Measuring the Feasibility of Teleoperated Driving in Mobile Networks

Stefan Neumeier*, Ermias Andargie Walelgne‡, Vaibhav Bajpai†, Jörg Ott† and Christian Facchi*
*Technische Hochschule Ingolstadt, Research Centre
(stefan.neumeier | christian.facchi)@thi.de
†Technische Universität München
(bajpaiv | ott)@in.tum.de
‡Aalto University
ermias.walelgne@aalto.fi

***Abstract*—Teleoperated Driving is the remote control driving of a vehicle by a human driver. The concept of Teleoperated Driving requires the use of mobile networks, which typically experience variable throughput, variable latency and uneven network coverage. To investigate whether Teleoperated Driving can be possible with contemporary mobile networks, we have conducted measurements while driving with vehicles in the real world. We used complementary measurement setups to obtain results that can be compared. The dataset consists of about 5200 km (4660 minutes) driving measurements. Results show that Teleoperated Driving could be possible, but the high variance of network parameters makes it difficult to use the system at all times. It appears that the speed of the vehicle and the distance to the base station may not influence Teleoperated Driving, while handover with changed radio technology, signal strength and distance to the teleoperation station may have an impact. Possible mitigations to overcome these problems along with a basic whitelisting approach is discussed.**

I. Introduction

The Working Group "Connectivity and Automated Driving" of the European Road Transport Research Advisory Council (ERTRAC) [11], a technology platform developing a common vision for road transport in Europe, show that vehicles will advance in autonomous driving features until they are fully autonomous. However, until fully autonomous vehicles are available, there can be situations where autonomous features would be useful but not existent. A solution to provide autonomous-like behavior of vehicles is Teleoperated Driving, where a vehicle is controlled remotely by a human driver when required. Even with fully autonomous vehicles, there will be situations where a system can not handle a situation and human intervention is necessary, e.g. complex road side work [17] or other obstacles [20]. In such a scenario, the remote operator takes over control and operates the vehicle as long as required, typically covering short distances. To safely control a remote vehicle in traffic, it is important that the teleoperator monitors the environment of the remote vehicle constantly and is able to deal with different situations. This can be difficult due to the non-deterministic behavior of the utilized mobile network. Sufficient bidirectional throughput is required to exchange data between the remote vehicle and the teleoperation station. One direction is required for the remote vehicle to provide the driver with environmental information, whereas the other direction is used for transmitting steering commands. Further challenges are latency [31] and jitter. Teleoperated vehicles use regular streets and thus have to deal with suddenly appearing obstacles. Latency is one of the key indicators to determine how safely one can control the vehicle, as transmission of steering commands and streams can get delayed. Due to the mobility of vehicles, a high frequency in the changes of network conditions can be expected [34]. This makes Teleoperated Driving even more challenging. Considering these obstacles, we want to know: Is Teleoperated Driving feasible with contemporary mobile networks? To answer this question the paper provides a first assessment on this topic. We investigate latency and throughput values while driving in the real world using three complementary measurement setups. We assess, whether factors such as handover, distance of a remote car to the teleoperation station or signal strength have an influence on the usability of the system. We show that Teleoperated Driving can be feasible and observe that signal strength, the distance to the teleoperation station and handover with changed radio technology can have an influence on Teleoperated Driving, while the speed of the vehicle and distance to base station do not have an influence. We further investigate whether an approach of white listing areas with good network conditions can help Teleoperated Driving.

The paper is structured as follows. Section II provides an overview of related work. Section III defines the network requirements for Teleoperated Driving. Section IV introduces the measurement setup and the collected dataset. Section V presents the results, discusses the influence of different parameters and proposes mitigations. Section VI discusses limitations and future work. Finally, Section VII concludes the paper.

II. Related Work

Winfield [33] presents the basic components of a teleoperated system: the robot (remote vehicle), the remote place of work (teleoperation station) and the connectivity between the two components, while different approaches for a Teleoperated Driving system design [15], [26] already exist. Chucholowski *et al.* [9] measured the latency of video-streams over 3G networks while driving. Their measurements reveal a highly

978-3-903176-17-1 / © 2019 IFIP

varying average latency of 121 ms and state that 3G connections may be sufficient for Teleoperated Driving. Kang *et al.* [20] measured latency transmitting a video stream over LTE and experienced 100 ms of delay. Keon Jang *et al.* [19] investigated the throughput of 3G and 3.5G while driving with cars and high speed trains. A considerable difference between stationary and mobile measurements was observed, whereby lower throughput over UDP and TCP, higher jitter and packet were witnessed in mobility scenarios when compared stationary conditions. Xiao *et al.* [34] measured the performance of cellular networks by conducting measurements at more than 300 km/h of speed and compared the results to stationary and mobility measurements at lower speeds of 100 km/h. They drove 120 km with vehicles and nearly 5000 km with a high speed train utilizing `iperf` and `traceroute` to measure throughput and latency via smartphones. Lauridsen *et al.* [21] drove about 19,000 km in rural, suburban and urban environments. Radio network scanners and smartphones were utilized to study latency, handover execution time, and coverage of four operational LTE networks. They witnessed LTE coverage of about 99% and an average handover latency of about 40 ms. Li *et al.* [22] compared CUBIC and BBR TCP congestion control while driving on a highway. They measured latency using ICMP and by measuring TCP connect times. In addition, throughput was measured by downloading a file. All measurements were conducted using a smartphone. They observed that latency is predominant in the range of 40 ms to 80 ms. TCP throughput in downloading a file is at a median of about 11 MBit/s. Parichehreh *et al.* [27] conducted measurements to compare three different congestion control algorithms in the LTE uplink. They show that the intended behavior of BBR can be seen, but device packet losses have been observed. Merz *et al.* [25] show that the performance of LTE stays robust up to 200 km/h, identifying the signal-to-noise ratio as an important factor to ensure robustness. While a lot of measurements have been conducted already, it is hard to map these results to Teleoperated Driving specifically. For instance, some studies [9], [19] focus on 3G networks only, some studies lack either throughput [21] or upload measurements[22], [25], while others include limited amount of driving (120 km) [34] or stick to specific routes [27]. Teleoperated Driving is presented in [17], but real measured values are missing. Perhaps results of previous work can still be compared to our study.

