Implementing Remote Driving in 5G Standalone Campus Networks

Michael Klöppel-Gersdorf[1][a], Adrien Bellanger[1], Tobias Füldner[2], Dirk Stachorra[2], Thomas Otto[1] and Gerhard Fettweis[2]

[1] *Fraunhofer IVI, Fraunhofer Institute for Transportation and Infrastructure Systems, Dresden, Germany*

[2] *Vodafone Chair, Mobile Communications Systems, TU Dresden, Dresden, Germany*

Keywords: Remote Driving, Teleoperation, 5G, Campus Networks, Yard Automation.

Abstract: While there have been enormous advances in automated driving functions in the recent years, there are still circumstances where automated driving is not feasible or not even desired. Teleoperation is one approach to keep the vehicle mobile in such situations, with remote driving being one mode of teleoperation. In this paper we describe a 5G remote driving environment based on a 5G Standalone campus network, explaining technological and hardware choices. The paper is completed with experiences from practical trials, showing that remote driving using the proposed environment is feasible on a closed area. The achieved velocities are similar to that of a direct human driver.

1 INTRODUCTION

Vehicle automation has seen huge advances in the recent years and the number of Operational Design Domains (ODDs) where autonomous driving is possible increases. Still, there are situations, which do not allow autonomous driving. While the vehicle can still obtain a risk-minimal state, teleoperation in the form of remote driving might allow the possibly uninhabited vehicle to continue its journey. Further applications of remote driving might be to allow individual mobility for people with disabilities (Domingo, 2021) or yard automation, where traditional drivers can hand over their trucks at the gate. Afterwards, the truck will be remotely driven to parking positions. This is, for instance, of interest in regions undergoing structural changes like the Lausitz region in eastern Germany, where truck drivers might be hard to find and employ. Freeing the drivers from parking tasks allows them to complete more tours.

1.1 Remote Driving

While teleoperation is regularly used in reconnaissance and disaster recovery as well as in Unmanned Aerial Vehicles (UAV), commercial applications in the vehicle driving domain are still scarce. To the best knowledge of the authors, only one company[1] in Europe is currently undertaking remote driving studies with uninhabited vehicles on public roads.

Research into remote driving is ongoing for more than ten years, with a first demonstration using a 3G public network to remotely control a vehicle going back to 2013 (Gnatzig et al., 2013). While this proved the feasibility of remote driving, latencies of more than 1 s were too high for commercial application. This also highlights the main bottle necks in the widespread deployment of the technique, namely, latency and bandwidth requirements. Although there was hope for the next generation of mobile networks, in the form of Long Term Evolution (LTE), even these proved not to be enough (Liu et al., 2017). The results for the current 5G technology look more promising, with first show cases of an end-to-end remote driving solution provided on public roads (Kakkavas et al., 2022) already presented. Other authors agree that 5G remote driving is possible, at least at sites with excellent coverage by 5G base stations (Saeed et al., 2019; Kim et al., 2022) and careful positioning of the remote operators at key locations of the network (Zulqarnain and Lee, 2021).

In this paper, we examine remote driving using 5G Standalone Campus networks on restricted areas, like yards. Our key performance indicators are an end-to-end latency in the video transmission lower than

[a] https://orcid.org/0000-0001-9382-3062

[1] https://vay.io

Klöppel-Gersdorf, M., Bellanger, A., Füldner, T., Stachorra, D., Otto, T. and Fettweis, G.
Implementing Remote Driving in 5G Standalone Campus Networks.
DOI: 10.5220/0011985900003479
In *Proceedings of the 9th International Conference on Vehicle Technology and Intelligent Transport Systems (VEHITS 2023)*, pages 359-366
ISBN: 978-989-758-652-1; ISSN: 2184-495X

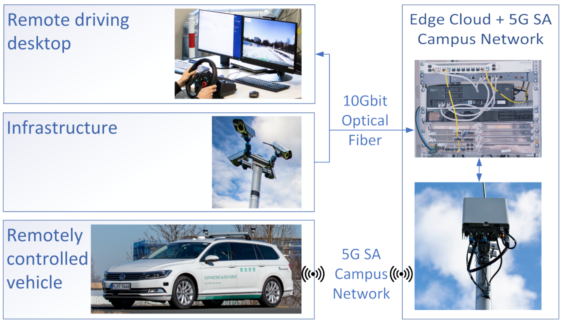

Figure 1: Main components of the remote driving demonstrator.

