

Teleoperated Support for Remote Driving over 5G Mobile Communications

Grigorios Kakkavas*, Kwame Nseboah Nyarko[†], Charbel Lahoud[†], David Kühnert[†], Peter Küffner[†], Matthias Gabriel[†], Shahab Ehsanfar[†], Maria Diamanti*, Vasileios Karyotis*, Klaus Mößner[†] and Symeon Papavassiliou*

*School of Electrical & Computer Engineering, National Technical University of Athens
Iroon Polytechniou 9, Zografou, Athens, 15780, Greece

e-mails: {gkakkavas, mdiamanti, vassilis}@netmode.ntua.gr, papavass@mail.ntua.gr

[†]Professorship for Communications Engineering, Technical University Chemnitz

Str. der Nationen 62, 09111 Chemnitz, Germany

e-mails: {kwame-nseboah.nyarko, charbel.lahoud, david.kuehnert}@etit.tu-chemnitz.de,

{peter.kueffner, matthias.gabriel, shahab.ehsan-far, klaus.moessner}@etit.tu-chemnitz.de

Abstract—The fifth-generation (5G) mobile communications promise far more capabilities than competing technologies and are expected to enable a wide range of vertical use cases. Among them, remote driving has attracted much interest from both academia and the industry, either as a stepping stone on the road to fully autonomous driving or as its complement constituting a fallback/backup service. This paper presents the design, development, and evaluation of a teleoperated support (TeSo) service over 5G mobile communications, realized within the framework of the “5G HEalth, AquacultuRe and Transport (5G-HEART)” EU project. We present an end-to-end service architecture and a complete implementation, accompanied by validation trials and performance measurements in a real pilot over a commercial 5G network deployment with an actual vehicle that has been extended to support a variety of experimentation options. With this work, we demonstrate the feasibility of the proposed TeSo service, and we quantify various performance metrics of interest, such as latency, jitter, throughput, and loss rate for longer-term reference use.

Index Terms—5G mobile communications, teleoperated support, remote driving, validation trials

I. INTRODUCTION

The significant advancements in autonomous transportation and mobile networks have already allowed the entry of automated vehicles into the market and on public roads. Nevertheless, there is still much distance to cover for the transition from partially to fully automated vehicles. In this regard, remote driving constitutes a practical and promising complement that could serve as an intermediate step for realizing the broader vision of pure automation. Remote driving and teleoperation refer to the remote control of a vehicle using the available mobile network infrastructure [1]. On the one hand, information about the vehicle’s state and surroundings is transferred to the remote location, enabling a remote operator/driver to assume control by sending at least steering wheel, throttle, and brake control commands back to the remotely operated vehicle. Most importantly, though, this form of remote operator fallback authority can assist in

providing the required advanced safety level to the autonomous driving technology as an additive service, especially in extreme cases and hazardous situations [2].

The enhanced performance of the currently deployed 5G networks in terms of achieved throughput, latency, reliability, and connection density is the primary enabler of remote driving and teleoperation. Specific levels of Quality of Service (QoS) need to be met in order to maintain the remote operator’s sense of presence and awareness and thus ensure timely maneuvering of the remotely controlled vehicle [3]. Although extensive theoretical analysis exists in the literature regarding the required network Key Performance Indicators (KPIs) and the different 5G network architectures to support them, practical implementations are scarce, small-scale, and provide primitive functionality. In order to advance and exploit the remote driving technology into more tangible products and services, realistic testing and evaluation with real pilots over the 5G network infrastructure are essential.

In this work, we leverage our previous theoretical KPIs’ analysis [4] and preliminary implementation [5] to finalize the design, development, and vehicle integration of a Teleoperated Support (TeSo) service for remote driving, realized in the context of the “5G HEalth, AquacultuRe and Transport (5G-HEART) Validation Trials” 5G PPP Phase 3 project [6]. Going one step further, we proceed to the actual trialing of the TeSo service in a real pilot over the 5G network. The validation trials were performed in Schlettau, Germany, using Vodafone’s 5G Standalone Access (SA) deployment. Specifically, we remotely controlled a research vehicle properly equipped with sensors and actuators, and we collected approximately 500 minutes of raw measurements and video recordings. The measurements captured the traffic exchanged between the vehicle and the remote operation center while conducting different maneuvering scenarios, i.e., straight course, left/right turn, obstacle avoidance, and parking. The collected data was subsequently analyzed to calculate the achieved throughput, latency, jitter, and packet loss rate of both downlink and uplink communications and assess the 5G network infrastructure’s effectiveness

and efficiency in supporting such a remote driving use case.