Approaches to overcome the issues of mobile networks in Teleoperated Driving have also been proposed. For instance, predictive displays [10], [12] can be used to show the path of the vehicle with respect to the latency and their use can effectively assist while driving. Buffers can be used to overcome the challenges with jitter [14] by smoothening the variability in delay to improve driving performance [12], [23]. An additional possible mitigation, where the remote vehicle automatically reacts to upcoming hazards, which the driver is not yet aware of due to the time delay, is presented in [18]. Another suitable approach is the use of a free corridor [30], where the driver decides the path taken by the car in situations where the connection is lost. In case of uplink throughput, an adaption of resolution is the first step. If conditions occur where such a mitigation is not enough or the resolution becomes too low, the rear camera can be lowered in resolution or even switched off, if not needed. Finally, it is also possible to lower the maximum speed of the remote vehicle and in parallel lower the frames per second of the video stream. For higher speeds it is important to see a fluent image, which is 25fps [29], but with lower speed the frames per second can be reduced and steering is still possible. Finally, it is also possible to use multiple sim-cards [17] of different providers. This enables the option to always use the best network available and transmit important information using multiple paths.

III. Requirements for Teleoperated Driving

We frame the network requirements for Teleoperated Driving within the scope of which Teleoperated Driving may deemed to be feasible with contemporary mobile networks. These requirements consist of minimum/maximum values, that may safely allow Teleoperated Driving. For sending control commands to the remote vehicle, a low amount of data is required. By continuously sending packets of steering commands to the remote vehicle every 10 ms, about 0.25 MBit/s are required. This value is based on the amount of data per packet and the transmission frequency. We decided to send a command packet every 10 ms to get a fluent control stream of steering commands. This allows even small adjustments of the remote vehicle. In addition, with 10 ms between two packets, packet loss/delay can be carried more efficiently as a single packet only counts for 10 ms of driving. The average amount of data per packet originates in measurements, where some basic information were transmitted, e.g. steering wheel angle, position of brake/gas pedals as well as additional triggers like enabled/disabled windshield wiper, etc. Tighter requirements exist for the uplink, which is used at least for streaming video data. For instance, it is known that for transmitting a view of 150° at least 3 MBit/s of uplink [9] are required. This 150° view is deemed sufficient to safely control vehicles in straight driving scenarios, but does not meet governmental regulations, e.g. in Germany. Utilizing a resolution of 640 x 480 and three 90° cameras (front: two, back: one), the amount of transmitted data can be kept at the level of 3 MBit/s [15]. Following the documentation of Youtube for its live-encoder settings [35] and Adobe's recommendations for live streaming [6], 1 MBit/s is deemed enough to carry one stream with sufficient resolution. Thus, sharing three camera streams adds up to about 3 MBit/s. To define the requirements for latency, we further conducted a small user study with five users. The study consisted of driving through `pylons` with different levels of latency, using OpenROUTS3D [3], a self-developed 3D driving simulator. It turned out that values above 300 ms make controlled driving nearly impossible. Subtracting the latency of sensors and actuators (roughly 50 ms), the maximum tolerable network latency is 250 ms. This value correlates with values determined by others [13] for gaming.

We also identified that, if the jitter stays below 150 ms, it is possible to safely control a car.

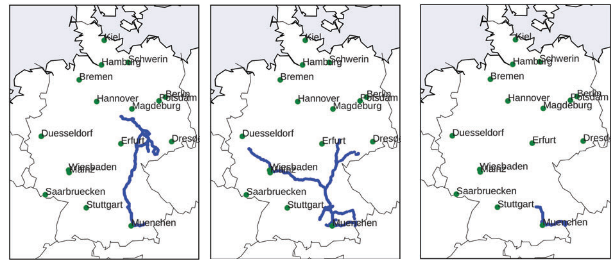
IV. Measurement Setup

We describe the hardware and software setup used to perform measurements of throughput (TCP) and latency (ICMP and UDP) while driving.

Hardware – An Android-based *Lenovo B* smartphone [1] and a *SierraWireless RV50X* LTE gateway [28], installed in a test vehicle, were used to conduct the measurements. The Android-based smartphone was used, to be more flexible as measurements could be carried out independently of the test vehicle. For conducting meaningful measurements, the smartphone needed to allow connectivity to all important mobile network technologies, which the *Lenovo B* does. The LTE gateway on the other hand was fixed to the trunk of the test vehicle. There it was connected to two antennas. One on the roof of the car, which provided LTE and GPS signals, and a second antenna inside the vehicle's trunk to provide diversity and lower the impact of interference. The Gateway was connected to an Ubuntu-based car PC via Ethernet.

Software – Three different tools, `ping`, `netradar` and `iperf3` were used. Tools were not executed in parallel to avoid side effects of reduced throughput or higher latency. The first measurements conducted for this paper consist of ICMP messages sent using a self-developed application [4] for Android-based smartphones. This application is able to gather environmental data for the measurements and execute the `ping` command to determine the Round Trip Time (RTT). While the application allows periodic (configurable) measurements, the configuration applied for this paper had no pause between measurements. Further measurements on the smartphone were collected using the `netradar` [8] measurement platform. `Netradar` is a crowd-sourced mobile measurement platform that measures and collects metrics related to mobile network performance across mobile devices. The measurement mainly focuses on the analysis of TCP throughput [32], UDP latency and contextual information related to each measurement. The server to which `netradar` connects is hosted at the Amazon Cloud in Europe. The throughput-measurements (uplink and downlink) using `iperf3` were conducted on the car PC, where the *SierraWireless RV50X* was configured as LTE gateway. The endpoint for measurements was a server hosted in Munich. Measurements ran continuously and gathered contextual information.