300 ms, reliable transmission of control signals, and achieved velocities similar to a normal human driver. In contrast to the commercial implementation mentioned above, we use a more simple sensor setup with only one camera, and we do not implement measures to handle connection loss due to the presence of a safety driver. Furthermore, we are using a 5G Standalone Campus network, i.e., we do not have to share bandwidth with other users and avoid having to use a multi-provider approach to guarantee connectivity. Implementation details are given to enable other researchers to build their own demonstrators.

The paper is organized as follows: The next chapter introduces all the components required to perform the remote driving task, whereas Section 3 presents the obtained results, which are discussed in the following section. The paper concludes with an outlook in the last section.

2 COMPONENTS

Fig. 1 shows an overview over all components of the demonstrator as well as the interconnection between all parts.

2.1 Remotely Controlled Vehicle

The test vehicle is a VW Passat, which was equipped for automated driving by IAV GmbH. The vehicle has an automatic transmission, freeing the remote driver from having to shift gears remotely. Steering is accessed using the servo motors employed by the parking and lane assistant. While the motors are only certified to carry out larger steering wheel motions at velocities up to 10 km/h, they can actually be operated at speeds up to 130 km/h. Both acceleration and steering wheel angle can be controlled using a custom Controller Area Network (CAN) interface. For video capture a single Logitech StreamCam with a resolution of 1280x720 pixels is used. It is mounted below the rear-view mirror. The opening angle of 78° provides sufficient information during forward motion. The CAN bus is accessed via a USB CAN bus interface, which is connected to the car computer, a NUVO 7160 GC. This computer features an Intel i7-8700 processor, 32GB of RAM and a NVIDIA 1080 graphics card. The latter was not utilized in the described implementation.

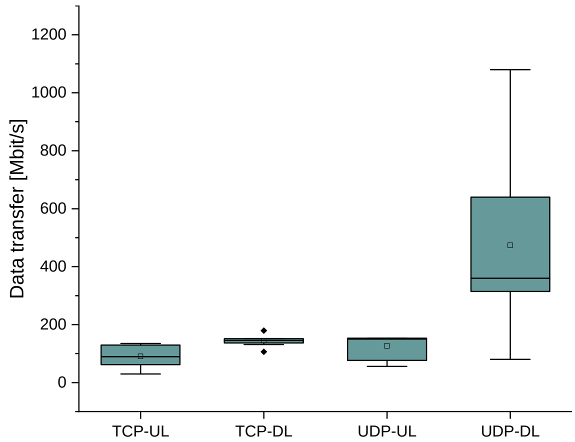

Figure 2: Box plot showing up- and download (labeled UL and DL, respectively) of eight commercially available 5G modems/routers using UDP and TCP.

For 5G connectivity, eight different commercially available routers and modems were evaluated. Tests were carried out at 50 m distance to a base station under line-of-sight conditions. As shown by Fig. 2 there is a certain spread in the capabilities of the different routers, especially with regards to UDP data transfer. This is of interest since the implemented solution requires high data rates for UDP upload. Although the transmission of control variables relies on TCP, this is not relevant due to the low volume of control signals being sent, i.e., only two float values are transmitted (for acceleration and steering wheel angle, respectively). Finally, we decided on a Mikrotik Chateau 5G. While this device is marketed as office router, it proved more than capable for the task, actually scoring first place in our comparison at UDP upload (153 Mbit/s), while being the cheapest option.

2.2 5G Edge Cloud

Edge clouds were introduced to move computation power nearer to the mobile devices, thereby saving bandwidth and allowing faster responses in comparison to traditional cloud computing (Cao et al., 2020). Our edge cloud is directly connected to the 5G radio and also features high bandwidth connections to both infrastructure as well as remote driving desktop

Table 1: Overview over the computation resources used in the edge cloud.

