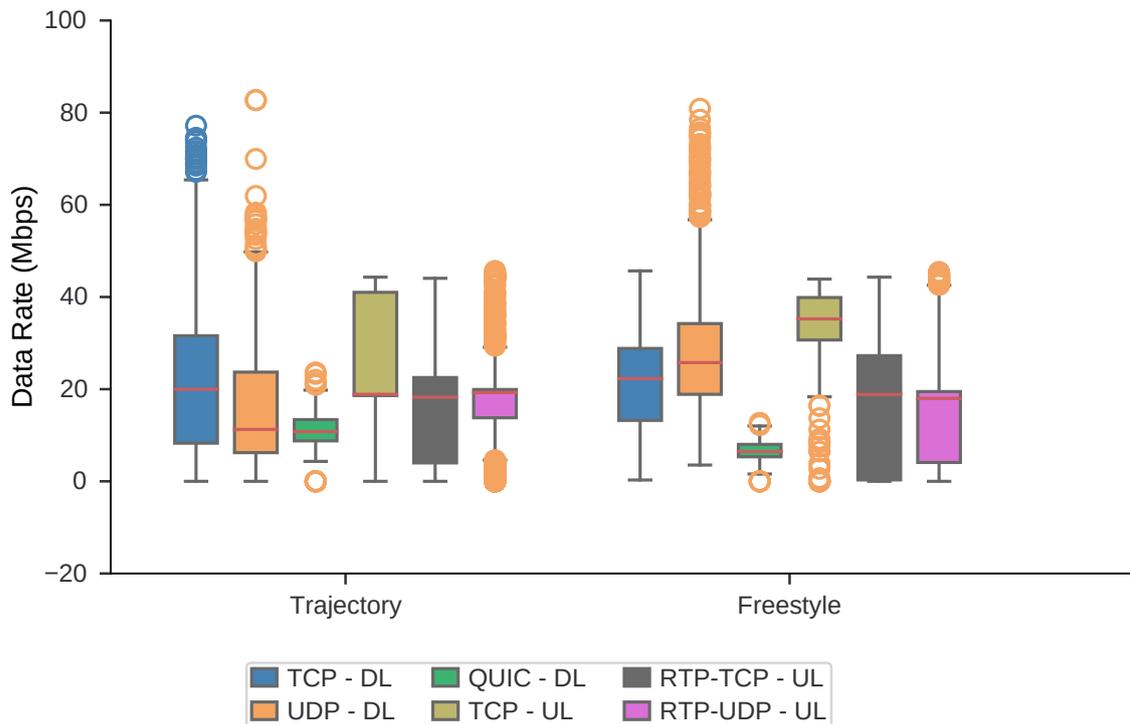


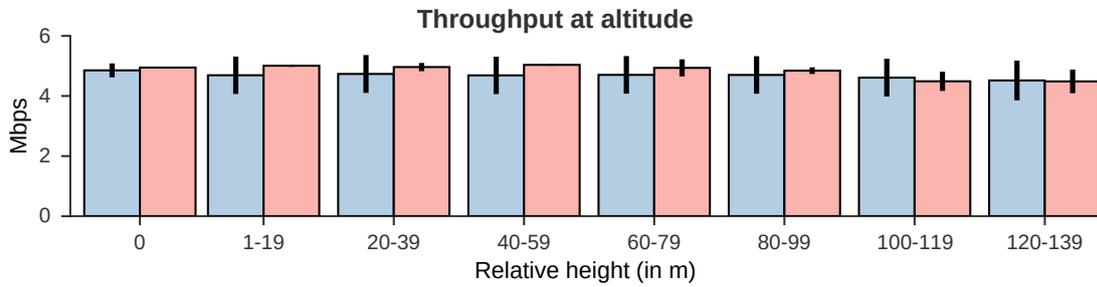
Figure 4.18: The flights during which data was gathered, were either conducted according to a fixed trajectory or the drone was guided at will (freestyle flight).

naturally also changed the orientation of the modem to the base stations. This could have had an advantageous effect on the throughput.

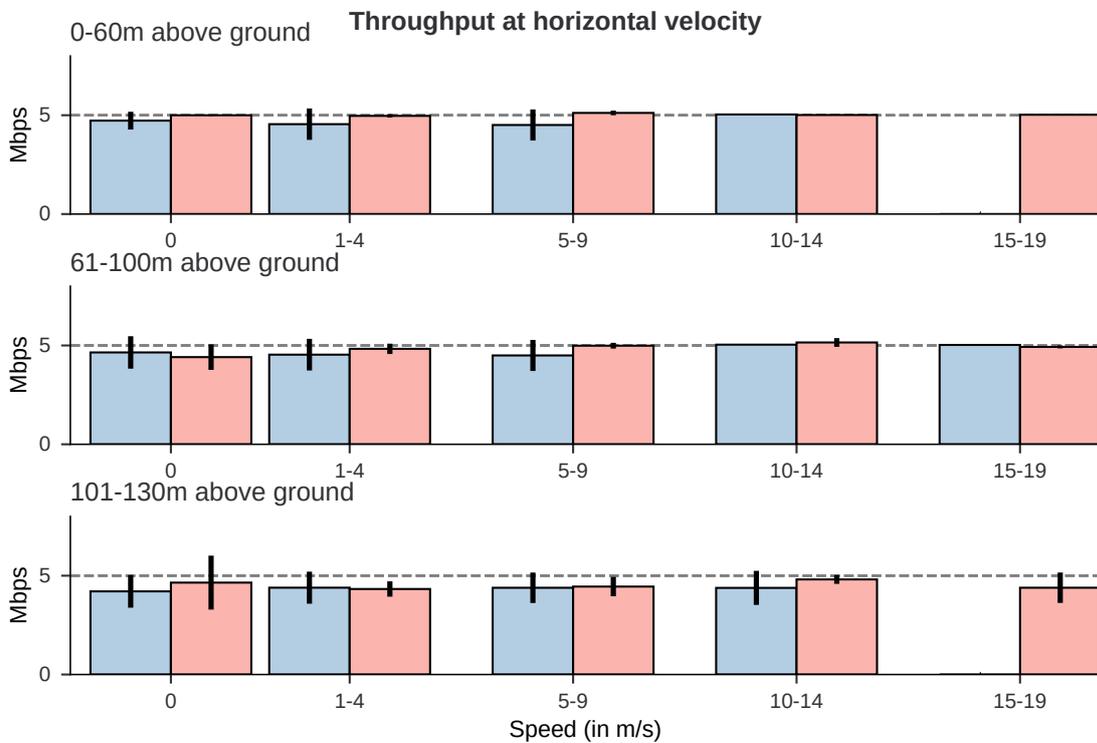
4.4.5 Influence of Altitude and Velocity

First, we investigated if a correlation between the throughput and the flight parameters altitude and velocity exists. Figure 4.19 shows the results of this analysis at the example of the UDP DL tests that generated 5 Mbps. We do not include the analysis of the unthrottled UDP and TCP UL tests as the same conclusions can be drawn from all of them.

To test correlation with altitude, the altitude range was split into 20 m-wide buckets into which the data points were sorted. The results of this analysis are visualized in Figure 4.19(a). We see that all aggregated data points are close to the 5 Mbps target mark while their standard deviation remains relatively small. No correlation between the altitude and the generated throughput is visible. Next, we examined the gathered data in search of a connection between the drone's velocity and throughput. To pin-point the influence of velocity, we separately analyzed vertical and horizontal velocity and eliminated the potential influence of altitude by further splitting the data into three different altitude ranges. Due to the design of the repeated flight trajectory, the drone

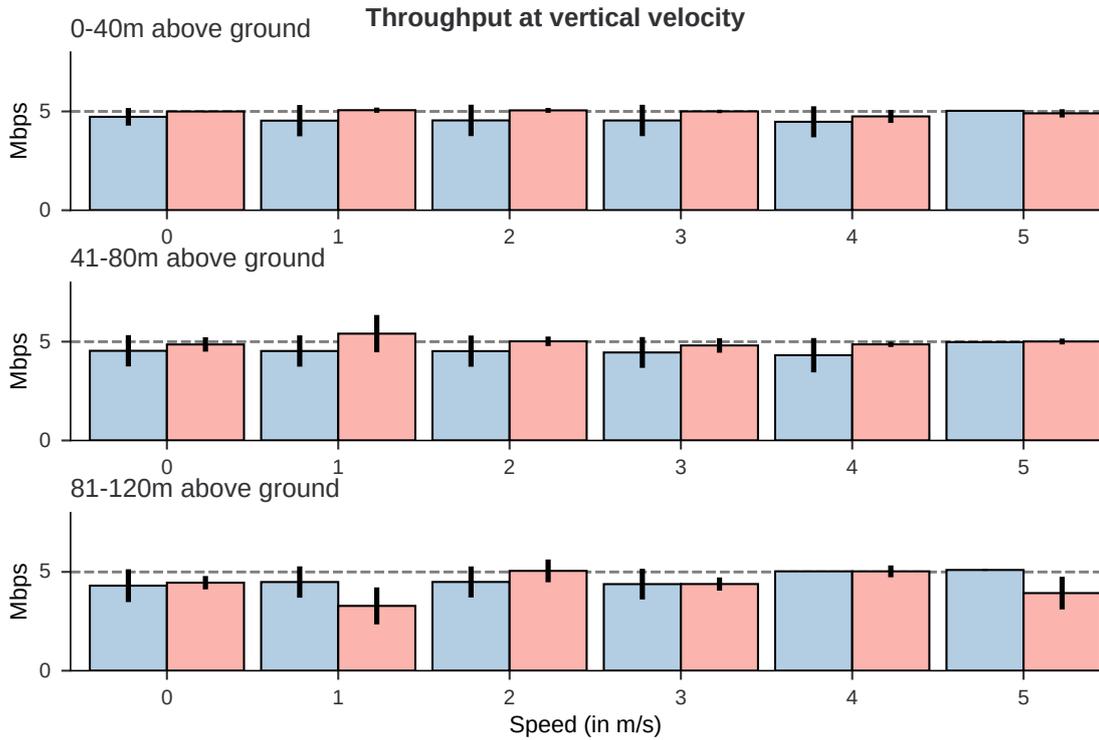


(a)



(b)

Figure 4.19: UDP downlink traffic with a 5 Mbps target split at different conditions. The left/blue column represents data from the urban test location while the right/red column shows data from the rural test location. (Continued on the next page.)