Dataset – Data collection considered only values gathered during driving with a car on all types of streets and areas (rural, suburban, urban; evenly distributed at its best) in Germany to avoid the influence of roaming implications [24]. This ensures to be as close to real world Teleoperated Driving scenarios as possible. The measurements cover the period of end of May 2017 to end of December 2017. The total driving time of ∼78 hours accompanies with ∼5200 km of driving. Measurements are split up into 2180 km (1528 minutes) for `ping`, 2670 km (2940 minutes) for `netradar` and 354 km (191 minutes) for *SierraWireless*. Most of the driving was conducted during daytime. The different routes can be seen in Fig. 1. The average driving speed is a little bit higher (∼66 km/h) than the average speed (∼42 km/h) reported [5] by the Germany Automobile Association (ADAC). That is caused by a higher amount of kilometers on the highway compared to an average driving scenario. Although minor deviations exist, the values are comparable. Discrepancy between the high number of kilometers measuring `ping/netradar` and the *SierraWireless* can be explained by the easiness of using the different measurement platforms. The measurements with the *SierraWireless* were only possible when driving with the test vehicle, where availability limits the applicability. In contrast, the smartphone could be carried within any vehicle. The measurements were conducted using Vodafone DE as telecom provider offering unlimited traffic with limitations of 100 MBit/s in downlink and 50 MBit/s in uplink.



(a) Ping (b) Netradar (c) SierraWireless

Fig. 1. Trajectory of (a) `ping`, (b) `netradar` and (c) `iperf3` measurements performed while driving in Germany.

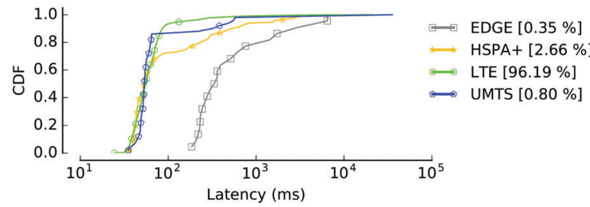


Fig. 2. CDF of the `ping` latencies with overall median of ∼55.14 ms in RTT (EDGE: ∼364 ms, HSPA+: ∼53 ms, LTE: ∼55 ms, UMTS: ∼54 ms).

V. Results

A. Latency

Inflated latency can cause delivery of steering commands and video streams to be delayed. We present results of latencies measured using both ICMP and UDP, since ICMP packets may be treated differently [16] than UDP packets on the path towards the destination.

`ping` – The ICMP packets of the *ping* application were sent either to a server hosted in Frankfurt or Munich. The server hosted in Munich was the target in about 60% of the measurements. The other 40% of the measurements used the server hosted in Frankfurt as target destination.

The connection type of the samples was about 96% LTE, about 3% HSPA+, about 0.8% UMTS and 0.35% EDGE. Mapping this to kilometers means that LTE was used in about 2090 km. UMTS seems to provide results with lower latency

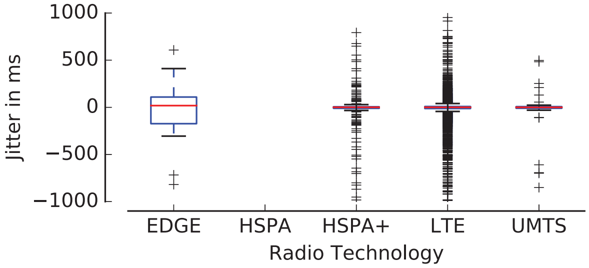


Fig. 3. Variance of `ping` latencies with overall median of ∼10 ms in Jitter.

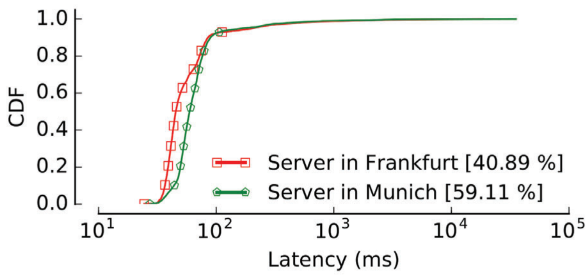


Fig. 4. CDF of `ping` latency based on the destination server in Munich/Frankfurt (Frankfurt: ∼45.3 ms, Munich: ∼59.4 ms in median).

than LTE, but this is caused by the low number of samples of UMTS measurements (0.8% UMTS, 96% LTE). In addition to the high LTE coverage, the results of the latency measurements are promising as well.

The median latency is ∼55.14 ms in RTT (Fig. 2). About 96% of the total RTT values are below the critical 250 ms. Unfortunately, there are about 4% of higher RTT values, partly greater than one second (1%). This makes Teleoperated Driving infeasible. Besides the raw latency values, jitter (Fig. 3) has to be considered as well. Jitter here references the variance of latency around the median values. For presentation, the jitter has been cut to 1000 ms, as about 99% of the values are below this threshold. The median jitter is at acceptable ∼10 ms (all technologies), but in about 5%, jitter is not suitable for Teleoperated Driving. Jitter comparison between UMTS and LTE faces the same issue as mentioned above.