Virtual machine	#Cores	RAM	Graphics card
MQTT	1	4 GB	
Video processing	12	32 GB	X
TURN/STUN	12	48 GB	

(compare Fig. 1). It consists of a server system comprising two Intel Xeon Gold 6338 with 32 physical cores each, clocked at 2.0 GHz. Each CPU is complemented by 128 GB RAM. Additionally, the system contains one NVIDIA RTX A6000 for video processing. The required software components (Message Queuing Telemetry Transport (MQTT) broker, video processing, TURN/STUN server) are deployed in virtual machines, an overview of which can be found in Table 1. The TURN/STUN server is required to enable the video connection. Additionally, it also provides a port forwarding to access the car computer. Please note that the computation resources are not completely used, allowing for additional tasks for the edge cloud. Especially the TURN/STUN server is massively over-provisioned, allowing for other non-project related tasks to be executed without interfering with remote driving. Furthermore, even in the presented solution most computational resources are reserved for the optional infrastructure monitoring (see Section 2.7).

2.3 5G Standalone (SA) Campus Network

We use a 5G SA (Standalone) campus network (Rischke et al., 2021), which employs a specific carrier frequency between 3.7 MHz – 3.8 MHz, outside of the public 5G networks. It is provided by a Nokia Digital Automation Cloud (NDAC) with 3GPP Release 15 (Dahlman and Parkvall, 2018) support, which consists of a 5G SA edge core server for local user and control plane with additional network management functions for SIM configuration and high level monitoring, and a 5G NR Radio Access Network (RAN), a distributed solution with Nokia Airscale Baseband Units and 2x2 AWHQF remote radio heads.

The installation uses a single sector installed at 4.5 m height. Even at this height, it is able to cover the whole test track, even providing sufficient signal at 200 m distance and non-line-of-sight conditions. The same hardware was able to achieve coverage up to 1 km under line-of-sight conditions and when installed at 10 m height. In both cases, the transmission power were the maximum permitted 10 W.

Figure 3: Screen shot of the remote driving desktop, showing the video stream from the vehicle as well as current velocity and the status of the acceleration (AC) and steering (ST) interfaces. The gray color indicates deactivated interfaces, i.e., no remote driving is possible at the moment.

2.4 Remote Driving Desktop

The remote driving desktop consists of a Lenovo all-in-one computer, providing an Intel i7-8700 CPU, 8 GB of RAM and a 23.8 inch display. A Logitech G29 racing wheel (including pedals) is used for realistic driving. This wheel is also able to provide force feedback and automatic centering.

2.5 Software

The software stack in the given implementation has to solve two tasks. First, sensor information has to be transmitted from the vehicle to remote driving desktop and second, control inputs have to be relayed to the vehicle. For the first task, it is necessary to determine which sensor information is actually required to perform the remote driving task. According to Nash et al. (Nash et al., 2016), the main sensory inputs used by drivers in the control of vehicle speed and direction are visual, vestibular and somatosensoric information. In the implemented solution, we concentrate on the visual sense by providing appropriate video information. The other two sources of information are far more difficult to duplicate since these would require a far more sophisticated workplace for the remote driver and are not considered here. Instead, additional speed and aural information is provided, as this might be necessary to evaluate certain driving scenes involving other participants (e.g., shouting or honking) as well as providing feedback on the vehicle (e.g., motor sounds or the sound of slipping tires). Hence, the task at hand is to provide video and audio from the vehicle as fast as possible. In the end, we decided to use WebRTC (Sredojev et al., 2015), which while not being optimal (Sato et al., 2022) is easy

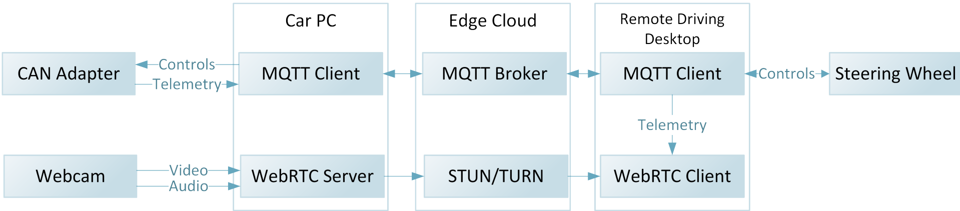

Figure 4: Data flow in the implemented solution. On the remote driving desktop, telemetry data (velocity, system state) is injected into the video stream (see Fig. 3), explaining the need to connect the WebRTC client directly to MQTT.