The key contributions of this work are three-fold and can be summarized as follows:

- (i) The finalized Teleoperated Support (TeSo) for remote driving service, developed within the framework of the 5G-HEART project, is presented in detail regarding the designed end-to-end system architecture and the complete software implementation.
- (ii) The experimental setup of the TeSo service’s validation trials in a real pilot over a commercial 5G SA network deployment is analyzed together with the measurement methodology followed during experimentation.
- (iii) After proper network and packet analysis, the quantitative analysis of the raw measurements provides numerical results for the main network KPIs and valuable insights into the network’s performance and stability.

The remainder of this paper is organized as follows. In Section II, we present the relevant literature and its limitations. Section III introduces the TeSo service’s architecture and technical details of the software implementation. Section IV describes the experimental setup and the followed measurement methodology. Finally, Section V encloses the performance evaluation of the TeSo trials, while Section VI concludes the paper.

II. RELATED WORK

Several attempts can be found in the literature that theoretically study the provisioning of remote driving services using 5G network connectivity from different perspectives, e.g., [4], [5], [7]–[9]. Considering the 5G-HEART project’s perspective, which adopts a “from theory to practice” scientific and technical path toward the final transport vertical validation trials, an analytical approach is followed in [4], and the high-level end-user QoS requirements of the TeSo service are quantified in the form of network KPIs values. Additionally, a preliminary 5G network slice dimensioning is performed that allows the concurrent satisfaction of the network KPIs of a multitude of vehicular services that are not limited to remote driving and are in the scope of the 5G-HEART project. Subsequently, in [5], we discuss and demonstrate the hardware and the preliminary software implementation of the different components that comprise a TeSo service. Similar to [4], a network KPIs and architectural analysis is presented in [7], differentiated by the fact that teleoperated transport and logistic services are jointly studied. On the other hand, a teleoperated driving system design that is mainly focused on the privacy and security aspects of the wireless communication networks is introduced in [8], incorporating failsafe modes to allow traceability in emergencies. Other works, e.g., [9], are dedicated to the development of Quality of Experience (QoE) evaluation systems on the impact of the network constraints on teleoperated driving scenarios.

In contrast to the extensive theoretical analysis, there exist only a handful of actual trialing efforts using 5G connectivity to support the functions and operation of automated vehicles in general (e.g., [10], [11]) and especially of remote driving

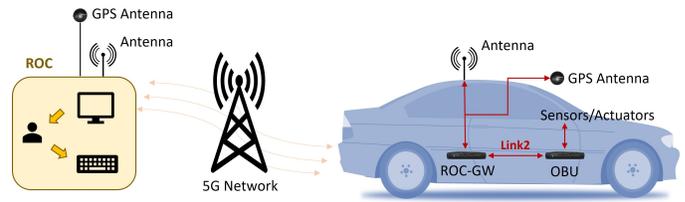


Fig. 1. High-level overview of the TeSo service’s architecture.

applications (e.g., [12]–[14]). Specifically, the works [10] and [11] consider basic functions of automated driving and evaluate the achieved coverage, mobility, latency, and packet loss, respectively, after experimentation and trialing in real 5G sites. Shifting the focus to remote driving applications and services, a simulation-based remote unit is used in [12] to control a research vehicle over the 5G network remotely. A latency evaluation that considers the WiFi-based wireless connectivity as a benchmark solution is also provided. In [13], a multiple Public Land Mobile Networks (PLMN) 5G Standalone (SA) experimental testbed is used, and a remote driving application is evaluated in terms of the achieved Reference Signal Received Power (RSRP) and Signal to Interference plus Noise Ratio (SINR). Following a more thorough investigation, the authors in [14] aim to distinguish the sensory and actuator latency in a teleoperation scenario, while a Long Term Evolution (LTE) network is considered to measure the end-to-end service’s latency.