(c)

Figure 4.19: UDP downlink traffic with a 5 Mbps target split at different conditions. The left/blue column represents data from the urban test location while the right/red column shows data from the rural test location. (Continued.)

spent most of its time at 40, 80, and 120 m. We therefore separated the data around those altitudes into 0–60 m, 61–100 m, and >101 m buckets.

To analyze the influence of horizontal velocity (movement parallel to the ground), all data points during which the drone did not move in the vertical direction (at a speed of less than one km/h) were gathered and further split based on the altitude at the time of their recording. The results are presented in Figure 4.19(b). They suggest that the drone’s speed parallel to the ground has no influence on throughput up to at least 19 m/s (68.4 km/h). The same analysis was performed to uncover the relation between vertical speed and throughput. Only those data points at which the horizontal speed was less than 1 km/h were used. The results are presented in Figure 4.19(c) and show no impact on throughput. This means that the speed during ascending and descending does not seem to influence the data rate.

These three analysis steps show no correlation between altitude and throughput or velocity and throughput. The influence of those external factors on the UE does seemingly not affect the throughput on the transport layer.

4.4.6 Summary

Our analysis of the throughput generated by each transport protocol crowned no winner: TCP and UDP have an equivalent median throughput. QUIC's results are promising but unfortunately not conclusive and should therefore be further investigated.

The mobile operators' performances in the two different test locations show that one can not rely on any particular operator performing well in any location. Furthermore, it is clear that the operator's rate specifications ("up to X Mbps") should not be trusted.

Our results are promising for the real-world application of LTE-controlled drones: The throughput does not drop with an increase in velocity or altitude. We feel confident that our results are also applicable to other flight scenarios, since we performed our tests while different flight trajectories were followed.

4.5 Video Streaming

We sent video from the drone flying in the air at a live playback rate to a server, which played the received video stream. The video had an average frame rate of 60 fps and an average bitrate of 20 Mbps. This process was orchestrated by a GStreamer application that is described in Subsection 3.2.4.

The RTP timestamps, that were recorded on streamer- and player-side, can be used to calculate the playback latency. The difference between what the client played and what the streamer pushed onto the network at each point in time can be calculated, by matching a streamer log entry with what the player recorded just before. As the RTP timestamp increases with a clock rate of 90,000 units per second, the playback latency can be converted to seconds by dividing the timestamp difference by 90,000.

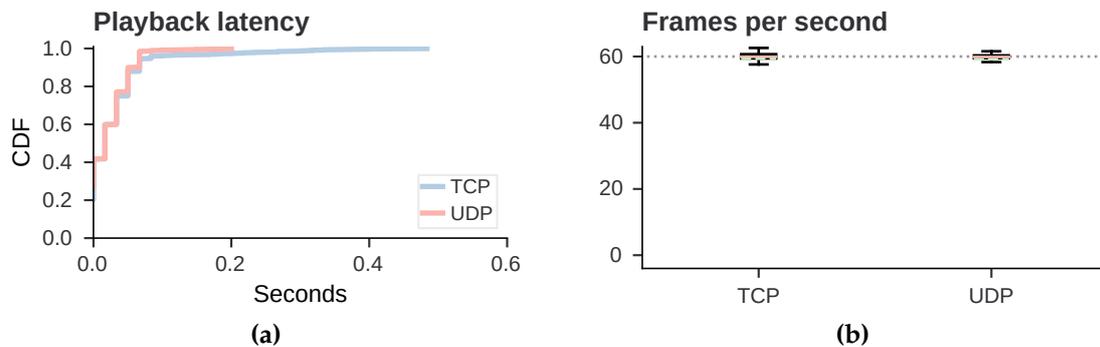


Figure 4.20: The results from tests on the ground show that our setup can deliver a smooth video stream with low latency over LTE.

4.5.1 Ground Tests

The video streaming application was tested on the ground at a different location than where the flights were performed. These tests showed that stable, smooth, and low-

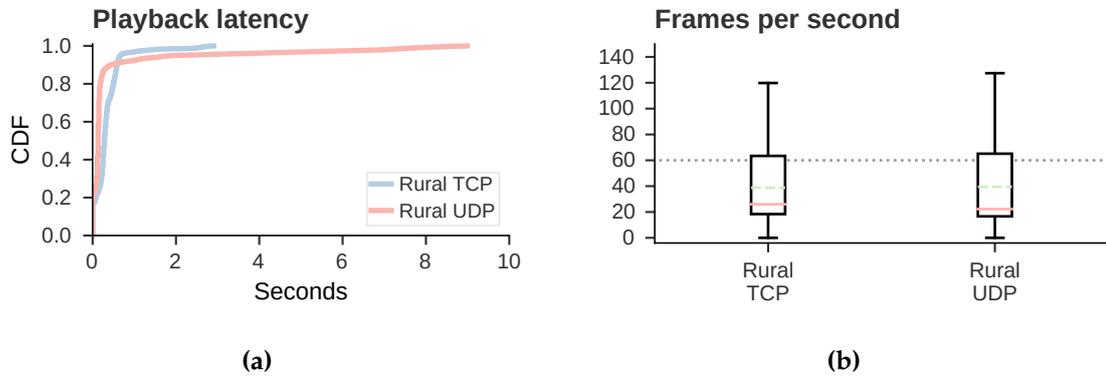


Figure 4.21: The results from video streaming tests on the ground in the rural test location.

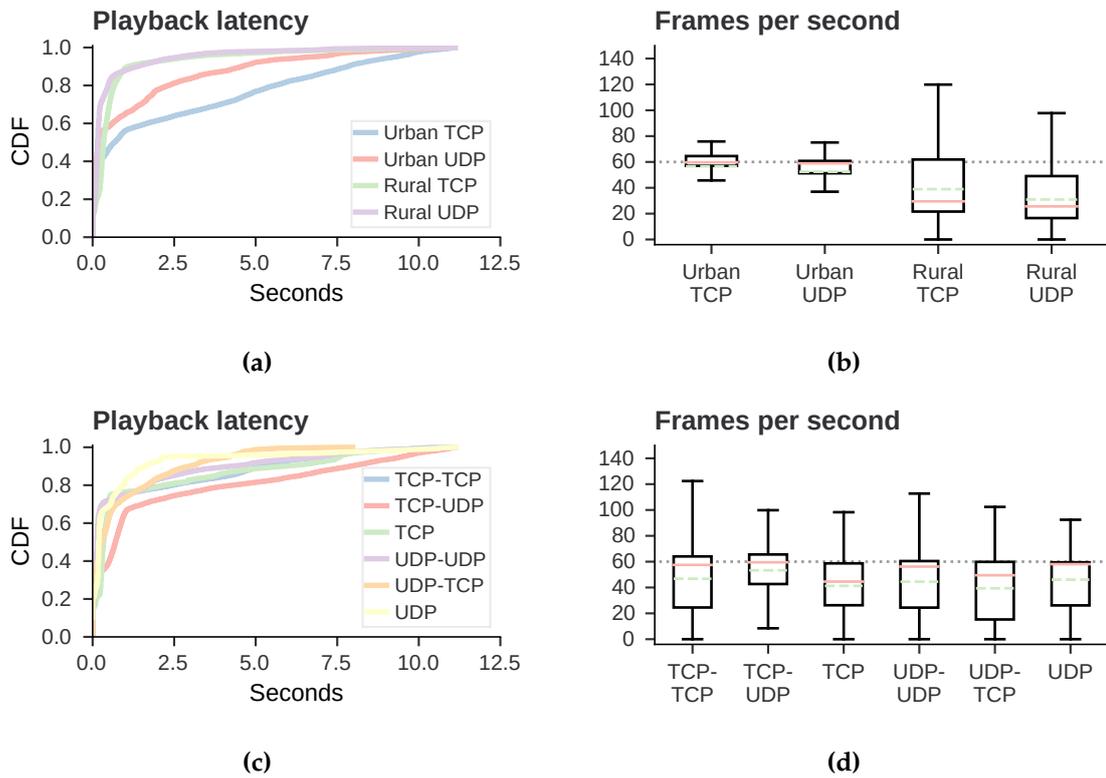


Figure 4.22: The graphs show the performance of RTP in the flight tests. The subfigures (a) and (b) compare the results split by test location and RTP’s transport protocol, while (c) and (d) investigate the impact of a concurrent 5 Mbps downlink traffic stream (“coupling”) on the streaming performance. The median/mean is marked by a orange/green line in each boxplot. The dotted line marks the video’s frame rate of 60 fps.

latency playback over LTE is possible. As visualized by Figure 4.20, the frame rate of the source material was held without positive or negative outliers. In addition, we performed ground tests in the rural test area whose performance is shown by Figure 4.21. Compared to the other ground test, the frame rate is significantly lower (on average around 40 fps) and the tail latency is much higher (up to 9 s). We did unfortunately not conduct ground tests in the urban test location.

4.5.2 Standalone RTP Tests

Figures 4.22(a) and 4.22(b) show aggregated results from the RTP flight tests split by test location and RTP's transport protocol. The most notable performance difference is made by the test location: In the urban area, the median playback rate (frames per second) is very close to the source rate and does not fluctuate much. Both protocols perform much worse in this regard in the rural area where the median frame rate is between 25 and 30 fps.

The performance situation does change when looking at playback latency: Both protocols have similar latency distributions in the rural area while UDP and TCP perform notably worse in the urban location. Analyzing the situation more closely, 89%/88% of the TCP/UDP playback happens with less than 1 s of playback latency in the rural area while the same is only true for 56%/66% of urban RTP-TCP/RTP-UDP connections. The playback latency difference stays consistent and only converges around 10 s. Still, 5% of the time, latency is higher than 9.2 seconds in case of TCP and 6.6 seconds in case of UDP. The results indicate that a trade-off exists between a low playback latency and smooth playback. Furthermore, the low playback rate in the rural area reveals that high-resolution video streaming can not be maintained, which implies an impaired Quality of Experience (QoE).