Given that there may be multiple teleoperation stations deployed at different places, investigating whether the location of a teleoperation station influences the latency is crucial. We chose two servers in cities with about 300 km of distance between each other. Latency to both servers is roughly comparable, but the one to the server in Frankfurt is lower (∼45 ms to ∼59 ms) as shown in Fig. 4. The average distance of the vehicle to the server in Munich and Frankfurt is 119 km and 270 km, respectively. Fig. 5 shows that Munich has greater maximum distances than Frankfurt. Comparing the measurements based on the distance of the vehicle to the server, it can be seen that a higher distance between vehicle and teleoperation station does not significantly lead to higher latency, but its variation drastically increases. As both servers are connected to the Internet with identical parameters, this

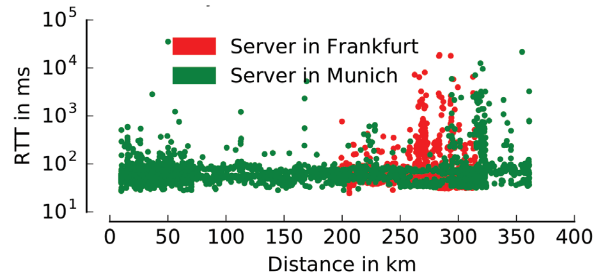


Fig. 5. Distribution of `ping` latency based on destination: Munich/Frankfurt.

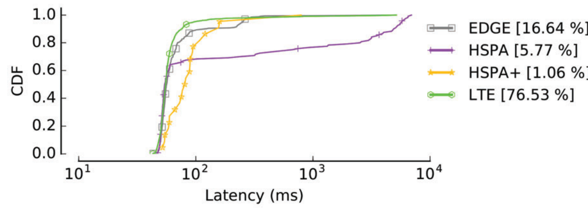


Fig. 6. CDF of the `netradar` latency measurements with overall median of ∼55 ms in RTT (EDGE: ∼56.5, HSPA: ∼54.4, LTE: ∼55.1, HSPAP: ∼81.3).

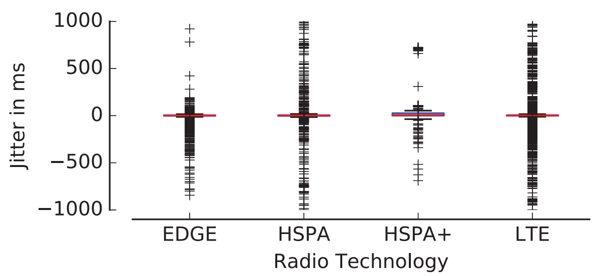


Fig. 7. Jitter of the `netradar` latency with overall median of ∼2 ms.

shows that the location of the teleoperation station is crucial.

`netradar` – The distribution of radio technology types with the `netradar` measurements was about 77% LTE, 17% EDGE, 6% HSPA and 1% HSPA+. Compared to the `ping` application, the LTE coverage was lower in the magnitude of 20%, leading to about 2056 km with LTE as connection type. Besides the differences in the coverage, the median latency, which is ∼55 ms in RTT (see Fig. 6), is comparable to the `ping` results. In about 96% of the samples, the latency was below the critical value of 250 ms. Although the LTE coverage was lower, the results are comparable to the `ping` measurements. The median jitter (Fig. 7, excluding values above 1000 ms (0.2%)) was ∼2 ms and thus more stable than on the `ping` measurements. About 4% of the measurements have a jitter greater than 150 ms.

B. Throughput

The quality of transmitted video streams has to be adapted to the available throughput dynamically. Conducting Teleoperated Driving is infeasible if throughput is too low or changes too frequent for algorithms to adapt.

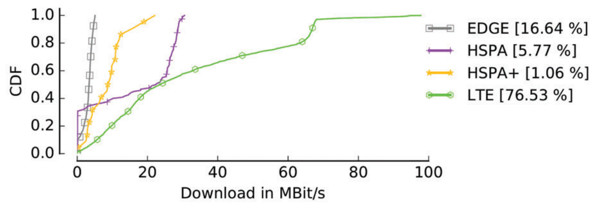


Fig. 8. CDF of `netradar` downlink measurements with overall median of ∼17 MBit/s (EDGE: ∼3.4 MBit/s, HSPA: ∼22.5 MBit/s, LTE: ∼23.5 MBit/s, ∼HSPAP, ∼9 MBit/s).

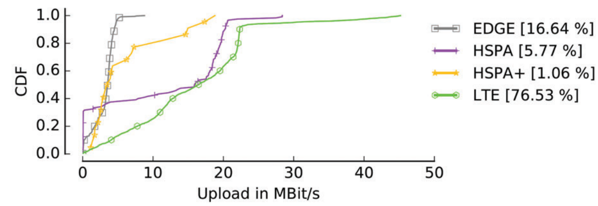


Fig. 10. CDF of `netradar` uplink measurements with overall median of ∼12 MBit/s (EDGE: ∼3.5 MBit/s, HSPA: ∼15.9 MBit/s, LTE: ∼16.5 MBit/s, HSPAP: ∼3.7 MBit/s).

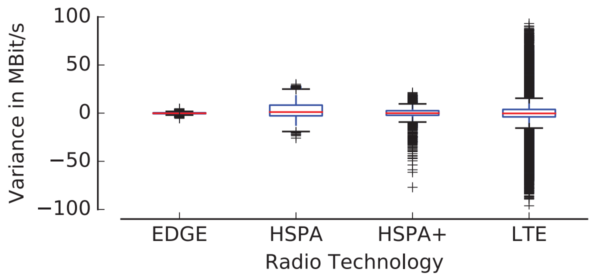


Fig. 9. Variance of `netradar` downlink measurements using different technologies with the overall median of ∼0.15 MBit/s.

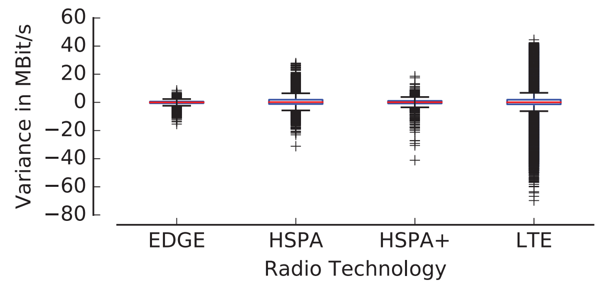


Fig. 11. Variance of `netradar` uplink with overall median of ∼0.07 MBit/s.