III. TELEOPERATED SUPPORT FOR REMOTE DRIVING: THE 5G-HEART PERSPECTIVE

A. End-to-End System Architecture

Fig. 1 illustrates the end-to-end system architecture for the proposed TeSo service. The experimentation vehicle is equipped with a number of sensors and actuators capable of measuring and controlling its velocity, steering wheel angle, and throttle/brake position, together with four cameras mounted on each side (i.e., front, back, right, and left). As can be seen, the main components of the employed setup are:

- the *Remote Operations Center (ROC)*, which serves as an interface with the human operator, displaying the received telemetry data/video streams and accepting the remote control/navigation input;
- the *On-board Unit (OBU)* that interfaces with the sensors, cameras, and actuators of the vehicle at a low level in order to capture operational and ambient data, make it available in a standard format to other HW/SW components, and translate the remote control commands to signals compatible with the vehicle’s actuators;
- the *Remote Operations Center-Gateway (ROC-GW)* that processes the data published by the OBU and acts as an intermediate point for the communication between the vehicle and the ROC, transmitting in the uplink to the remote location the video streams/sensor data and receiving in the downlink the remote control commands; and

- the *network infrastructure*, which is responsible for realizing the communications between the ROC and the ROC-GW.

On the vehicle side, the ROC-GW, the OBU, and the various sensors/actuators are interconnected in a local mesh network, whereas the remote communications between the ROC and the ROC-GW are realized over the public network leveraging, among others, the 5G cellular technology.

B. TeSo Service Implementation

The proposed TeSo service was implemented in C++ using the DRAIVE Link¹ framework as a middleware to publish and subscribe to sensor data and actuator commands. In the remainder of the current section, we provide details about the main software components of the developed solution.

1) *Sensor/Actuator Data Format*: In Link, the communication between nodes that are part of the same local mesh network is realized via *messages* using a publish-subscribe communication pattern. Messages carry different data as payloads and are sent between the nodes using network transport protocols. The data types have a FlatBuffers² table format. Data objects hold the actual data, and they are an instance of the corresponding data types. The same data objects are used for the communication between the ROC-GW and the ROC over the 5G network using the ZeroMQ³ pub-sub pattern. The Flatbuffers table data types used for the TeSo service are the following:

1. *Camera* containing the JPEG-compressed frames for each video stream (front, back, right, and left).
2. *Vehicle State* retaining the vehicle's velocity in m/s and the corresponding timestamp.
3. *Automation State* including the currently set throttle percentage, brake percentage, and wheel angle in radians.
4. *GNSS Position* holding the latitude, longitude, and corresponding timestamp.
5. *Throttle Control* denoting the desired throttle percentage.
6. *Brake Control* conveying the desired throttle percentage.
7. *Steering Wheel Control* indicating the desired wheel angle in radians.

2) *Remote Operations Center-Gateway (ROC-GW)*: The ROC-GW node acts both as a subscriber with seven input pins and a publisher with three output pins. Generally, output pins assemble the desired data into messages and send them to other nodes by pushing them to the mesh, whereas input pins receive these messages and decompose them into data objects for further processing. In this particular case, the input pins receive the messages that carry data objects of types 1–4 described in Section III-B1 and forward them to the ROC over the 5G network. To that end, the respective Flatbuffers tables are encapsulated inside ZeroMQ messages and are transmitted to the ROC using ZeroMQ sockets and the pub-sub messaging pattern, with ROC-GW being the publishing

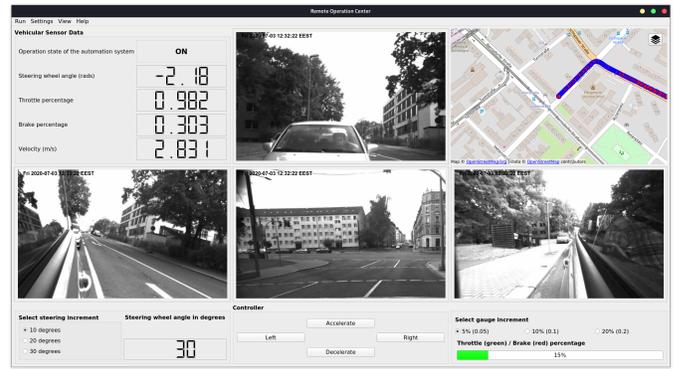


Fig. 2. ROC GUI application.