The long latency but consistent data delivery in the urban location give the impression that large queues in the LTE infrastructure could be responsible for this behavior. In this case however, the data is being sent by the Pi so that the throughput bottleneck is most likely the cellular uplink. The uplink queue in the LTE base station has therefore only a low utilization. Delivering the video's data rate of 20 Mbps should not pose a challenge to the LTE or AWS infrastructure.

The difference in playback latency can also not be explained by a difference in transport-level latency: The average latency difference between uplink RTP-TCP and iPerf-TCP connections in the urban area is only 2 ms (iPerf-TCP: 31 ms, RTP-TCP: 29 ms). Comparing the same set of connections with regard to packet loss shows a difference (see Tables 4.3 and 4.4): The packet loss during RTP connections is up to 2.6 times higher than during iPerf uplink connections. It remains however unclear how a packet loss increase in the detected dimension can have the drastic effect on the RTP-level metrics that we observe. A difference in iPerf-TCP's and RTP-TCP's payload size (because of a different MTU) is also not an explanation for the observed behavior: The packet sizes from the rural and urban locations across iPerf and RTP tests remain the same (1346 byte).

4.5.3 Coupled RTP Tests

In addition to the discussed tests, where the Pi's downlink was idle while the video was sent on the uplink, we performed "coupled" tests. In those tests, a concurrent 5 Mbps traffic stream was sent on the downlink while RTP traffic was served on the uplink. Figures 4.22(c) and 4.22(d) present the results from those tests.

The two RTP transport protocols respond differently to concurrent downlink traffic. In case of RTP-UDP, the playback rate performance (frames per second) is best without any coupled traffic, slightly worse with UDP on the downlink, and noticeably worse with TCP on the downlink. Similarly, in terms of playback latency, uncoupled RTP-UDP performs best, while RTP-UDP with coupled TCP traffic performs a bit better than with coupled UDP traffic. These results show that the usage of the downlink influences the performance of the uplink. The greater negative impact of downlink TCP traffic is probably due to the TCP acknowledgments that compete with the video traffic on the uplink.

We also tested RTP-TCP with concurrent downlink traffic. Surprisingly, in terms of playback frame rate, RTP-TCP with no downlink traffic performed significantly worse than RTP-TCP coupled with either TCP or UDP. In terms of latency, the differences are smaller: RTP-TCP coupled with UDP performs worst with the other two configurations performing similarly. The RTP-TCP coupling results are difficult to explain. We would have expected that the coupled downlink traffic competes with RTP-TCP's acknowledgments and therefore worsens the streaming performance.

4.5.4 Summary

The presented video streaming results are in many cases unexpected and difficult to explain. The most pressing open questions are: What causes the extraordinarily large performance gap between ground- and air-served video transfers? What difference in the urban and rural test locations causes the seeming trade-off between playback latency and playback rate? Why does RTP-TCP perform better with concurrent downlink traffic than without?

These questions will need to be further investigated before one transport protocol can be recommended over the other for video traffic. However, given conventional wisdom, it is already interesting that TCP does not generally perform worse than UDP in this real-time application.

4.6 LTE Emulation

The goal of an LTE emulator is to subject an application to the same conditions that it would experience over an actual LTE connection. The specific conditions are available bandwidth, average delay, delay variance, temporary link failures due to handovers, and effects due to specific LTE features such as Discontinuous Reception (DRX). Multiple

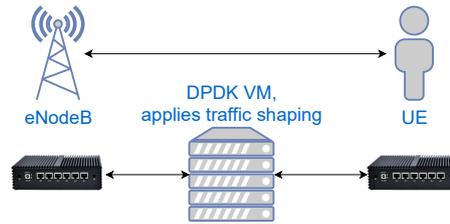


Figure 4.23: Our LTE emulation architecture that is based on [89].

works have performed measurements and developed models to approximate those different aspects of LTE (see Section 2.4).

Stratmann et al. built an LTE emulator in [89] on top of MoonGen [26], a framework to develop Data Plane Development Kit (DPDK) [96] applications in Lua [59]. They added latency queue data structures and used MoonGen’s CRC-based rate control to precisely control the send times. Multiple mechanisms of LTE were modeled to accurately reflect it’s behavior in different situations: the DRX states are for example explicitly handled. Packet loss however is simply modeled by a Bernoulli distribution. The authors based their work on measurement data provided by [16] and showed that their MoonGen emulation script provokes very similar behaviors in terms of bandwidth, latency, and packet losses.

While the MoonGen-based emulation script performs well in those regards, its scope is still limited. First, handovers are not modeled, even though they arguable have a great impact on the reliability of LTE as shown in Subsection 4.3.5. Both the frequency of handover occurrences and the duration of HIT must be modeled. Second, the emulation script does not consider temporary and changing parameters such as signal strength, cell distance, cell density, or velocity. For aerial vehicles, the altitude and surroundings are further parameters that conceivably have an influence on LTE performance.

4.6.1 Emulation Setup

We setup two machines in the chair’s network that acted as a UE and a server that communicated over an LTE channel. DPDK was running inside a Virtual Machine (VM) that was part of the same network as the communication partners. The endpoints were configured to route traffic to each other through the DPDK Virtual Machine (VM): The destination IP of a packet is set to the communication partner but the Ethernet source address (MAC) is set to the emulator’s interface. The emulator receives the packets, shapes the traffic, and forwards the packets with the destination MAC matching the receiver’s MAC address.

Figure 4.24 visualizes the architecture of the LTE emulator. Incoming packets are stored in the receive queue and read by a dedicated thread. If a packet’s destination IP matches one of communication parties’ IP addresses, it is put into one of dedicated forwarding queues. Two threads—one for each direction—read from those queues and apply the LTE emulation mechanisms, e.g., delaying packets by a certain amount. The

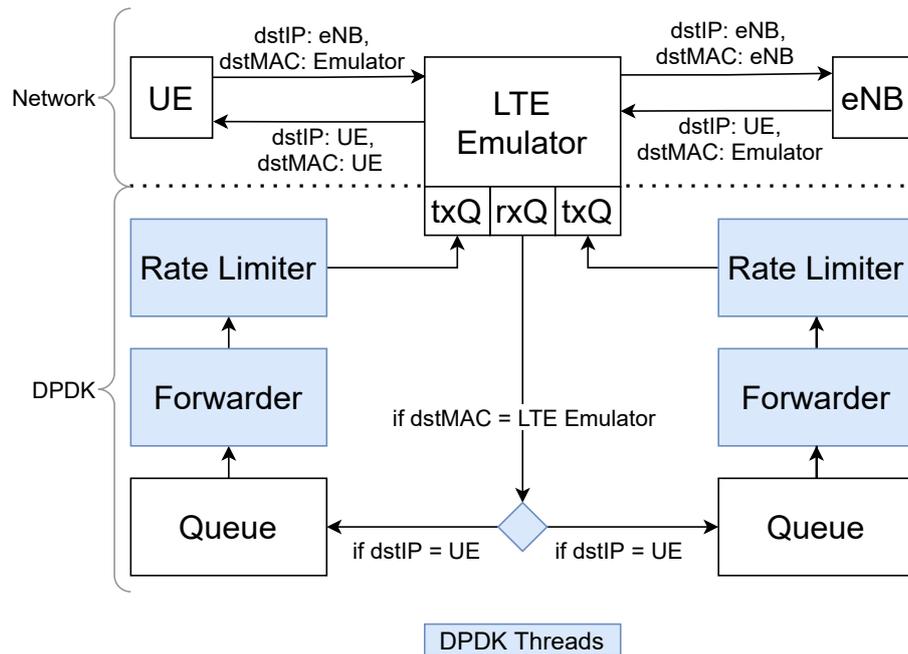


Figure 4.24: The re-architected MoonGen LTE emulation script.

packets are finally handed to a rate limiting thread that eventually appends the packets to one of the transmit queues of the NIC.

The machine running the MoonGen-based LTE emulator needs to run on a supported CPU and use a supported NIC². At least five CPU cores are necessary to run the emulation threads in addition to a main DPDK and a statistics thread. Memory-wise, no more than 2 GB are required to run the emulation script.

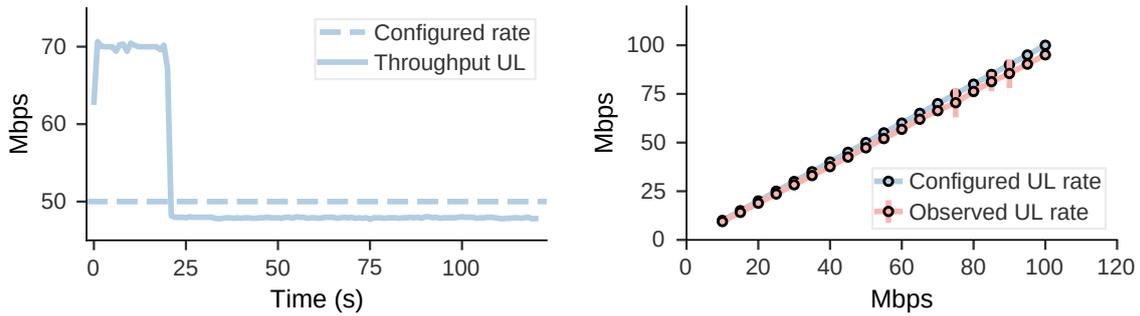
4.6.2 Emulation of Handovers

A notable part of LTE that is missing from the emulation script developed by Stratmann et al., is handover emulation. We have observed that handovers cause latency spikes (see Subsection 4.3.5) and therefore think that they have a noticeable influence on LTE's performance.

The handover procedure of LTE is described in Subsection 2.1.4. During an handover, downlink traffic sent by the server is forwarded from the Serving Enhanced Node B (S-eNB) to the Target Enhanced Node B (T-eNB). It is queued until the end of the HIT, before being forwarded to the UE. Similarly, in the reverse direction (uplink), the packets are queued at the UE during HIT, before being sent to the T-eNB.