`netradar` **–** Fig. 8 shows the CDF of downlink measurements. As can be seen, the downlink of HSPA and HSPA+ seem to behave unexpected; HSPA seems to be faster than HSPA+. This is caused by the low number of measurements with those network technologies. The median downlink is ∼17 MBit/s. In about 95% of the measurements, the downlink speed is above 0.25 MBit/s and thus sufficient for Teleoperated Driving. The median variance is acceptable ∼0.15 MBit/s, but the maximum variance of ∼96 MBit/s is very high (Fig. 9).

Adequate uplink speed is required to provide qualitative sufficient video streams. The median uplink speed is ∼12 MBit/s (Fig. 10). In about 87% of the measurements the uplink speed is above 3 MBit/s and thus sufficient for Teleoperated Driving. While the median variance is acceptable ∼0.07 MBit/s, the maximum variance is ∼70 MBit/s and thus very high (Fig. 11). In general, these throughput results seem viable for Teleoperated Driving, but there is a high percentage of about 13%, where the uplink speed is insufficient. If only considering LTE connectivity, uplink and downlink is sufficient in about 95% of the time.

SierraWireless – The setup was connected to LTE in about 91.3% and to UMTS in 8.7% of the time. Compared to the `netradar` measurements, the connectivity is better, but with less samples. The median downlink speed is ∼28 MBit/s (Fig. 12). This is nearly double the value of the `netradar` results. Downlink is sufficient for Teleoperated Driving in more than 99% of the samples. The median variance in the downlink is ∼0.41 MBit/s, with a maximum variance of ∼43 MBit/s. The median uplink speed (Fig. 13) is ∼18 MBit/s, which is higher compared to the `netradar` results. The uplink speed is sufficient for Teleoperated Driving in about 98% of the measurements. The median variance in the uplink is ∼0.07 MBit/s, with a maximum variance of ∼26 MBit/s.

C. Measurements on Identical Routes

A comparison of `netradar` and `ping` measurements, where both were on the identical route were further examined. The median latency for `netradar` on this route is ∼55 ms, whereas `ping` shows ∼57 ms in RTT. The connectivity was about 97% LTE and 3% UMTS at the `ping` measurements and about 96% LTE and 4% EDGE during the `netradar` measurements. Results are roughly comparable with the same Hardware and different measurements techniques.

We further study whether the measurement platform makes a significant difference. The median downlink speed of `netradar` on the specific route is ∼15 MBit/s, whereas the downlink of *SierraWireless* is ∼32 MBit/s. The median uplink speed of the `netradar` measurements is ∼13 MBit/s, at the *SierraWireless* it is ∼20 MBit/s. Even with these measurements, there is a significant difference between both systems. The connectivity of `netradar` was about 91% LTE and 9% EDGE, whereas the connectivity of *SierraWireless* was 100% LTE on the identical route. Only considering the LTE connectivity of `netradar`, the uplink is ∼14 MBit/s and the downlink is ∼16 MBit/s. That is not a significant difference compared to the overall measurements on this route. Thus, the *SierraWireless* results are still better. This most likely is attributed to the setup with two antennas. Compared to the antenna inside a smartphone, they seem to provide better results. Thus, the use of better hardware, e.g. more antennas and better positioning of them, can improve feasibility of Teleoperated Driving.

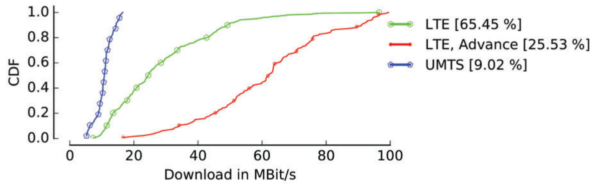


Fig. 12. CDF of *SierraWireless* downlink with overall median of ∼28 MBit/s (LTE: ∼24.5 MBit/s, LTE-A: ∼62.1 MBit/s, UMTS: ∼10.8 MBit/s).

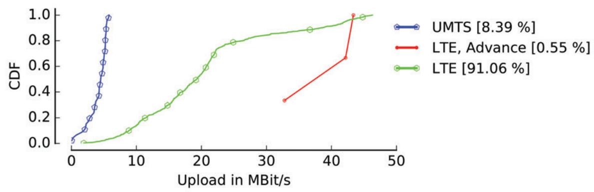


Fig. 13. CDF of *SierraWireless* uplink with overall median of ∼18 MBit/s (LTE: ∼19.2 MBit/s, LTE-A: ∼42.2 MBit/s, UMTS: ∼4.5 MBit/s).

D. Comparison of Different Scenarios

In addition to the comparison of different measurement setups at the identical routes, factors such as vehicle speed, handover, signal-strength, distance to the base station and their influence on the performance were further investigated.

Handover – We wanted to investigate whether handover has an influence on Teleoperated Driving. Both types of handover were investigated, e.g. switch with and without changing radio technology. The `netradar` measurements consist of only 12 switches in cell and 19 switches of radio technology. With *SierraWireless*, we witnessed 60 cell switches and no radio technology changes during the downlink measurements with 54 and one switch respectively with uplink measurements.

The overall median latency of the `netradar` measurements is ∼55 ms. When only switching cells, but keeping the same radio technology, there is no negative influence on the latency. When switching network technology during a measurement, latency increases by about 15%. In case of throughput, the median downlink is ∼17 MBit/s. With a cell-only switch, this value decreases to ∼14 MBit/s. In case of switching radio technology, the downlink drastically decreases to ∼3 MBit/s. The median uplink speed is ∼12 MBit/s. Keeping the same radio technology does not influence this value, but when changing the network technology, the uplink speed decreases to ∼3 MBit/s.

The median downlink speed for *SierraWireless* is ∼28 MBit/s. When performing a handover by keeping the same radio technology, the speed does not change. The median uplink speed is ∼18 MBit/s. When changing only the cell but keeping the radio technology, the uplink speed stays the same. In case of changes in the network technology, there is also no real difference, but the number of switches is only one. In general it can be said, that Teleoperated Driving is feasible when switching cells but keeping radio technology and infeasible when switching radio technology.