endpoint and ROC the subscribing endpoint. Particularly for the four camera input pins (i.e., front, back, right, and left), the received messages corresponding to individual frames are first suitably processed (i.e., resolution and JPEG-compression) before being sent to the ROC. On the other hand, the output pins pack data objects of types 5–6 that correspond to remote control commands into messages and push them to the local mesh network in order to reach the OBU and, ultimately, the vehicle's actuators. These control data objects are first received from the ROC over the 5G network using ZeroMQ sockets and the pub-sub message pattern in the reverse direction (i.e., this time, ROC is the publishing and ROC-GW the subscribing endpoint).

3) *Remote Operations Center (ROC)*: The ROC GUI application (Fig. 2) was implemented leveraging the Qt5⁴ framework and its signals and slots mechanism, according to which separate threads are used for receiving data objects of types 1-4 (see Section III-B1) from the ROC-GW using ZeroMQ SUB sockets and, after the required processing and data type conversions, suitable signals connected to the appropriate display slots of the main window's widgets are emitted and the respective information is presented to the operator. More precisely, there is a separate thread for receiving:

- the video frames of each camera that are ultimately displayed at the four screens comprising the bulk of the main window,
- the GNSS position coordinates that are used to draw a marker at the 2D map (written in QML by leveraging the Open Street Map Plugin⁵) located at the upper right corner of the window, and
- the vehicle and automation state telemetry data describing the current conditions of the vehicle that are reported in the upper left corner of the window.

The main window with all its component widgets is constructed by the main (or GUI) thread, which is also responsible for obtaining the remote operator's input (either from the push buttons located at the bottom of the window or via the arrow keys of the keyboard) and sending the corresponding data

¹<https://draive.com/docs/link2/>

²<https://google.github.io/flatbuffers/>

³<https://zeromq.org/>

⁴<https://www.qt.io/>

⁵<https://doc.qt.io/qt-5/location-plugin-osm.html>

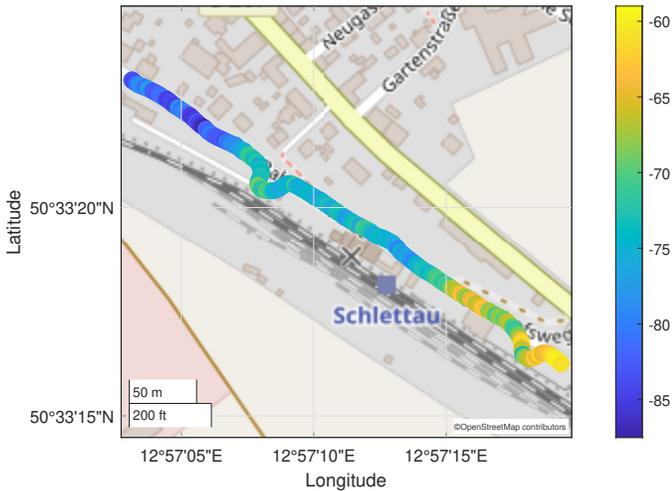


Fig. 3. 5G RSRP values across the trial pathway.

objects of types 5-8 to the ROC-GW, using ZeroMQ PUB sockets. At the bottom left corner, the user-provided angle is displayed in degrees, along with the respective selected increment step, whereas at the bottom right corner, a bar displays the user-provided throttle (or brake) percentage in green (or red) color, together with the corresponding selected increment step.

IV. VALIDATION TRIALS

This section presents in detail the experimental setup and the measurement methodology of the conducted validation trials. The trials presented comprise field studies representing realistic scenarios. This is particularly important in the case of functionality and effectiveness investigations that involve end-user engagement and field-deployable validation.

A. Experimental Setup

The mobile trials took place in a rural environment with a large number of cellular users. The 5G connection between the vehicle and the base station was operating in band n3 (UL: 1770.1 MHz, DL: 1865.1 MHz) and band n77 (UL: 3450.0 MHz, DL: 3450 MHz) of the spectrum. The 5G receiver node emulating the user equipment was deployed inside the research vehicle. Preliminary experimentation was conducted in the field to investigate the KPIs, availability, and reliability and obtain a fine-grain image of the provided 5G performance. Moreover, it should be noted that during the trials, the 5G signal was degraded by many factors such as NLOS (Non-Line-of-Sight) propagation, windows, trees, and house walls penetration. As shown in Fig. 3, the trials occurred over a wide RSRP range running from -62 dBm to -87 dBm.