These processes can be emulated by the MoonGen script by introducing an additional temporary delay on the up- and downlink forwarding threads, that execute the custom emulation logic (see Figure 4.24). These threads repeatedly pull packets from the queue

²See <https://core.dpdk.org/supported/>



(a) The throughput of an UDP iPerf transfer where the LTE emulator was configured to limit the throughput to 50 Mbps. This limit is only respected after the first 20 seconds. The iPerf sender generated 70 Mbps.

(b) The mean throughput after the first 60 seconds of TCP iPerf transfers. The rate is shaped accurately.

Figure 4.25: Benchmark results from uni-directional transfers using our LTE emulator.

of incoming traffic and, e.g., wait, to induce the configured latency, before passing the packets on to the rate limiter. During a handover, the forwarding threads do not process any packets. They wait until the handover is done and resume normal operation. The packets, that have accrued in the receive queue during the handover, are forwarded according to the specified rate limit.

To accurately coordinate the execution of handovers between the two forwarding threads, the next handover is always scheduled as soon as the current one ends. To do so, the number of seconds until the next handover starts is drawn from a normal distribution with user-specified mean and variance values. Our analysis of the frequency of handovers, that is described in Subsection 4.1.3, can inform the parameter choice. The duration of HIT is also modeled by a normal distribution as a function of the load on the target cell (unit: people/cell/minute), following the research in [38]. However, as described in Subsection 4.1.4, our measurements suggest that a multimodal distribution is needed to accurately model the duration of HIT.

4.6.3 Performance Evaluation

The latency emulation accuracy of the LTE emulator was tested by sending 300 ICMP ping packets, one after the other, and taking the average of the reported RTT. We emulated latencies between 10 and 60 ms with increments of 5 ms. The difference between the measured and configured RTT was 1.0185 ms on average (SD: 0.0431 ms) while the RTT to the LTE emulation machine was 0.579 ms.

The bandwidth capabilities of the emulated LTE link can be separately defined for each direction. Implementation-wise, both directions use the same logic. We first tested bandwidth shaping of the uplink and encountered an unexpected behavior: The configured bandwidth limit only came in effect after about 23 seconds (see Figure 4.25(a)). This behavior is reproducible and the duration of the initial phase—during which the

emulator is not working as expected—scales with the bandwidth target. We measured an approximate start-up phase duration of 9 seconds for 20 Mbps, and of 26 seconds for 65 Mbps. However, after the start-up phase, the bandwidth emulation seems to be very accurate. We cut off the first 60 seconds of each iPerf connection and plotted the results in Figure 4.25(b).

Duplex tests were performed in addition to the discussed unidirectional tests. For each test, we configured the LTE emulator to enforce a downlink rate limit of 40 Mbps and increased the uplink limit from 10 to 100 Mbps with a step size of 5 Mbps. The results of those tests are presented in Figure 4.25. The throughput during the first minute of each test is ignored and the final value is calculated as the average throughput of the remaining two minutes. During each test, the iPerf sender generated 20 Mbps more than the configured limit, e.g., if MoonGen is instructed to limit the throughput to 30 Mbps, iPerf sent 50 Mbps. It is apparent that neither the uplink nor the downlink throughput consistently reaches its respective limit. It appears as if a counter-interaction between the up- and down-streams exists: The downlink stays close to its target throughput in the tests between 35 and 55 Mbps, while the throughput of the uplink is up to 35 Mbps below its maximum capacity. The reverse situation occurs at 85 and 90 Mbps.

A remedy for this problem could not be found. We contacted the authors of the original paper and revised the configuration of the VM but could not pin down the problem. The unidirectional tests showed that the throughput limiting is functional after a start-up period of up to one minute. While this confirms the potential of this emulation approach, we were not comfortable to proceed to the stage of performing comparative measurements: As long as the cause of this problem has not been identified, it is likely that other aspects of the emulation do not work as expected. We believe that this problem is most likely caused by the virtualized execution environment. The work that Stratmann et al. have presented [89], gives reason for hope that “bare-metal” hardware is able to achieve the desired results.

5 Discussion

We investigated if LTE is a suitable communication link for the remote-piloting of aerial vehicles and which level of reliability and performance different transport protocols provide. We are in particular interested in the transfer of control information and live video between an aerial vehicle and a pilot using the transport protocols TCP, UDP, and QUIC.

We found that it is challenging to achieve the high level of reliability that is required for the remote-piloting with LTE: The transport layer latency does increase with altitude, with lower signal strengths, and in face of handovers. Furthermore, packet losses happen regularly and often in temporal proximity to other packet losses. The video streaming performance in our flight tests is poor, as we often observe a very high playback latency and frequent frame drops. The transport protocols TCP and UDP provide similar throughput but TCP operates with a lower latency. QUIC's results are promising but do not allow a final judgment.

In the following, we present and discuss our results with regard to their significance for reliability, video transmission and the performance of the individual transport protocols.

5.1 Reliability

Researchers have discovered increased interference levels at higher altitudes due to the LoS to multiple cells that use the same frequencies [49]. Our analysis of transport layer metrics shows that a higher altitude is not correlated with lower throughput or higher packet loss. The worst-case latencies do increase however, especially if UDP is used (see Subsection 4.3.4). Since network latencies are important to the quality of the remote-piloting use case, mobile network operators should adapt their networks to decrease interference levels. Examples of promising techniques are 3D beamforming and multi-cell cooperation [105].

Our tests in multiple locations revealed stark differences in the performances of the mobile networks. In the rural test location, P2's modems switched between up to 17 cells while P1's modems were served by the same cell throughout all measurements (see Table 4.1). These conditions led to a data rate difference; P2's throughput was almost twice as large as P1's (see Figure 4.16). The lack of alternative cells in P1's network however also endangers reliability: In case this cell fails, it is likely that P1's service would be significantly degraded due to the lack of suitable fallback cells.

A critical part of LTE's design is its break-before-make handover mechanism that

fundamentally restricts how reliable LTE can be. In our measurements, the duration of handovers did not differ in the urban and the rural locations. The HIT took 20 ms on average but was only seldom longer than 200 ms. The influence of handovers on the transport layer is noticeable as a rise in latency close to the occurrence of handovers. The extent of this increase is however larger than we would have expected. The additional delay probably arises from reconfigurations in the LTE infrastructure that we could not observe from our point of view. These spikes in latency do not seem to be predictable and cause a challenge to the reliability requirement due to the time-sensitivity of the remote-control traffic.

The suitability of LTE for serving control commands to flying vehicles with regard to reliability is further challenged by our packet loss analysis. Looking at the TCP transfers, on average 0.0592% of packets were lost on the downlink and 0.0204% on the uplink (see Tables 4.2 and 4.3). Both numbers conflict with the specifications compiled by 3GPP, who set the maximum PER at 0.001 for the use of LTE with aerial vehicles [4]. We can not determine if those losses happened on the radio channel—in which case LTE’s error control was not able to recover from all transmission errors—or in the network path between the base station and our cloud server.

From the perspective of an LTE link user, one has only limited leverage to tackle the issues of latency spikes and packet losses. Network operators and standard bodies should therefore continue to work on improvements to cellular technology. However, developers of LTE-connected aerial vehicles can also improve the system’s reliability. One opportunity would be to design the vehicle to make autonomous decisions if the communication link to the remote pilot is unavailable. The ability to perform emergency landings safely without human intervention would be an important safety feature. Another instrument to increase the reliability of the drone’s connection to the ground is the use of multiple modems that connect to the networks of different providers. In case one of the LTE links becomes unavailable, the communication could instantly switch to another link. Our results show that modems do usually not experience interruptions at the same time: Handovers do often not correlate, even if the modems are served by the same provider. An example of this behavior is shown in Figure 4.2. Beyond that, multipath protocols provide further capabilities, such as the throughput aggregation of multiple links. Relevant projects are Multipath TCP (MPTCP) [31], Multipath QUIC (MPQUIC) [20] and Multipath RTP (MP RTP) [87], which is directed at multipath video transmission. A first approach to using MPTCP to increase reliability is studied in [56].

5.2 Video Streaming

As long as an human pilot operates an aerial vehicle beyond visual line-of-sight, live video footage seems to be mandatory. In our tests, the video’s playback latency is regularly higher than one second, which appears to be too high for a safe operation of aerial vehicles. Our throughput analysis shows that the transmission of the video data rate of 20 Mbps was only achievable in the urban area. In the rural location, both

operators median throughput was significantly below the video's data rate. As our tests represent real-world conditions, the live video transmission engine must be able to adapt to those conditions.

To find guidance about which playback parameters (latency, frame rate, resolution, picture quality, etc.) should be prioritized in case the network only provides low performance, one should look at the research on cloud gaming. In cloud gaming, powerful servers run the game logic, send the rendered frames to the player, and receive their control input in return. Similar to remote piloting, a video game often requires quick reactions from the player. Researchers have investigated the importance of several parameters on the player's QoE: One study for example found that the frame rate is more important than the picture quality for fast-paced action games [88]. Those findings should guide the research on serving live video from an UE to a pilot towards the goal of prioritizing the QoE-relevant parameters.

5.3 Transport Protocols

The maximum achievable throughput is not only relevant for the playback of live video but could also enable additional use cases: A drone that has additional sensors, such as a high-resolution camera or a LiDAR scanner, could transmit the generated data to enable real-time processing on the ground. Similarly, flying taxis may require multiple or 360 degree cameras which would multiply the required bandwidth.

The throughput comparison of transport protocols revealed that no candidate outperformed the other one. TCP and UDP perform very similar on the downlink; differences on the uplink are not conclusive (see Figure 4.16). Since TCP does however provide additional features, such as retransmissions and reordering of packets, its performance must be rated higher than UDP's. We did not test the protocol's fairness, i.e., their behavior if multiple connections concurrently transmit data over the same channel. It is however likely, that TCP connections would divide the network resources better with regard to overall performance than multiple UDP connections would. Furthermore, in terms of latency, TCP consistently outperformed UDP. Due to UDP's lack of flow and congestion control, this was expected in cases where the generated data rate saturated the link. However, TCP still provided a lower latency when a low-throughput 5 Mbps stream was pushed as an approximation of control data. This suggests that, even in case of transmitting latency-sensitive data, TCP performs better than UDP.