to deploy and works out-of-the-box[2] for video and audio. Other possible solutions are custom streams generated by ffmpeg or gstreamer, but these require a careful choice of parameters. We also considered a commercially available protocol, but did not consider it further as it required all participants to use the same local network. Although this could be achieved in our setup, it is generally not achievable in praxis. In prior tests using WiFi both WebRTC and the commercially available protocol achieved Glass-to-Glass latency (also called end-to-end latency) below 100 ms, whereas with a custom ffmpeg stream only latencies in the 800 ms range were achieved. In our case, the WebRTC server was deployed on the vehicle computer. Normally, the server would have been deployed in the edge cloud, but this would then require a secure connection to access the vehicles camera. This is a restriction imposed by all modern browsers to avoid loss of private video data, the only exception being connections to the local machine. Since remote driving desktop and vehicle are in different network segments and cannot access each other directly, a TURN/STUN server, which uses coturn, is implemented in the edge cloud to allow the WebRTC connection. The implemented solution runs inside a web browser and provides a custom website including the video stream and additional status information as seen in Fig. 3.

MQTT (ISO/IEC 20922:2016, 2016) is used to transport control information to the vehicle, using two MQTT topics for acceleration and steering wheel angle, respectively. These values are obtained from the Logitech steering wheel using PyGame 2[3]. An additional topic is used to relay the readiness of the remote driver (see Section 2.8 below). In the vehicle, a Python software module translates the values received to appropriate values for the custom CAN interface. MQTT is an obvious choice, since it has wide traction in Internet of Things (IoT) projects and is easy to deploy. An overview over the complete data flow in the deployed solution is shown in Fig. 4.

[2]https://github.com/TannerGabriel/WebRTC-Video-Broadcast

[3]https://github.com/pygame/pygame/releases/tag/2.0.0

Figure 5: Test track used within the project. The locations EC and R mark the position of the edge cloud server and the radio, respectively.

2.6 Test Track

The test track used can be described as pretzel (see Fig. 5), with dimensions 90 m × 50 m. The green areas inside the track are actually two hills of approximately 1.5 m height. This complicates remote driving tasks, as it is not possible to look over these areas. The road inside the curves has a width of 3.5 m, corresponding to the standard width of a lane in public traffic in Germany. The container containing the edge cloud server and the post carrying the 5G radio are located at the eastern side of the test track.

2.7 Infrastructure

The complete test track is covered by eight cameras, located at two masts on top of the hills of the test track. Image recognition algorithms carried out in edge cloud detect objects moving on the track and have the possibility of generating warnings in case of impending crashes (Klöppel-Gersdorf and Otto, 2022). Using the provided NVIDIA RTX A6000 allows to update objection positions with up to 12 Hz, i.e., video processing adds a latency of about 80 ms. Given the speed limit on the test track, this means that even in the worst case the position of the object reported is less than a meter away from the current position.

A detailed description of the video surveillance system can be found in (Klöppel-Gersdorf et al., 2021).

2.8 Safety Architecture

The safety architecture of our vehicles requires a specially trained safety driver to be always present, especially in teleoperation mode. The systems are designed such that the safety driver can override any steering or acceleration command being sent by the remote driving desktop. In case of disturbances, the interfaces can be completely disabled, i.e., only the safety driver can operate the vehicle. Since a trained driver is always present, no measures for handling connection loss were implemented. Furthermore, we implemented a custom protocol for activating the remote driving functionality: First the safety driver has to enable the vehicle interface, which will be relayed to the remote driver. The remote driver then indicates the readiness to carry out the remote driving task by setting the interfaces in the ready state by pressing a special button on the gaming steering wheel. Finally, the safety driver has to confirm by setting the interfaces to active. Both, the safety and the remote driver, can abort the remote driving functionality at any time. In this case, the vehicle returns to standard operation mode. In addition, a loud acoustical signal is provided.

If desired, objects recognized by the infrastructure can be visualized in a top-down view providing the remote driver with additional information of the scene, even of objects not captured by the vehicle's camera. This can be useful to increase the remote driver's telepresence, i.e., their feeling of actually being inside the remote situation.