The research vehicle's OBU, a Nuvo-4000 with Intel i7-3840QM 2.80GHz 4-core CPU and 16GB RAM, was interconnected with the following sensors via suitable Link nodes: four Allied Vision⁶ machine vision MAKO series

⁶<https://www.alliedvision.com/en/products/>

cameras and a u-blox⁷ AEK-4T GNSS device. Steering and throttling were remotely actuated by a dSPACE⁸ automation box and servomotors integrated into the vehicle and interfaced with an in-house MATLAB application. The ROC-GW and ROC software agents were hosted on two Dell Optiplex-7070s with Intel i9-9900 3.10 GHz 8-core CPUs and 32GB RAM, additionally equipped with a u-blox EVK-M8T and EVK-6T GNSS device, respectively. Traffic between the ROC-GW and ROC was securely transmitted using a SOC 2 compliant virtual private networking solution. 5G connectivity in-vehicle was realized via a commercial CPE router, from which signal and cell information were collated.

B. Measurement Methodology

Fig. 4 depicts the testing and measurement setup employed during the validation trials. As can be seen, the end-to-end chain of the developed TeSo service comprises three parts: the (inter)connection of the vehicle's sensors and actuators with the OBU and ROC-GW, the communications between the ROC-GW and ROC over the 5G cellular network, and the inclusion of the human in the vehicle's control-loop through the ROC GUI application. Roughly speaking, the human operator analyzes the presented video streams and sensor data, decides the desirable actions/maneuvers, and inputs the appropriate control commands that are transmitted back to the vehicle over the network. Within the scope of this work, we focus on measuring and providing suitable KPIs for this second part of the operation chain, which depends on the 5G connection, network coverage, and overall link conditions, and therefore varies over time and space. To that end, we take measurements for each data stream in the downlink and the uplink between the ROC-GW and the ROC at monitoring points 1–2 and 3–4, respectively. More precisely, traffic is captured at the points mentioned above using the tcpdump⁹ command-line packet analyzer, and the recorded .pcap files are post-processed and analyzed using the Wireshark¹⁰ network protocol analyzer and the pyshark¹¹ Python packet parser to calculate one-way latency, jitter, throughput, and loss rate. The necessary clock synchronization is ensured by Pulse-Per-Second (PPS) synchronization of each host to the GNSS reference time using chrony¹². The PPS signal jitter achieved was in the order of 1–20 μ s on average.

V. PERFORMANCE EVALUATION

In the following, we present a realistic evaluation of the designed and implemented TeSo service according to the setup described in Section IV-A. A large number of brief experimental rounds with the research vehicle being actively controlled remotely from a distance of around 36 km were conducted over five days. Table I provides an overview of the

⁷<https://www.u-blox.com/en/positioning-chips-and-modules>

⁸<https://www.dspace.com/en/lttd/home/products.cfm>

⁹<https://www.tcpdump.org/>

¹⁰<https://www.wireshark.org/>

¹¹<https://kiminewt.github.io/pyshark/>

¹²<https://chrony.tuxfamily.org/>

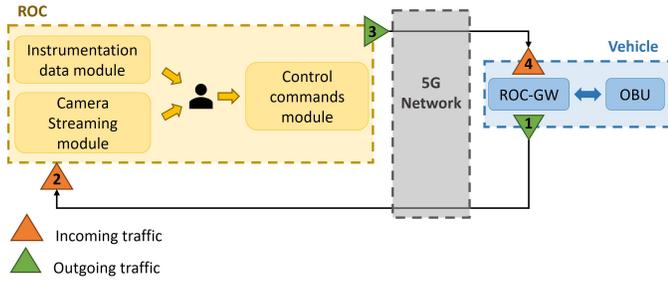


Fig. 4. Testing and measurement setup.

main results of interest for a small subset of those experimental scenarios. The selected rounds pertain to two different days in order to document the proposed service’s behavior under varying environmental conditions.