Judging from our results, the selection of UDP over TCP does not seem recommendable for any use case. These results are surprising as it is generally accepted that TCP's reliability features make it ill-suited for real-time applications. Our analyses suggest that either flow or congestion control are necessary to perform well in LTE networks. It must be noted however, that UDP's performance was possibly impaired by traffic policies set by the network operators. UDP's advantages might become more obvious without those—if they exist.

These conclusion in mind, a protocol that possess the datagram semantics of UDP—

independent messages with no order or retransmission logic—but a mechanism such as congestion control to pace the transmission, could possibly perform well over LTE. One protocol that matches this description is DCCP [30] which has also been standardized as a transport protocol for RTP [75]. A study that investigates DCCP’s performance over an emulated LTE link has promising results. Tests in the air would show if the central factor that distinguishes UDP’s and TCP’s performance is congestion control.

6 Conclusion

We conducted drone measurements to establish if LTE is suitable for the remote-piloting of aerial vehicles. Furthermore, we examined which level of reliability and performance different transport protocols provide. We found that it is challenging to fulfill the strict reliability requirements of this task using an LTE link due to its varying latency and the occurrence of correlated packet losses. We demonstrated that LTE is capable of streaming high-quality video with a low delay but observed that it failed to maintain a constant QoE in the air. Based on our results, TCP must be recommended over UDP for the transport of remote-control traffic over an LTE link. QUIC's performance is promising but we were not able to provide the necessary results for a conclusive judgment.

We have investigated our research question by (i) instrumenting an UAV with multiple LTE-connected single-board computers that we used to conduct network tests in the air, and (ii) by developing tests that simulate the transfer of remote-control commands and live-video traffic. Using this approach, we could measure the influence of location, altitude, and velocity on the LTE link and on transport-layer metrics, such as throughput, latency, and packet loss.

We detected significant differences between the cellular performances in different locations and over different providers regarding handover behavior and throughput capabilities. Generally, LTE handovers occur frequently but are conducted reliably and only entail a brief interruption of connectivity. Nevertheless, the handover procedure seems to cause spikes in transport layer latency. The network delay also tends to increase with altitude and if the cellular signal strength drops. An effect of altitude or velocity on the throughput is however not detectable.

Comparing the transport protocols in terms of throughput, TCP and UDP perform similarly, while QUIC's performance can not be rated due to errors in our test design. The measured packet loss rate of TCP and UDP is below 0.1% while approximately 0.85% of QUIC's packets were lost. All of those values are above the PER threshold of 0.001 set by the 3GPP for the remote-piloting of aerial vehicles [4].

Our RTP-based video streaming setup provides low frame rates in the rural test location. This might be due to the low bandwidth that is offered by the rural LTE link. In the urban area, where sufficient bandwidth is available, the live video is played with high frame rates but also with a high delay that is inadequate for remote-piloting. We could not determine the reason for this behavior.

The measurement setup that we developed can be employed to repeat our measurements, but it also allows implementing and executing novel tests. The Raspberry Pi single-board computers automatically connect to an LTE network and setup the logging functionality, that provided the data on which we based our analyses.

6.1 Limitations & Future Work

We chose to use the Raspberry Pi single-board computers due to their low energy consumption and low weight. This choice unfortunately caused erroneous recordings during our first test day and likely also caused the issues with calculating latency that are described in Subsection 4.3.1. Future studies should investigate if more powerful hardware can be used during flights.

The depth of our results about QUIC falls short of our expectations. This is partly due to mistakes with our measurement design, partly due to the novelty of QUIC, as described in the introduction of Chapter 4. The results that we obtained show a promising throughput performance, especially considering QUIC's encryption overhead, but also raise questions about the implications of its relatively high PER. A focused study on QUIC's performance over cellular links in the air would therefore be well motivated. It should focus on QUIC's unique features, such as streams, and possibly also proposed extensions like the QUIC datagram mode [74].

We laid the foundation to confirm our measurement results in an emulated environment and describe the potential for future extensions, such as the possibility of modeling the impact of handovers. Unfortunately, we could not yet reproduce our measurement results due to a dysfunction of the emulator. The most promising next step would be to setup the emulation system on bare-metal hardware.

The reliability of communication could be improved by concurrently utilizing multiple LTE connections. The potential of transport protocol extensions such as Multipath TCP (MPTCP) [31], Multipath QUIC (MPQUIC) [20] and Multipath RTP (MP RTP) [87] should be explored. Another promising candidate in the transport protocol space is DCCP [30], which could at least close the performance gap between UDP and TCP or perhaps even perform better.

The ongoing development of electrically powered aerial vehicles promises to unlock application areas in a variety of commercial and governmental applications. Cellular technology could form the backbone network for many applications but as of today, more research and development is needed to fulfill that potential.

Glossary

- 3rd Generation Partnership Project (3GPP)** 3, 4, 10, 68, 71
- Automatic Repeat reQuest (ARQ)** 5, 6, 44, 48
- Access Stratum (AS)** 5
- Amazon Web Services (AWS)** 24
- Beyond Visual Line of Sight (BVLoS)** vii, 1, 10
- Component Carrier (CC)** 3
- Congestion Control (CC)** 3, 12, 26
- Control and Non-Payload Communication (CNPC)** 10, 15
- Cyclic Redundancy Check (CRC)** 5, 6
- Datagram Congestion Control Protocol (DCCP)** 15, 70, 72
- downlink (DL)** 8, 10, 11, 14, 40, 46, 52, 53, 55, 56
- Data Plane Development Kit (DPDK)** 63, 64
- Discontinuous Reception (DRX)** 62, 63
- Evolved UMTS Terrestrial Radio Access Network (E-UTRAN)** 3
- Explicit Congestion Notification (ECN)** 12
- Evolved Packet Core (EPC)** 3–5
- Evolved Packet System (EPS)** 3
- Forward Error Correction (FEC)** 12
- frames per second (FPS)** 27
- Geosynchronous Equatorial Orbit (GEO)** 1
- Global Positioning System (GPS)** 9
- Hybrid ARQ (HARQ)** 5, 6, 48
- Handover Interruption Time (HIT)** 6, 8, 14, 15, 32, 37–39, 63–65, 68, 76
- High Speed Packet Access (HSPA)** 4
- HyperText Transfer Protocol (HTTP)** 12
- Head-of-Line (HoL)** 12, 13, 20
- Internet Engineering Task Force (IETF)** 26
- Inertial Measurement Unit (IMU)** 9
- Internet Protocol (IP)** 5, 12, 13, 20
- International Telecommunication Union’s Radio Section (ITU-R)** 3
- Low Earth Orbit (LEO)** 1
- Long Term Evolution (LTE)** 3, 4, 6, 10, 16, 17, 39, 67, 71
- Line of Sight (LoS)** 9, 11, 19, 36, 37, 67
- Media Access Control (MAC)** 5, 6
- Medium Earth Orbit (MEO)** 1
- Mobility Management Entity (MME)** 4, 8
- Multipath QUIC (MPQUIC)** 14, 68, 72
- Multipath RTP (MPRTP)** 68, 72
- Multipath TCP (MPTCP)** 14, 68, 72
- Non-Access Stratum (NAS)** 5
- Next Generation Mobile Networks Alliance (NGMN)** 10
- Network Interface Controller (NIC)** 26, 64
- Non Line of Sight (NLoS)** 11
- One-Way Delay (OWD)** 44, 45
- Packet-data network Gateway (P-GW)** 4
- Packet Data Convergence Protocol (PDCP)** 5
- Packet-data network (PDN)** 4
- Protocol Data Unit (PDU)** 5, 6
- Packet Error Rate (PER)** 6, 10, 40, 42, 68, 71, 72
- Point-to-Point Protocol (PPP)** 24
- Qualcomm Diagnostic Monitor (QCDM)** 25
- Quality of Experience (QoE)** 61, 69, 71
- Quality-of-Service (QoS)** 3, 14, 21

- Random Access CHannel (RACH)** 8
Radio Access Technology (RAT) 4
Random Access (RA) 8, 15
Radio Frequency (RF) 2, 36, 39
Radio Link Control (RLC) 5, 6
Radio Resource Control (RRC) 5
Reference Signal Received Power (RSRP) 8
Received Signal Strength Indication (RSSI) 8, 34, 47, 48
RTP Control Protocol (RTCP) 14
Real-Time Transport Protocol (RTP) 11, 14, 27, 71
Round-Trip Time (RTT) 6, 13, 15, 16, 25, 44, 45, 65
Serving Gateway (S-GW) 4, 8
Serving Enhanced Node B (S-eNB) 8, 9, 64
Satellite Communication (SATCOM) 1
Stream Control Transmission Protocol (SCTP) 15
System Architecture Evolution (SEA) 3
Signal to Interference plus Noise Ratio (SINR) 8, 15
Target Enhanced Node B (T-eNB) 8, 64
Transmission Control Protocol (TCP) 6, 11–13, 20, 27, 39, 42, 67, 71
Transport Layer Security (TLS) 13
Unmanned Aerial Vehicle (UAV) vii, 1, 9–11, 15–17, 71
User Datagram Protocol (UDP) 11, 13, 20, 27, 42, 67, 71
User Equipment (UE) 4–6, 8, 11, 15, 16, 32, 39, 46, 47, 49, 50, 58, 63, 64, 69
uplink (UL) 10, 14, 40, 46, 47, 52, 53, 55, 56
Universal Mobile Telecommunications System (UMTS) 3, 4
Visual Line of Sight (VLoS) 1
Virtual Machine (VM) 63, 66
Enhanced Node B (eNB) 4, 6, 8
Electrical Vertical Take-off and Landing Vehicle (eVTOL) vii, 1, 9, 10, 17