Speed – With Teleoperated Driving, the remote vehicle will have different speeds based on where it is driven. Considering the speed, we wanted to investigate whether different levels of speed have influence on the latency or throughput. We observe, for Teleoperated Driving, there is no real influence of the speed of the vehicle regarding latency or throughput. Even if there is a slightly higher performance at higher speeds with the *SierraWireless* measurements, this observation is only because speeds above 100 km/h were conducted on the highway, which provide a better network coverage with base-stations similar to observations made by previous studies [25].

Signal Strength – Due to the mobility of the remote vehicle, it can happen that the signal strength changes, e.g. in tunnels, and thus might have negative influence on Teleoperated Driving. In case of throughput (uplink/downlink), there is a clear tendency. The better the signal strength, the higher the throughput. For latency, no clear tendency is witnessed.

Distance – Finally, a vehicle when driving will change its distance to a base-station. We analyzed this distance based on the data from OpenCellID [2] to investigate whether there is an influence either on latency or the throughput. The distance to the base station was usually less than 5 km and no obvious influence of the distance on the values for throughput (uplink/downlink) and latency were observed.

E. Whitelisting as Possible Mitigation

In general, latency, jitter and throughput values are promising for Teleoperated Driving. However, our measurements are limited and only reflect the network states of the routes taken. We believe this to be acceptable for a first assessment, but not exhaustive for real world use cases. Therefore, we propose the approach of whitelisting with frequent probing. This whitelisting is a simple approach to mitigate critical situations by allowing Teleoperated Driving only in areas that are measured to provide sufficient network performance. To examine, whether Teleoperated Driving would be feasible in areas with good network connectivity, a typical whitelisting scenario, we drove in an area with LTE or LTE-Advanced coverage only. We chose a time in the afternoon at which a lot of commuters are driving, to get measurements with a high number of other road users. The driven route is a 5 km long circle around the historic center of Ingolstadt. We drove on it four times, leading to a total of about 20 km (60 minutes). The driving activity was split up into two parts. The first part consisted of two rounds for measuring over TCP. The second part consisted of two rounds for measuring over UDP, resulting in about 10 km and ∼30 minutes for each part. The standard Linux `ping` utility on the car PC was executed all the time. Both parts were driven consecutively without breaks in between. The measurements were conducted with the test vehicle and the aforementioned *SierraWireless* setup. In contrast to the previous measurements, a maximum bandwidth of 5 MBit/s was specified to be used. Thus, we are able to explore how stable values are. We chose 5 MBit/s to have the minimum required 3 MBit/s for uplink plus additional 2 MBit/s as margin. Measurements on the first part

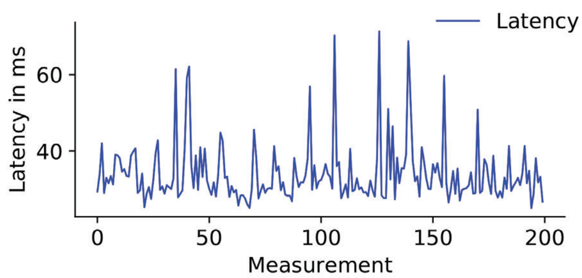
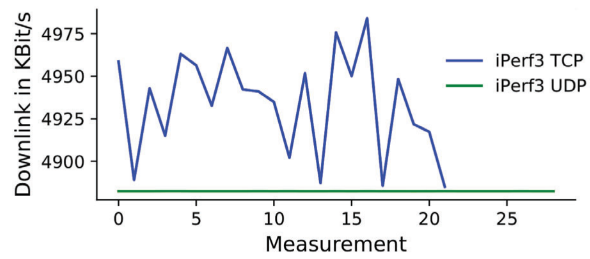
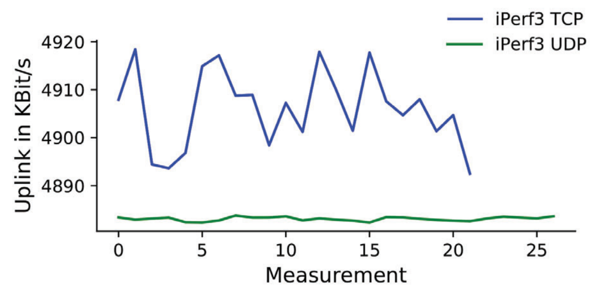


Fig. 14. Latency measured during the test drives with median of ∼31 ms.



(a) Downlink



(b) Uplink

Fig. 15. Downlink (median ∼4.94 MBit/s for TCP, median ∼4.88 MBit/s for UDP) and uplink (median ∼4.90 MBit/s for TCP, 4.88 MBit/s for UDP) measured during the test drives.

were conducted continuously with the sequence: (1) `ping`, `iperf3` download (TCP), `ping` (2) `ping`, `iperf3` upload (TCP), `ping`. TCP was replaced by UDP in the second part of the activity. Due to the fact that `ping` is measured more frequently, we have a higher number of `ping` measurements than that of TCP (or UDP).

The measured latency (Fig. 14) has a median of ∼31 ms, whereas the maximum is 71 ms. To investigate whether the `ping` values can be used, we also measured the RTT during the TCP measurements. The median value of ∼27 ms is comparable, but the maximum of 45 ms is lower. Hence, the 71 ms sample point seems to be an outlier. During our test drive in the white listed area, there is no measurement in which the latency is above the critical value of 250 ms, which means that Teleoperated Driving is possible. Jitter is always below 150 ms and thus no issue there either.