3 RESULTS

We conducted several test drives on our test track under various environmental conditions, including sunny days as well as roads covered by ice. Remote driving was successful under all this conditions, where success is defined by completing several laps without leaving the track. As the width of the lane on the test track coincides with the width of public lanes, this indicates that our solution would also be suitable to drive on public roads. Maximum velocities achieved where 28 km/h under dry conditions and 15 km/h when driving on ice. The speed under dry conditions is similar to what a driver directly driving the vehicle could achieve due to the radius of the curves on the test track. While higher velocities

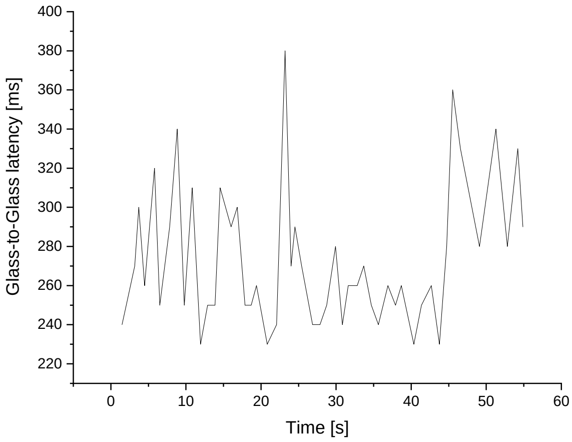

Figure 6: Glass-to-Glass latency for one minute. Minimum, maximum and average latency were 230 ms, 380 ms and 272.7 ms, respectively, with a standard deviation of 35.55 ms.

would have been possible when driving on ice, we had to deal with the fact that the vehicle automatically canceled all required interfaces when activating Anti-lock Braking System (ABS), which happened frequently at higher velocities under such conditions. Other difficulties included driving in the direction of the sun, as this turned the camera essentially blind. This could be circumvented by using a more suitable camera model, which is able to adapt light sensibility faster.

As also remarked by other researchers (compare for instance (Tener and Lanir, 2022)), estimating the vehicle's velocity proved difficult for the remote driver. Therefore, a direct measurement was included in the video stream (see Fig. 3). Furthermore, we also found that owning a suitable drivers license and even regular driving experience are not enough to act as remote driver. To the contrary, our remote driver needed intensive schooling over several weeks while slowly increasing the velocity as well as direct drives in the remotely controlled vehicle to get acquainted with the peculiarities of the vehicle.

Fig. 7 shows the round-trip-time from edge cloud and vehicle to the remote driving desktop while carrying out the driving task. While the time from edge cloud to remote driving desktop is negligible (due to the usage of 10Gbit fibre network), communication over 5G adds some measurable latency, with a mean latency of 17.6 ms. Comparing this with the results in (Gnatzig et al., 2013), latency in 5G networks is down to 10% of the latency observed in 3G networks. As described above, two MQTT topics containing a single float value are used to transmit the control information to the vehicle, i.e., the payload is much smaller than the Maximum Transfer Unit (MTU) of the con-