List of Figures

2.1	The LTE architecture (adapted from [53, 38, 6]).	4
2.2	The lower layers of the LTE protocol stack.	5
2.3	The LTE handover process.	7
2.4	The propagation methods that are relevant for mobile broadband networks.	9
3.1	The urban test location.	18
3.2	The rural flight location.	18
3.3	The flight trajectories that were conducted.	19
3.4	A schematic depiction of our measurement setup.	23
3.5	Our measurement setup consists of a drone and a payload, that conducted the network tests.	23
4.1	The trajectory of two conducted flights.	31
4.2	A depiction of the handover behavior of two modems.	33
4.3	The frequency of handovers at different altitudes.	35
4.4	The frequency of handovers at different vertical velocities.	35
4.5	The frequency of handovers at different horizontal velocities.	36
4.6	The distribution of the Handover Interruption Time (HIT) from all handovers that occurred in the urban and rural test locations.	38
4.7	The packet losses over time.	43
4.8	The impact of link saturation and test location on TCP's and UDP's latency is depicted.	46
4.9	Comparing the impact of the transfer direction on the latency of TCP and UDP transfers. The data is taken from all unlimited throughput tests.	47
4.10	The correlation between latency and signal strength.	48
4.11	The latency of TCP and UDP iPerf connections from bandwidth-limited downlink tests is split by altitude.	49
4.12	The development of latency and the occurrence of handovers is depicted.	50
4.13	Two handover occurrences and the transport layer latency are depicted.	50
4.14	The throughput of the protocols is compared.	52
4.15	The throughput of QUIC and TCP during all flights of type 4.	53
4.16	The throughput of DL and UL connections split into urban and rural tests.	54
4.17	The up- and downlink throughput of the throughput tests split by the operator network over which they were conducted.	55
4.18	The influence of flight trajectory on the throughput.	56
4.19	UDP downlink traffic with a 5 Mbps target split at different conditions.	57

4.20	Results from RTP ground tests.	59
4.21	The results from video streaming tests on the ground in the rural test location.	60
4.22	The graphs show the performance of RTP in the flight tests.	60
4.23	Our LTE emulation architecture.	63
4.24	The re-architected MoonGen LTE emulation script.	64
4.25	Benchmark results from uni-directional transfers using our LTE emulator.	65

List of Tables

3.1	The test configurations of the test flights.	21
3.2	An overview of the conducted test days.	29
4.1	Total number of handovers split by the location at which they were recorded.	33
4.2	Loss during iPerf TCP downlink connections.	41
4.3	Loss during iPerf TCP uplink connections.	41
4.4	Loss during RTP-TCP connections (uplink).	41
4.5	Loss during iPerf UDP uplink transfers.	41
4.6	Loss during QUIC downlink transfers.	41
4.7	The transport layer latencies split by test location, protocol, and transfer direction/type.	44

Bibliography

- [1] 3GPP. URL: <https://www.3gpp.org/> (visited on 2021-07-09).
- [2] 3GPP. *3GPP TS 36.101 V17.2.0*. Tech. rep. 2021. URL: <https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=2411>.
- [3] 3GPP. *3GPP TS 36.331 V16.4.0*. Tech. rep. 2021.
- [4] 3GPP, T. Specification, G. Radio, and A. Network. *Study on Enhanced LTE Support for Aerial Vehicles (release 15)*. Tech. rep. Release 15. 2017. URL: https://www.3gpp.org/ftp/specs/archive/36_series/36.777.
- [5] M. Akselrod, N. Becker, M. Fidler, and R. Luebben. "4G LTE on the Road - What Impacts Download Speeds Most?" In: *2017 IEEE 86th Vehicular Technology Conference (VTC-Fall)*. 2017, pp. 1–6. DOI: 10.1109/VTCFall.2017.8288296.
- [6] I. F. Akyildiz, D. M. Gutierrez-Estevez, and E. C. Reyes. "The evolution to 4G cellular systems: LTE-Advanced." In: *Physical communication 3.4* (2010), pp. 217–244.
- [7] M. B. Albaladejo, D. J. Leith, and P. Manzoni. *Measurement-Based Modelling of LTE Performance in Dublin City*. 2015. arXiv: 1506.02804 [cs.NI].
- [8] Amazon Web Services Inc. *AWS EC2 Instance Types*. URL: <https://aws.amazon.com/ec2/instance-types/> (visited on 2019-06-07).
- [9] T. Andre, K. A. Hummel, A. P. Schoellig, E. Yanmaz, M. Asadpour, C. Bettstetter, P. Grippa, H. Hellwagner, S. Sand, and S. Zhang. "Application-driven design of aerial communication networks." In: *IEEE Communications Magazine* 52.5 (2014), pp. 129–137. ISSN: 01636804. DOI: 10.1109/MCOM.2014.6815903.
- [10] Archer. *Archer Unveils eVTOL Aircraft*. 2021. URL: <https://www.archer.com/news/archer-unveils-evtol-aircraft> (visited on 2021-07-19).
- [11] S. Awang Nor, R. Alubady, and W. Abduladeem Kamil. "Simulated performance of TCP, SCTP, DCCP and UDP protocols over 4G network." In: *Procedia Computer Science* 111 (2017), pp. 2–7. ISSN: 1877-0509. DOI: <https://doi.org/10.1016/j.procs.2017.06.002>. URL: <https://www.sciencedirect.com/science/article/pii/S1877050917311754>.
- [12] A. Baltaci, M. Kluegel, F. Geyer, S. Duhovnikov, V. Bajpai, J. Ott, and D. Schupke. "Experimental UAV Data Traffic Modeling and Network Performance Analysis." In: *IEEE International Conference on Computer Communications (INFOCOM)* (2021).

- [13] A. Baltaci, E. Dinc, M. Ozger, A. Abdulrahmen, C. Cavdar, and D. Schupke. "A Survey of Wireless Networks for Future Aerial COMMunications (FACOM)." In: *IEEE Communications Surveys and Tutorials* Submitted. (2021).
- [14] G. Barb and M. Ottesteanu. "4G/5G: A Comparative Study and Overview on What to Expect from 5G." In: *43rd International Conference on Telecommunications and Signal Processing (TSP)*. IEEE, 2020, pp. 37–40. ISBN: 978-1-7281-6376-5. DOI: 10.1109/TSP49548.2020.9163402.
- [15] Bayerisches Landesamtes für Statistik. *Tabelle 12411-001 Fortschreibung des Bevölkerungsstandes: Gemeinden, Stichtage (letzten 6)*. URL: <https://www.statistikdaten.bayern.de/genesis/online?operation=table&code=12411-001&bypass=true&levelindex=1&levelid=1626764655014> (visited on 2021-07-03).
- [16] N. Becker, A. Rizk, and M. Fidler. "A measurement study on the application-level performance of LTE." In: *2014 IFIP Networking Conference*. IEEE. 2014, pp. 1–9.
- [17] L. Bertizzolo, T. X. Tran, J. Buczek, B. Balasubramanian, R. Jana, Y. Zhou, and T. Melodia. "Streaming From the Air: Enabling High Data-rate 5G Cellular Links for Drone Streaming Applications." In: *Proceedings of ACM Conference (Conference'17) 1.1* (2021). arXiv: 2101.08681. URL: <http://arxiv.org/abs/2101.08681>.
- [18] Bitmovin Inc. *2020 Bitmovin Video Developer Report*. Tech. rep. 2020. URL: <https://go.bitmovin.com/video-developer-report-2020>.
- [19] R. Brünig, M. Kosek, and Aloÿs. *qperf*. URL: <https://github.com/rbruenig/qperf>.
- [20] Q. D. Coninck and O. Bonaventure. *Multipath Extensions for QUIC (MP-QUIC)*. Internet-Draft draft-deconinck-quic-multipath-07. Internet Engineering Task Force, 2021. URL: <https://datatracker.ietf.org/doc/html/draft-deconinck-quic-multipath-07>.
- [21] D-Link. *DWM-222 4G LTE USB Adapter*. URL: <https://eu.dlink.com/uk/en/products/dwm-222-4g-lte-usb-adapter> (visited on 2021-07-16).
- [22] DJI. *Matrice 600 Pro User Manual*. 2018. URL: https://dl.djicdn.com/downloads/m600%20pro/1208EN/Matrice_600_Pro_User_Manual_v1.0_EN_1208.pdf.
- [23] DJI. *Spreading Wings S1000 User Manual*. 2014. URL: http://dl.djicdn.com/downloads/s1000/en/S1000_User_Manual_v1.10_en.pdf.
- [24] J. A. Donenfeld. *Wireguard*. URL: <https://www.wireguard.com/> (visited on 2021-07-02).
- [25] M. Duke, E. Blanton, W. Eddy, and R. T. Braden. *A Roadmap for Transmission Control Protocol (TCP) Specification Documents*. RFC 4614. 2006-09. DOI: 10.17487/RFC4614. URL: <https://rfc-editor.org/rfc/rfc4614.txt>.