We also measured the uplink and downlink speed with TCP and UDP. Fig. 15a shows measured downlink over time. The median downlink speed is ∼4.94 MBit/s (TCP) and ∼4.88 MBit/s (UDP), with minimum values of 4.88 MBit/s (TCP) and 4.88 MBit/s (UDP). This is above the minimal required value of 0.25 MBit/s. Moreover, the variance of values is not critical for Teleoperated Driving. Fig. 15b shows the the uplink speed. The median uplink speed is ∼4.90 MBit/s (TCP) and ∼4.88 MBit/s (UDP), whereas the minimum values are 4.89 Mbit/s (TCP) and 4.88 MBit/s (UDP). These values are above the required minimum of 3 MBit/s. The fluctuation of the uplink speed leaves Teleoperated Driving feasible. For upload and download, the values never reach the specified 5 MBit/s, but that is based on the measurement method of `iperf3`. During our measurements 16 handovers without and 5 with network technology changes occurred. However, it can be seen that there is no real influence on the performance. The packet loss during the UDP measurements is in the order of 10^{-4}% and thus less an issue for Teleoperated Driving. In general it can be said that a whitelisting-based approach could work. Our results confirm the applicability of whitelisting [17] for Teleoperated Driving. We are aware that our measurements provide a snapshot of the network state, altough we attempt to select a route and a time with a real amount of traffic. The whitelisting approach has to be advanced further by permanently probing areas, e.g. normal vehicles conducting periodic network performance measurements and by sending them to a cloud service. The cloud service can be advanced with data, following the predictability of connected cars [7]. Using this cloud-service, teleoperated vehicles may plan and update their route based on the incoming data. If situations are changing dynamically, adjustments can be applied early before a remote vehicle enters a dangerous area. With the increased number of measurements, a more accurate map can be built and Teleoperated Driving might be possible. A complementary blacklisting can also follow, so that areas where Teleoperated Driving will not be possible are blocked. Although using such a navigation-based approach can help, network parameters can become critical even within whitelisted areas. For such situations, other approaches, e.g. based on the vehicles' monitoring system [26] have to be applied.

VI. Limitations and Future Directions

The results presented in this paper are limited by the amount and type of measurements we conducted. Changes in network conditions are always likely to occur and will influence the measurements we have conducted. The results reflect end-user's perspective and do not consider how providers treat different packets. We treat the network connection as black box, e.g we are thus not able to obtain information on how busy particular cells were. The whitelisting approach is only presented in principle and does not yet consider changes in network conditions. Nevertheless, these results can be used to get a first impression whether Teleoperated Driving could be feasible with contemporary mobile networks.

In the future, a user study with more participants will be conducted, to investigate whether skilled and trained drivers are able to cope with network conditions of high delay. The cloud-based navigation system will be improved and tested, to see, whether the approach is feasible in everyday scenarios.

VII. Conclusion

We observed that Teleoperated Driving may be feasible with contemporary mobile networks, especially with LTE and LTE-Advanced capability in whitelisted areas. In most cases (`ping` and `netradar`: 96%), the latency was below 250 ms, the uplink was above 3 MBit/s (*SierraWireless*: 98%, `netradar`: 87%) and the downlink above 0.25 MBit/s (*SierraWireless*: 99%, `netradar`: 95%). This indicates that in the majority of our measurements, Teleoperated Driving could be used. With the test drive on a specific route, we showed that the whitelisting approach can work, but the basic approach has to be improved considering changes in network conditions to provide accurate maps for Teleoperated Driving. Frequent probing and providing results to a cloud-based service might help. Teleoperated vehicles are then able to dynamically adapt their route with changes in network conditions. There are also cases in which Teleoperated Driving does not work. Latencies above 1 second, high jitter and low throughput are not tolerable and have to be avoided. Further, handover between two cells can negatively influence Teleoperated Driving, as it might lower the throughput and increases the latency. The signal strength has influence on the throughput but not on the latency. The better the signal strength, the higher the throughput. The position of the teleoperation station is also crucial. Fluctuation of the latency increases if the remote vehicle is further away from the teloperation station. Improved hardware and multiple antennas can help with the connectivity and thus also increase the usability of Teleoperated Driving. In general it can be said that at least within our measurements, Teleoperated Driving can be feasible with contemporary mobile networks, but more measurements and clever approaches are required.

References

[1] Lenovo B smartphone. https://lnv.gy/2VvFZ50. Accessed: 27.09.2018.

[2] OpenCellID. https://www.opencellid.org. Accessed: 15.05.2019.

[3] OpenROUTS3D. https://git.io/fj8tn. Accessed: 16.05.2019.

[4] ping utility for Android. https://bit.ly/2Hr4Tzi, 2019.

[5] ADAC. Mobility in Germany - Selected Results (org: Mobilitaet in Deutschland - Ausgewaehlte Ergebnisse). https://bit.ly/2w0qRCD, 2010.

[6] Adobe Developer Connection. Recommended bit rates for live streaming. https://adobe.ly/2wd8pH7. Accessed on 11.10.2018.

[7] Carlos E. Andrade, Simon D. Byers, Vijay Gopalakrishnan, Emir Halepovic, David J. Poole, Lien K. Tran, and Christopher T. Volinsky. Connected Cars in Cellular Network: A Measurement Study. In *Internet Measurement Conference*, 2017.

[8] Vaibhav Bajpai and Jürgen Schönwälder. A Survey on Internet Performance Measurement Platforms and Related Standardization Efforts. *IEEE Communications Surveys and Tutorials*, 17(3):1313–1341, 2015.

[9] Frederic Chucholowski, Tito Tang, and Markus Lienkamp. Teleoperated Driving Robust and Secure Data Connections. *ATZelektronik worldwide*, 9(1):42–45, Feb 2014.

[10] Frederic Emanuel Chucholowski. Evaluation of Display Methods for Teleoperation of Road Vehicles. *Journal of Unmanned System Technology*, 3(3):80–85, 2016.

[11] ERTRAC Working Group ”Connectivity and Automated Driving”. Automated Driving Roadmap. resreport, ERTRAC, July 2017.