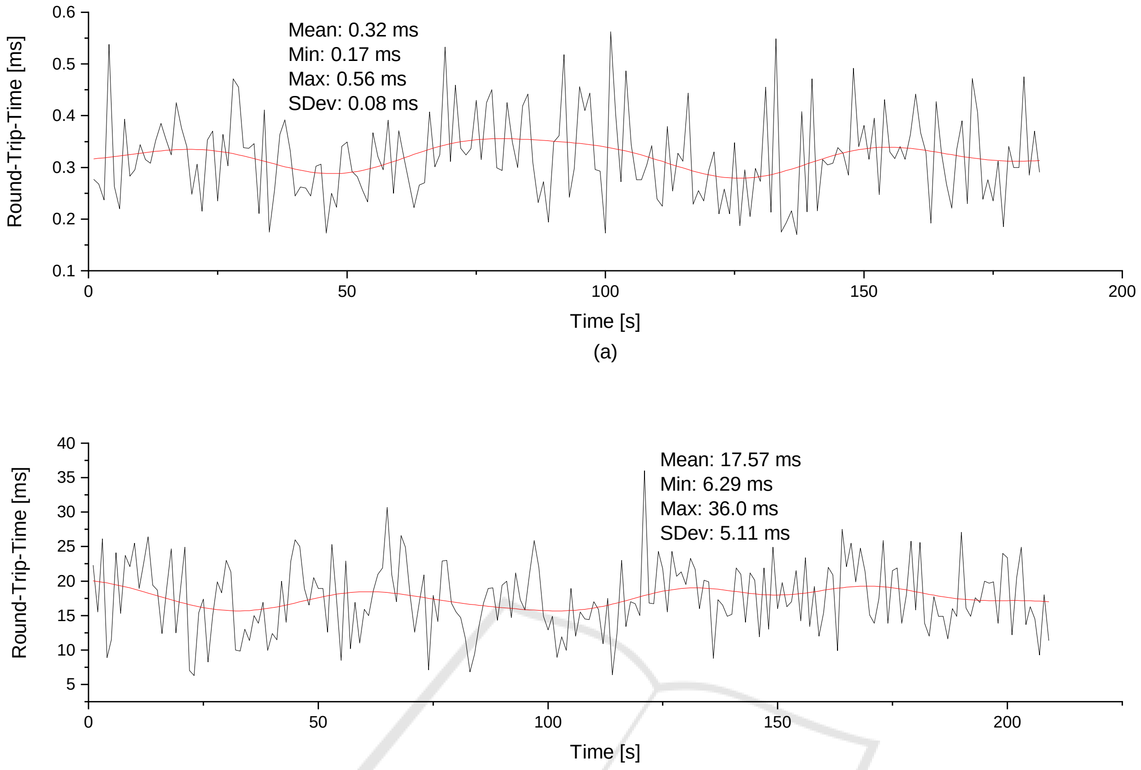

Figure 7: Round-Trip-Time from edge cloud to remote driving workplace (a) and from vehicle to remote driving workplace (b) while carrying out the remote driving task. Red lines are the result of low-pass filtering.

nection and every control value is encapsulated in a single network packet. We did not observe any packet loss during the experiments. Transmission time was similar to the round-trip-time reported above, i.e., about 20 ms. Besides network latency there is also the question of Glass-to-Glass latency of the whole video system, which also includes pre-processing in the camera as well as encoding and decoding an h.264 video stream. The corresponding measurements are shown in Fig. 6. The numbers achieved confirm the simulated results in (Sato et al., 2022), but contradict our initial measurements using WiFi. We later confirmed that part of this discrepancy can be explained by using older hardware in the actual demonstration than during the initial tests. Tests using a modern notebook for displaying the WebRTC stream reduced Glass-to-Glass latency to values just below 200 ms. Still, even at the present state, we are able to stay below the 300 ms given in (Neumeier et al., 2019) for comfortable remote driving. Also, the relatively low variance in latency means we do not need to add artificial latency to smooth out the latency distribution as employed in (Gnatzig et al., 2013; Liu et al., 2017).

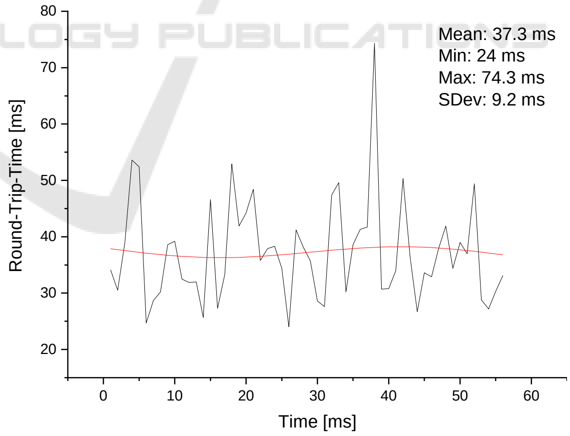

Figure 8: Round-trip-time from edge cloud to a server in the building, where the remote driving desktop is located, using a public 5G connection.

4 ARE WE THERE, YET?

The answer to this question very much depends on the use case. For applications on restricted areas, like yards or parking decks, which can be completely covered by 5G antennas, the answer is a definitive yes,

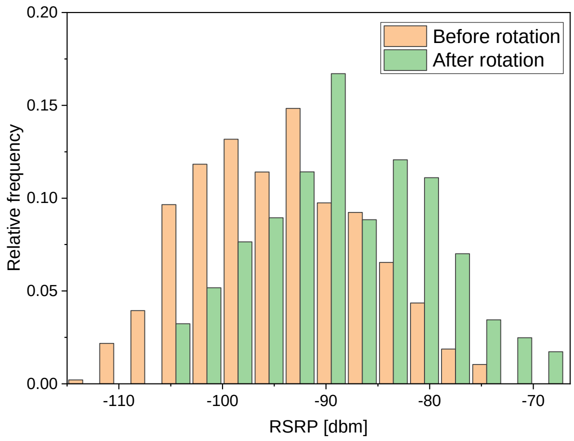

Figure 9: Histogram showing the distribution of RSRP on our testing grounds before and after adjusting the radio. Measurements were taken using a Rhode and Schwarz QualiPoc. Adjusting the radio significantly improved the reception.