-
- [26] P. Emmerich, S. Gallenmüller, D. Raumer, F. Wohlfart, and G. Carle. “MoonGen: A Scriptable High-Speed Packet Generator.” In: *Proceedings of the 2015 Internet Measurement Conference*. IMC '15. New York, NY, USA: Association for Computing Machinery, 2015, pp. 275–287. ISBN: 9781450338486. DOI: 10.1145/2815675.2815692.
- [27] European Union Aviation Safety Agency (EASA). *Open Category - Civil Drones*. 2021. URL: <https://www.easa.europa.eu/domains/civil-drones-rpas/open-category-civil-drones> (visited on 2021-07-11).
- [28] European Union Aviation Safety Agency (EASA). *Special Condition for small-category VTOL aircraft*. Tech. rep. 2019, p. 31. URL: <https://www.easa.europa.eu/sites/default/files/dfu/SC-VTOL-01.pdf>.
- [29] S. Floyd and K. Fall. “Promoting the Use of End-to-End Congestion Control in the Internet.” In: *IEEE/ACM Transactions on Networking* 7.4 (1999), pp. 458–472. ISSN: 10636692. DOI: 10.1109/90.793002.
- [30] S. Floyd, M. J. Handley, and E. Kohler. *Datagram Congestion Control Protocol (DCCP)*. RFC 4340. 2006. DOI: 10.17487/RFC4340. URL: <https://rfc-editor.org/rfc/rfc4340.txt>.
- [31] A. Ford, C. Raiciu, M. J. Handley, O. Bonaventure, and C. Paasch. *TCP Extensions for Multipath Operation with Multiple Addresses*. RFC 8684. 2020. DOI: 10.17487/RFC8684. URL: <https://rfc-editor.org/rfc/rfc8684.txt>.
- [32] A. Fotouhi, S. Member, H. Qiang, S. Member, M. Ding, S. Member, M. Hassan, S. Member, L. G. Giordano, and A. Garcia-rodriguez. “Survey on UAV Cellular Communications: Practical Aspects, Standardization Advancements, Regulation, and Security Challenges.” In: *IEEE Communications Surveys and Tutorials* 21.4 (2019), pp. 3417–3442.
- [33] Global mobile Suppliers Association (GSA). *5G - LTE to 5G: June 2021 – Global Update*. 2021. URL: <https://gsacom.com/paper/lte-to-5g-june-2021-global-update/>.
- [34] Goldman Sachs. *Drones: Reporting for Work*. 2017. URL: <https://www.goldmansachs.com/insights/technology-driving-innovation/drones/> (visited on 2021-07-12).
- [35] C. Gomez, A. Arcia-Moret, and J. Crowcroft. “TCP in the Internet of Things: From Ostracism to Prominence.” In: *IEEE Internet Computing* 22.1 (2018), pp. 29–41. ISSN: 10897801. DOI: 10.1109/MIC.2018.112102200.
- [36] I. Grigorik. *High Performance Browser Networking*. Boston, MA: O’Reilly Media, Inc., 2013. URL: <https://hpbn.co/>.
- [37] *GStreamer*. URL: <https://gstreamer.freedesktop.org/> (visited on 2021-06-20).

- [38] D. Han, S. Shin, H. Cho, J.-m. Chung, D. Ok, and I. Hwang. "Measurement and stochastic modeling of handover delay and interruption time of smartphone real-time applications on LTE networks." In: *IEEE Communications Magazine* 53.3 (2015), pp. 173–181. doi: 10.1109/MCOM.2015.7060501.
- [39] S. Hayat, E. Yanmaz, and R. Muzaffar. "Survey on Unmanned Aerial Vehicle Networks for Civil Applications: A Communications Viewpoint." In: *IEEE Communications Surveys and Tutorials* 18.4 (2016), pp. 2624–2661. ISSN: 1553877X. doi: 10.1109/COMST.2016.2560343.
- [40] R. W. Heath. "Introduction to Wireless Digital Communication: A Signal Processing Perspective." In: Pearson, 2017. Chap. 5.5 Introd. ISBN: 9780134431871.
- [41] International Telecommunication Union. *Characteristics of unmanned aircraft systems and spectrum requirements to support their safe operation in non-segregated airspace*. Tech. rep. 2009. URL: <https://www.itu.int/pub/R-REP-M.2171-2009>.
- [42] International Telecommunication Union. *Requirements related to technical performance for IMT-Advanced radio interface(s)*. Tech. rep. Report ITU-R M.2134, 2008. URL: https://www.itu.int/dms_pub/itu-r/opb/rep/R-REP-M.2134-2008-PDF-E.pdf.
- [43] *iPerf - The ultimate speed test tool for TCP, UDP and SCTP*. URL: <https://iperf.fr/> (visited on 2021-06-07).
- [44] J. Iyengar and M. Thomson. "RFC 9000 QUIC: A UDP-Based Multiplexed and Secure Transport Abstract." In: (2021), pp. 1–151.
- [45] V. Jacobson. "Congestion avoidance and control." In: *ACM SIGCOMM computer communication review* 18.4 (1988), pp. 314–329.
- [46] S. Kacianka and H. Hellwagner. "Adaptive video streaming for UAV networks." In: *Proceedings of the 7th ACM Workshop on Mobile Video, MoVid 2015* (2015), pp. 25–30. doi: 10.1145/2727040.2727043.
- [47] A. M. Kakhki, A. Razaghpanah, A. Li, H. Koo, R. Golani, D. Choffnes, P. Gill, and A. Mislove. "Identifying Traffic Differentiation in Mobile Networks." In: *Proceedings of the ACM SIGCOMM Internet Measurement Conference, IMC 2015-October*. June (2015), pp. 239–251. doi: 10.1145/2815675.2815691.
- [48] M. Kosek, T. Shreedhar, and V. Bajpai. "Beyond QUIC v1: A First Look at Recent Transport Layer IETF Standardization Efforts." In: *IEEE Communications Magazine* 59.4 (2021), pp. 24–29. doi: 10.1109/MCOM.001.2000877.
- [49] I. Kovacs, R. Amorim, H. C. Nguyen, J. Wigard, and P. Mogensen. "Interference Analysis for UAV Connectivity over LTE Using Aerial Radio Measurements." In: *2017 IEEE 86th Vehicular Technology Conference (VTC-Fall)*. 2017, pp. 1–6. doi: 10.1109/VTCFall.2017.8287891.
- [50] P. Kumar. "QUIC - A Quick Study." In: (2020). arXiv: 2010.03059 [cs.NI].

-
- [51] J. F. Kurose and K. W. Ross. *Computer Networking - A Top-Down Approach*. 6th ed. Pearson, 2013. ISBN: 9780132856201.
- [52] J. F. Kurose and K. W. Ross. *Computer Networking - A Top-Down Approach*. 7th ed. 2017. ISBN: 9780133594140.
- [53] A. Larmo, M. Lindström, M. Meyer, G. Pelletier, J. Torsner, and H. Wiemann. "The LTE link-layer design." In: *IEEE Communications magazine* 47.4 (2009), pp. 52–59.
- [54] J. Lazzaro. *Framing Real-time Transport Protocol (RTP) and RTP Control Protocol (RTCP) Packets over Connection-Oriented Transport*. RFC 4571. RFC Editor, 2006, pp. 78–85. arXiv: arXiv:1011.1669v3. URL: <http://www.rfc-editor.org/rfc/rfc4571.txt>.
- [55] D. Lee, B. E. Carpenter, and N. Brownlee. "Observations of UDP to TCP Ratio and Port Numbers." In: *2010 Fifth International Conference on Internet Monitoring and Protection*. 2010, pp. 99–104. DOI: 10.1109/ICIMP.2010.20.
- [56] W. Lee, J. Y. Lee, H. Joo, and H. Kim. "An MPTCP-Based Transmission Scheme for Improving the Control Stability of Unmanned Aerial Vehicles." In: *Sensors* 21.8 (2021). ISSN: 14248220. DOI: 10.3390/s21082791.
- [57] D. Leon, R. Hakenberg, J. Rey, A. Miyazaki, and V. Varsa. *RTP Retransmission Payload Format*. RFC 4588. 2006. DOI: 10.17487/RFC4588.
- [58] X. Lin, V. Yajnanarayana, S. D. Muruganathan, S. Gao, H. Asplund, H.-L. Maattanen, M. Bergstrom, S. Euler, and Y.-P. E. Wang. "The Sky Is Not the Limit: LTE for Unmanned Aerial Vehicles." In: *IEEE Communications Magazine* 56.4 (2018), pp. 204–210. DOI: 10.1109/MCOM.2018.1700643.
- [59] *Lua*. URL: <https://www.lua.org/> (visited on 2021-07-20).
- [60] M. Mazur, A. Wiśniewski, and J. McMillan. *Clarity from above: PwC global report on the commercial applications of drone technology*. Tech. rep. May. 2016, p. 36. URL: www.dronepoweredolutions.com.
- [61] M. Meyer, H. Wiemann, M. Renfors, J. Torsner, and J.-f. Cheng. "ARQ Concept for the UMTS Long-Term Evolution." In: *IEEE Vehicular Technology Conference*. 2006, pp. 1–5. DOI: 10.1109/VTCF.2006.442.
- [62] D. Mishra and E. Natalizio. "A Survey on Cellular-connected UAVs: Design Challenges, Enabling 5G/B5G Innovations, and Experimental Advancements." In: *Computer Networks* 182 (2020). ISSN: 13891286. DOI: 10.1016/j.comnet.2020.107451. arXiv: 2005.00781.
- [63] R. Muzaffar, C. Raffelsberger, A. Fakhreddine, J. L. Luque, D. Emini, and C. Bettstetter. "First Experiments with a 5G-Connected Drone." In: *Proceedings of the 6th ACM Workshop on Micro Aerial Vehicle Networks, Systems, and Applications*. DroNet '20. New York, NY, USA: Association for Computing Machinery, 2020. ISBN: 9781450380102. DOI: 10.1145/3396864.3400304.