[12] J. Davis, C. Smyth, and K. McDowell. The Effects of Time Lag on Driving Performance and a Possible Mitigation. *IEEE Transactions on Robotics*, 26(3):590–593, June 2010.

[13] Matthias Dick, Oliver Wellnitz, and Lars Wolf. Analysis of Factors Affecting Players' Performance and Perception in Multiplayer Games. In *ACM SIGCOMM Workshop on Network and System Support for Games*, NetGames '05, pages 1–7, New York, NY, USA, 2005. ACM.

[14] P. M. d'Orey, A. Hosseini, J. Azevedo, F. Diermeyer, M. Ferreira, and M. Lienkamp. Hail-a-Drone: Enabling teleoperated taxi fleets. In *2016 IEEE Intelligent Vehicles Symposium (IV)*, pages 774–781, June 2016.

[15] Sebastian Gnatzig, Frederic Chucholowski, Tito Tang, and Markus Lienkamp. A System Design for Teleoperated Road Vehicles. In *Proceedings of the 10th International Conference on Informatics in Control, Automation and Robotics*, pages 231–238, July 2013.

[16] Hang Guo and John Heidemann. Detecting ICMP Rate Limiting in the Internet. In Robert Beverly, Georgios Smaragdakis, and Anja Feldmann, editors, *Passive and Active Measurement*, pages 3–17, Cham, 2018. Springer International Publishing.

[17] Mark Harris. CES 2018: Phantom Auto Demonstrates First Remote-Controlled Car on Public Roads. https://bit.ly/2D26Z87, 2018. Accessed:28.11.2018.

[18] A. Hosseini and M. Lienkamp. Predictive safety based on track-before-detect for teleoperated driving through communication time delay. In *2016 IEEE Intelligent Vehicles Symposium (IV)*, June 2016.

[19] Keon Jang, Mongnam Han, Soohyun Cho, Hyung-Keun Ryu, Jaehwa Lee, Yeongseok Lee, and Sue B. Moon. 3G and 3.5G Wireless Network Performance Measured from Moving Cars and High-speed Trains. In *ACM Workshop on Mobile Internet Through Cellular Networks*, 2009.

[20] Lei Kang, Wei Zhao, Bozhao Qi, and Suman Banerjee. Augmenting self-driving with remote control: Challenges and directions. In *Workshop on Mobile Computing Systems & Applications*, 2018.

[21] M. Lauridsen, L. C. Gimenez, I. Rodriguez, T. B. Sorensen, and P. Mogensen. From LTE to 5G for Connected Mobility. *IEEE Communications Magazine*, 55(3):156–162, March 2017.

[22] Feng Li, Jae Won Chung, Xiaoxiao Jiang, and Mark Claypool. TCP CUBIC versus BBR on the Highway. In *Passive and Active Measurement Conference*. Springer International Publishing, 2018.

[23] Ruilin Liu, Daehan Kwak, Srinivas Devarakonda, Kostas Bekris, and Liviu Iftode. Investigating Remote Driving over the LTE Network. In *Conference on Automotive User Interfaces and Interactive Vehicular Applications*, AutomotiveUI '17, 2017.

[24] Anna Maria Mandalari, Andra Lutu, Ana Custura, Ali Safari Khatouni, Özgü Alay, Marcelo Bagnulo, Vaibhav Bajpai, Anna Brunström, Jörg Ott, Marco Mellia, and Gorry Fairhurst. Experience: Implications of roaming in europe. In *MobiCom 2018*, pages 179–189, 2018.

[25] Ruben Merz, Daniel Wenger, Damiano Scanferla, and Stefan Mauron. Performance of LTE in a High-velocity Environment: A Measurement Study. In *Workshop on All Things Cellular: Operations, Applications, & Challenges*, AllThingsCellular '14, 2014.

[26] Stefan Neumeier, Nicolas Gay, Clemens Dannheim, and Christian Facchi. On the Way to Autonomous Vehicles - Teleoperated Driving. In *AmE 2018 - Automotive meets Electronics*, 2018.

[27] A. Parichehreh, S. Alfredsson, and A. Brunstrom. Measurement Analysis of TCP Congestion Control Algorithms in LTE Uplink. In *Traffic Measurement and Analysis Conference*, pages 1–8, June 2018.

[28] SierraWireless. AirLink RV50 Datasheet. https://bit.ly/2W7dkrj. Accessed on 27.09.2018.

[29] Tito Tang, Frederic Chucholowski, and Markus Lienkamp. Teleoperated driving basics and system design. *ATZ worldwide*, 116(2), Feb 2014.

[30] Tito Tang, Pascal Vetter, Markus Lienkamp, Simon Finkl, and Korbinian Figel. Teleoperated Road Vehicles The ”Free Corridor” as a Safety Strategy Approach. In *Mechanical Design and Power Engineering*. Trans Tech Publications, 2014.

[31] Ermias Andargie Walelgne, Setälä Kim, Vaibhav Bajpai, Stefan Neumeier, Jukka Manner, and Jörg Ott. Factors affecting performance of web flows in cellular networks. In *IFIP Networking Conference*, 2018.

[32] Ermias Andargie Walelgne, Jukka Manner, Vaibhav Bajpai, and Jörg Ott. Analyzing Throughput and Stability in Cellular Networks. In *IEEE/IFIP Network Operations and Management Symposium*, pages 1–9, 2018.

[33] Alan Ft Winfield. Future Directions in Tele-operated Robotics. In *Telerobotic Applications*. 2000.

[34] Q. Xiao, K. Xu, D. Wang, L. Li, and Y. Zhong. TCP Performance over Mobile Networks in High-Speed Mobility Scenarios. In *Conference on Network Protocols*, pages 281–286, Oct 2014.

[35] YouTube Help. Live encoder settings, bitrates and resolutions. https://bit.ly/2oob5w0. Accessed: 11.10.2018.