especially if one is able to deploy a private 5G campus network and guarantee mostly Line-of-Sight (LOS) connections. While we also had success under certain Non-Line-of-Sight (NLOS) conditions, this very much depends on the geometry of the premise and would at least require a careful positioning of the radio heads. On the other hand, regarding applications in public traffic, the answer is still no, due to data rates and complete coverage by 5G radio required. This is evident from Fig. 9, where even a slight readjustment of the antenna led to improved reception. For public traffic, this means, that either a large number of radios is required to allow the service during the total duration of the trip or outages of the service are to be expected, which is also confirmed by the theoretical considerations in (Saeed et al., 2019). More generally, as den Ouden et al. (den Ouden et al., 2022) pointed out, a certain level of robustness of the network must be guaranteed to safely carry out the remote driving task.

In addition to the 5G SA campus network, the test track is also equipped with a router accessing the public 5G network. We choose this connection to get an estimate of the round-trip-time in such a setup, i.e., how much latency would differ if the connection to the vehicle was routed over the public network instead of using the campus network. As endpoint, a server in the same building as the remote driving desktop was chosen. The results are shown in Fig. 8. While the values are higher than in the campus network, the difference should not matter much in practical implementation at least with regards to the video transmission. On the other hand, doubling the latency in comparison to the 5G campus network could have some detrimental impact on the transmission of control values to the vehicle, but this is still a question of research. Nevertheless, one has to keep in mind that the 5G router was immobile at the time of measurement and in close vicinity to a 5G base station, actual results while driving would certainly be worse.

The current implementation uses only a single camera with 78° opening angle. According to EU directives (Economic Commission for Europe of the United Nations, 2010), the horizontal field of view should at least be 180°. When using the current camera model, at least three cameras would be required to achieve this requirement. Incidentally, the currently used resolution of 720p is exactly three times the number of pixels of three camera streams with 640×480 pixels, i.e., with a slight decrease in resolution also three cameras could be supported by our solution. Furthermore, it would be possible to apply selective downsampling to further decrease the required bandwidth (Dehshalie et al., 2022) at the cost of increased latency due to processing. Alternatively, it would also be possible to employ cameras with included encoding capabilities as this would lift the bottle neck on the computation power of the vehicle computer. In this case, the number of cameras is only limited by the available bandwidth.

Another topic relevant for driving in a public 5G network is cyber security. While this topic is out of the scope of this paper, we still want to mention that this is actually one advantage of using 5G campus networks, as these can be operated completely separate from public internet, making it more difficult for threat actors to access the network.

5 CONCLUSIONS

In this paper we examined remote driving using a 5G Standalone campus network. The results indicate that remote driving is indeed feasible under these conditions. While parts of these results also carry over to remote driving in public traffic, there is still the question if the mobile connection is good enough, especially at locations having bad reception.

For practical deployment the question of how to handle connection loss has still to be answered, as we relied on a safety driver in this case, who might not be available in practical deployments. Another avenue of future research considers speeding up the video transmission system. While this can be achieved by employing newer hardware, even lower latencies might be achievable by carefully tuning the video codec. Last but not least, due to constraints on the test track, only velocities of up to 28 km/h were achieved. Even when only considering driving in urban areas, velocities of 70 km/h and more should be safely demonstrated, especially since higher velocities also put higher load on the video transmission due to faster changing scenery as well as the possible need to change base stations.

ACKNOWLEDGEMENTS

This work was supported by the project “5G Lab Germany Forschungsfeld Lausitz” under contract VB5GFLAUS, funded by the Federal Ministry of Transport and Digital Infrastructure (BMVI), Germany. We would like to thank Lars Natkowski for acting as remote driver and Ina Partzsch for the valuable feedback.

REFERENCES

Cao, K., Liu, Y., Meng, G., and Sun, Q. (2020). An overview on edge computing research. *IEEE Access*, 8:85714–85728.

Dahlman, E. and Parkvall, S. (2018). Nr-the new 5g radio-access technology. In *2018 IEEE 87th Vehicular Technology Conference (VTC Spring)*, pages 1–6. IEEE.