The reported mean one-way latencies for each data stream correspond to the time differences of the underlying ZeroMQ Message Transport Protocol (ZMTP) messages (i.e., reassembled ZMTP message frames¹³) measured at the respective monitoring points in the downlink and in the uplink between the ROC-GW and the ROC (i.e., monitoring points 1–2 and 3–4 in Fig. 4). Jitter is calculated using the following formula:

$$J_{ttr} = \frac{1}{n} \sum_{i=1}^n |D_i|,$$

where D_i is the time difference between two consecutive (i.e., i and $i + 1$) ZMTP messages. Mean throughput is computed over the entire duration of each scenario. As a representative visualization, Figs. 5 and 6 illustrate the evolution of throughput and the CDFs of the measured latencies of the received messages for selected data streams of a particular scenario.

Overall, as can be seen, throughput is higher in the uplink, driven by the four video camera streams, whereas the lowest latency values are generally recorded in the downlink and correspond to the three types of remote control commands (i.e., throttle percentage, brake percentage, and steering wheel angle). Such observations are in accordance with the desired behavior of the remote driving use case, where the involved instrumental sensor data and video streams that are communicated to the remote location of the human operator impose the uplink throughput requirement and the need for the remote human operator (or cloud-based application) to control the course and speed of the vehicle in real-time by sending appropriate control commands calls for low latency.

VI. CONCLUSION

This work sought to demonstrate the design, complete implementation, and experimental evaluation of a TeSo service over 5G mobile communications developed within the 5G-HEART project. In particular, we were able to showcase the effectiveness of the proposed end-to-end setup, from both the network and the user application perspective, by realizing

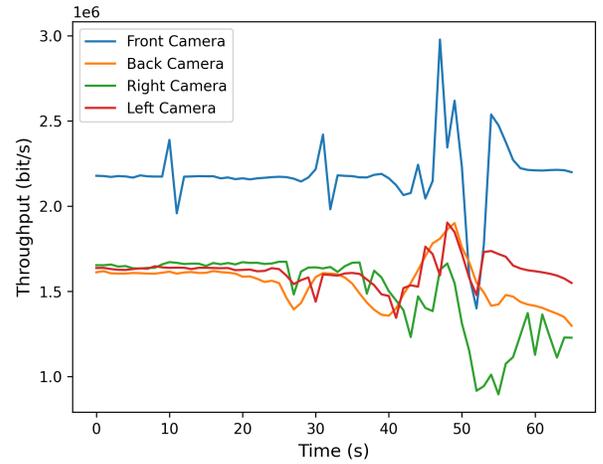


Fig. 5. Throughput of the four camera streams for scenario 20220511-01.

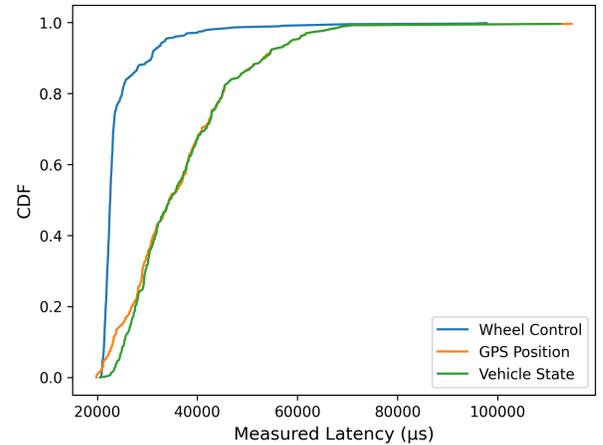


Fig. 6. CDFs of one-way latencies of the wheel control, GPS position, and vehicle state data streams for scenario 20220511-01.

validation trials in a real pilot and providing relevant measurements and results. Future work will seek to assess the end-to-end operation of the proposed service in longer rounds within a denser urban environment and compare the performance of 4G and 5G mobile communications under the same conditions to highlight the advantages brought by the newest generation of cellular networks for this particular use case.