- [64] R. Muzaffar, E. Yanmaz, C. Raffelsberger, C. Bettstetter, and A. Cavallaro. "Live multicast video streaming from drones: an experimental study." In: *Autonomous Robots* 44.1 (2020), pp. 75–91. ISSN: 1573-7527. DOI: 10.1007/s10514-019-09851-6.
- [65] M. Naveed, S. Qazi, B. A. Khawaja, and M. Mustaqim. "Evaluation of video streaming capacity of UAVs with respect to channel variation in 4G-LTE Surveillance Architecture." In: *2019 8th International Conference on Information and Communication Technologies (ICICT)*. 2019, pp. 149–154. DOI: 10.1109/ICICT47744.2019.9001975.
- [66] K. Nepomuceno, I. N. de Oliveira, R. R. Aschoff, D. Bezerra, M. S. Ito, W. Melo, D. Sadok, and G. Szabó. "QUIC and TCP: A Performance Evaluation." In: *2018 IEEE Symposium on Computers and Communications (ISCC)*. 2018, pp. 45–51. DOI: 10.1109/ISCC.2018.8538687.
- [67] NGMN. URL: <https://www.ngmn.org/> (visited on 2021-07-09).
- [68] NGMN Alliance. *Verticals URLLC Use Cases and Requirements (v2.5.4)*. Tech. rep. NGMN, 2020. URL: <https://www.ngmn.org/wp-content/uploads/200210-Verticals-URLLC-Requirements-v2.5.4.pdf>.
- [69] H. C. Nguyen, R. Amorim, J. Wigard, I. Z. Kovács, P. Mogensen, I. Z. Kovacs, and P. Mogensen. "Using LTE Networks for UAV Command and Control Link: A Rural-Area Coverage Analysis." In: *2017 IEEE 86th Vehicular Technology Conference (VTC-Fall)*. Vol. 2017-Septe. 2017, pp. 1–6. ISBN: 9781509059355. DOI: 10.1109/VTCFall.2017.8287894.
- [70] E. Obiodu, A. K. Abubakar, and N. Sastry. "Is it 5G or not? Investigating doubts about the 5G icon and network performance." In: *IEEE Global Internet (GI) Symposium 2021 ()*. URL: <https://infocom.info/workshops/track/GI>.
- [71] P1 Security. *QCSuper*. URL: <https://github.com/P1sec/QCSuper> (visited on 2021-06-19).
- [72] G. Papastergiou, G. Fairhurst, D. Ros, A. Brunstrom, K.-J. J. Grinnemo, P. Hurtig, N. Khademi, M. Tüxen, M. Welzl, D. Damjanovic, S. Mangiante, et al. "De-Ossifying the Internet Transport Layer: A Survey and Future Perspectives." In: *IEEE Communications Surveys & Tutorials* 19.1 (2017), pp. 619–639. ISSN: 1553877X. DOI: 10.1109/COMST.2016.2626780.
- [73] A. Pathak, H. Pucha, Y. Zhang, Y. C. Hu, and Z. M. Mao. "A Measurement Study of Internet Delay Asymmetry." In: *Passive and Active Network Measurement*. Ed. by M. Claypool and S. Uhlig. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 182–191. ISBN: 978-3-540-79232-1.
- [74] T. Pauly, E. Kinnear, and D. Schinazi. *An Unreliable Datagram Extension to QUIC*. Internet-Draft draft-ietf-quic-datagram-03. Internet Engineering Task Force, 2021, pp. 1–9. URL: <https://datatracker.ietf.org/doc/html/draft-ietf-quic-datagram-03>.

-
- [75] C. Perkins. *RTP and the Datagram Congestion Control Protocol (DCCP)*. rfc5762. 2010-04. DOI: 10.17487/RFC5762. URL: <https://rfc-editor.org/rfc/rfc5762.txt>.
- [76] J. Postel. *User Datagram Protocol*. STD 6. RFC Editor, 1980-08. URL: <http://www.rfc-editor.org/rfc/rfc768.txt>.
- [77] S. Qazi, A. S. Siddiqui, and A. I. Wagan. "UAV based real time video surveillance over 4G LTE." In: *2015 International Conference on Open Source Systems & Technologies (ICOSST)*. IEEE. 2015, pp. 141–145.
- [78] *quicly*. URL: <https://github.com/h2o/quicly> (visited on 2021-07-20).
- [79] Raspberry Pi Foundation. *Raspberry Pi 2 Model B*. URL: <https://www.raspberrypi.org/products/raspberry-pi-2-model-b/> (visited on 2021-06-07).
- [80] Raspberry Pi Foundation. *Raspberry Pi 3 Model B*. URL: <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/> (visited on 2021-06-07).
- [81] Raspberry Pi Foundation. *Raspberry Pi OS*. URL: <https://www.raspberrypi.org/software/operating-systems/> (visited on 2021-06-19).
- [82] Y. Sahraoui, A. Ghanam, S. Zaidi, S. Bitam, and A. Mellouk. "Performance evaluation of TCP and UDP based video streaming in vehicular ad-hoc networks." In: *2018 International Conference on Smart Communications in Network Technologies (SaCoNeT)*. 2018, pp. 67–72. DOI: 10.1109/SaCoNeT.2018.8585447.
- [83] L. Schroth. *Drone Market Shares in the USA After China-US Disputes*. 2021. URL: <https://droneii.com/drone-market-shares-usa-after-china-usa-disputes> (visited on 2021-07-04).
- [84] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. "RTP: A Transport Protocol for Real-Time Applications." In: (2003), pp. 1–89. URL: <https://www.rfc-editor.org/info/rfc3550>.
- [85] W. Seidel. "Mixture Models." In: *International Encyclopedia of Statistical Science*. Ed. by M. Lovric. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 827–829. ISBN: 978-3-642-04898-2. DOI: 10.1007/978-3-642-04898-2_368.
- [86] S. S. Shapiro and M. B. Wilk. "An Analysis of Variance Test for Normality (Complete Samples)." In: *Biometrika* 52.3/4 (1965), p. 591. ISSN: 00063444. DOI: 10.2307/2333709.
- [87] V. Singh, T. Karkkainen, J. Ott, S. Ahsan, and L. Eggert. *Multipath RTP (MPRTP)*. Internet-Draft draft-ietf-avtcore-mprtp-03. Internet Engineering Task Force, 2016. URL: <https://datatracker.ietf.org/doc/html/draft-ietf-avtcore-mprtp-03>.
- [88] I. Slivar, M. Suznjevic, and L. Skorin-Kapov. "The Impact of Video Encoding Parameters and Game Type on QoE for Cloud Gaming: a Case Study using the Steam Platform." In: *2015 7th International Workshop on Quality of Multimedia Experience, QoMEX 2015* (2015), pp. 1–6. DOI: 10.1109/QoMEX.2015.7148144.

- [89] L. Stratmann, B. Walker, and V. A. Vu. "Realistic Emulation of LTE With MoonGen and DPDK." In: *Proceedings of the 14th International Workshop on Wireless Network Testbeds, Experimental evaluation & Characterization*. 2020, pp. 87–94.
- [90] Y. Su, Y. Liu, Y. Zhou, J. Yuan, H. Cao, and J. Shi. "Broadband LEO satellite communications: Architectures and key technologies." In: *IEEE Wireless Communications* 26.2 (2019), pp. 55–61.
- [91] *systemd*. URL: <https://systemd.io/> (visited on 2021-07-16).
- [92] A. S. Tanenbaum and D. J. Wetherall. *Computer Networking*. 5th ed. 2011. ISBN: 9780132126953.
- [93] M. Tayyab, X. Gelabert, and R. Jäntti. "A survey on handover management: From LTE to NR." In: *IEEE Access* 7 (2019), pp. 118907–118930.
- [94] TeleWell. *TW-LTE/4G/3G Cat 4 modem*. URL: <https://telewell.fi/en/product/3g4glte-products/TW-LTE4G3Gmodeemi/tw-lte4g3g-cat-4-modem> (visited on 2021-07-16).
- [95] The Bufferbloat community. *Technical Introduction to Bufferbloat*. URL: <https://www.bufferbloat.net/projects/bloat/wiki/TechnicalIntro/> (visited on 2021-07-20).
- [96] The Linux Foundation Projects. *DPDK – Data Plane Development Kit*. URL: <https://www.dpdk.org/> (visited on 2021-07-20).
- [97] The Tcpdump Group. *tcpdump*. URL: <https://www.tcpdump.org/> (visited on 2021-07-02).
- [98] *Transmission Control Protocol*. RFC 793. 1981-09. DOI: 10.17487/RFC0793. URL: <https://rfc-editor.org/rfc/rfc793.txt>.
- [99] F. Wilcoxon. "Individual Comparisons by Ranking Methods." In: *Biometrics Bulletin* 1.6 (1945), pp. 80–83. ISSN: 00220493. DOI: 10.1093/jee/39.2.269.
- [100] Wireshark Foundation. *Wireshark*. (Visited on 2021-06-19).
- [101] M. P. Wylie-Green and T. Svensson. "Throughput, Capacity, Handover and Latency Performance in a 3GPP LTE FDD Field Trial." In: *2010 IEEE Global Telecommunications Conference GLOBECOM 2010*. 2010, pp. 1–6. ISBN: 9781424456383. DOI: 10.1109/GLOCOM.2010.5683398.
- [102] V. Yajnanarayana, Y. E. Wang, S. Gao, S. Muruganathan, and X. Lin. "Interference Mitigation Methods for Unmanned Aerial Vehicles Served by Cellular Networks." In: *2018 IEEE 5G World Forum (5GWF)*. IEEE, 2021, pp. 118–122. DOI: 10.1109/5GWF.2018.8517087.
- [103] M. Yousef, F. Iqbal, and M. Hussain. "Drone Forensics: A Detailed Analysis of Emerging DJI Models." In: *2020 11th International Conference on Information and Communication Systems (ICICS)*. 2020, pp. 66–71. DOI: 10.1109/ICICS49469.2020.239530.

- [104] A. Yu and T. A. Benson. "Dissecting Performance of Production QUIC." In: *Proceedings of the Web Conference 2021*. 2021, pp. 1157–1168.
- [105] Y. Zeng, J. Lyu, and R. Zhang. "Cellular-connected UAV: Potential, challenges, and promising technologies." In: *IEEE Wireless Communications* (2019). ISSN: 15580687. DOI: 10.1109/MWC.2018.1800023. arXiv: 1804.02217.
- [106] Y. Zeng, Q. Wu, and R. Zhang. "Accessing from the sky: A tutorial on UAV communications for 5G and beyond." In: *Proceedings of the IEEE 107.12* (2019), pp. 2327–2375.
- [107] L. Zhang, T. Okamawari, and T. Fujii. "Performance evaluation of TCP and UDP during LTE handover." In: *2012 IEEE Wireless Communications and Networking Conference (WCNC)*. 2012, pp. 1993–1997. DOI: 10.1109/WCNC.2012.6214116.
- [108] M. Zolanvari, R. Jain, and T. Salman. "Potential Data Link Candidates for Civilian Unmanned Aircraft Systems: A Survey." In: *IEEE Communications Surveys Tutorials* 22.1 (2020), pp. 292–319. ISSN: 1553877X. DOI: 10.1109/COMST.2019.2960366.