Dehshalie, M. E., Prignoli, F., Falcone, P., and Bertogna, M. (2022). Model-based selective image downsampling in remote driving applications. In *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*, pages 3225–3230.

den Ouden, J., Ho, V., van der Smagt, T., Kakes, G., Rommel, S., Passchier, I., Juza, J., and Tafur Monroy, I. (2022). Design and evaluation of remote driving architecture on 4g and 5g mobile networks. *Frontiers in Future Transportation*, 2.

Domingo, M. C. (2021). An overview of machine learning and 5g for people with disabilities. *Sensors*, 21(22).

Economic Commission for Europe of the United Nations (2010). Regulation No 125 of the Economic Commission for Europe of the United Nations (UN/ECE) – Uniform provisions concerning the approval of motor vehicles with regard to the forward field ofvision of the motor vehicle driver. Standard, UN/ECE.

Gnatzig, S., Chucholowski, F., Tang, T., and Lienkamp, M. (2013). A system design for teleoperated road vehicles. In *ICINCO (2)*, pages 231–238.

ISO/IEC 20922:2016 (2016). ISO/IEC 20922:2016 Information technology – Message Queuing Telemetry Transport (MQTT) v3.1.1. Standard, ISO.

Kakkavas, G., Nyarko, K. N., Lahoud, C., Kühnert, D., Küffner, P., Gabriel, M., Ehsanfar, S., Diamanti, M., Karyotis, V., Mößner, K., and Papavassiliou, S. (2022). Teleoperated support for remote driving over 5g mobile communications. In *2022 IEEE International Mediterranean Conference on Communications and Networking (MeditCom)*, pages 280–285.

Kim, J., Choi, Y.-J., Noh, G., and Chung, H. (2022). On the feasibility of remote driving applications over mmwave 5 g vehicular communications: Implementation and demonstration. *IEEE Transactions on Vehicular Technology*, pages 1–16.

Klöppel-Gersdorf, M., Trauzettel, F., Koslowski, K., Peter, M., and Otto, T. (2021). The fraunhofer ccit smart intersection. In *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, pages 1797–1802. IEEE.

Klöppel-Gersdorf, M. and Otto, T. (2022). A framework for robust remote driving strategy selection. In *Proceedings of the 8th International Conference on Vehicle Technology and Intelligent Transport Systems - VEHITS,*, pages 418–424. INSTICC, SciTePress.

Liu, R., Kwak, D., Devarakonda, S., Bekris, K., and Iftode, L. (2017). Investigating remote driving over the lte network. In *Proceedings of the 9th International Conference on Automotive User Interfaces and Interactive Vehicular Applications*, AutomotiveUI ’17, page 264–269, New York, NY, USA. Association for Computing Machinery.

Nash, C. J., Cole, D. J., and Bigler, R. S. (2016). A review of human sensory dynamics for application to models of driver steering and speed control. *Biological cybernetics*, 110(2):91–116.

Neumeier, S., Wintersberger, P., Frison, A. K., Becher, A., Facchi, C., and Riener, A. (2019). Teleoperation: The holy grail to solve problems of automated driving? sure, but latency matters. pages 186–197.

Rischke, J., Sossalla, P., Itting, S., Fitzek, F. H. P., and Reisslein, M. (2021). 5g campus networks: A first measurement study. *IEEE Access*, 9:121786–121803.

Saeed, U., Hämäläinen, J., Garcia-Lozano, M., and David González, G. (2019). On the feasibility of remote driving application over dense 5g roadside networks. In *2019 16th International Symposium on Wireless Communication Systems (ISWCS)*, pages 271–276.

Sato, Y., Kashihara, S., and Ogishi, T. (2022). Robust video transmission system using 5g/4g networks for remote driving. In *2022 IEEE Intelligent Vehicles Symposium (IV)*, pages 616–622. IEEE.

Sredojev, B., Samardzija, D., and Posarac, D. (2015). Webrtc technology overview and signaling solution design and implementation. In *2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pages 1006–1009.

Tener, F. and Lanir, J. (2022). Driving from a distance: Challenges and guidelines for autonomous vehicle teleoperation interfaces. In *CHI Conference on Human Factors in Computing Systems*, pages 1–13.

Zulqarnain, S. Q. and Lee, S. (2021). Selecting remote driving locations for latency sensitive reliable teleoperation. *Applied Sciences*, 11(21).