REFERENCES

- [1] T. Zhang, “Toward automated vehicle teleoperation: Vision, opportunities, and challenges,” *IEEE Internet Things J.*, vol. 7, no. 12, pp. 11 347–11 354, Dec. 2020.
- [2] K. Kuru, “Conceptualisation of human-on-the-loop haptic teleoperation with fully autonomous self-driving vehicles in the urban environment,” *IEEE O. J. Intell. Transp. Syst.*, vol. 2, pp. 448–469, 2021.
- [3] G. Kakkavas, M. Diamanti, A. Stamou, V. Karyotis, S. Papavassiliou, F. Bouali, and K. Moessner, “5G network requirement analysis and slice dimensioning for sustainable vehicular services,” in *2021 17th International Conference on Distributed Computing in Sensor Systems (DCOSS)*. IEEE, Jul. 2021, pp. 495–502.
- [4] G. Kakkavas, M. Diamanti, *et al.*, “Design, development, and evaluation of 5G-enabled vehicular services: The 5G-HEART perspective,” *Sensors-basel.*, vol. 22, no. 2, p. 426, Jan. 2022.

¹³<https://rfc.zeromq.org/spec/23/>

TABLE I
SUMMARY METRICS OF SELECTED EXPERIMENTAL SCENARIOS

Data Stream	20220511-01				20220511-02			
	Mean Lat. (μs)	Jitter (μs)	Mean Throughput (bps)	Loss Rate (%)	Mean Lat. (μs)	Jitter (μs)	Mean Throughput (bps)	Loss Rate (%)
GPS Position	36617.49	13116.07	4654.3	0	26921.96	6663.69	4613.6	0
Front Camera	53537.61	7752.73	2183725.15	0.001574	40012.83	7492.74	2152510.4	0.016938
Back Camera	53176.58	7654.68	1551682.42	0	38882.8	6413.47	1543335.05	0.023198
Right Camera	51443.37	8848.42	1495421.91	0	41555.98	8597.07	1457143.8	0
Left Camera	52393.6	8816.91	1617039.09	0.001627	38792.72	6354.13	1567578.3	0
Throttle Control	28801	7636.54	335.12	0	23020.5	2997	102.2	0
Brake Control	29498		155.62	0	23096		9.63	0
Wheel Control	24463.54	4011.69	9537.32	0	24656.07	3913.7	9480.49	0
Vehicle State	37152.21	12434.08	4239.27	0	28505.15	6444.55	4226.4	0
Autom. State	39692.39	15426.54	4136	0	32035.05	9999.38	5021	0

Data Stream	20220512-01				20220512-02			
	Mean Lat. (μs)	Jitter (μs)	Mean Throughput (bps)	Loss Rate (%)	Mean Lat. (μs)	Jitter (μs)	Mean Throughput (bps)	Loss Rate (%)
GPS Position	38938.55	14206.14	4763.35	0	32516.78	9035.81	4672	0
Front Camera	51011.49	7590.75	2261223.55	0.005142	45234.14	6419.47	1820697.5	0
Back Camera	43501.43	6803.07	1181072.03	0	41703.67	7761.71	1151809.52	0
Right Camera	50252.32	7919.72	1218657.29	0	43683.84	8974.12	970273.83	0.008523
Left Camera	46599.97	6877.04	1639176.87	0	43374.09	6457.96	1412235.86	0.006134
Throttle Control	33301.22	7535.12	276.18	0	34603.19	10968.4	243.91	0
Brake Control	42684.67	1056.5	43.57	0	29295.25	9447.67	44.65	0
Wheel Control	30769.06	10560.5	9346.71	0	35528.23	5928.75	9559.26	0
Vehicle State	36110.96	12388.31	4198.19	0	33276.74	8542.37	4261.33	0
Autom. State	36180.84	13192.52	4070.19	0	33903.35	9646.81	5058.86	0

Data Stream	20220512-03				20220512-04			
	Mean Lat. (μs)	Jitter (μs)	Mean Throughput (bps)	Loss Rate (%)	Mean Lat. (μs)	Jitter (μs)	Mean Throughput (bps)	Loss Rate (%)
GPS Position	27904.83	7423.42	4598.04	0	45510.17	13695.96	4662.94	0
Front Camera	41042.25	7876.68	1704200.94	0	63505.16	7968.65	2390781.93	0
Back Camera	38012.05	6109.16	1202972.15	0	56570.85	9188.98	1209900.51	0
Right Camera	39233.04	7512.01	1315872.64	0	58653.18	9576.3	1042614.67	0.018302
Left Camera	39076.04	7125.61	1413714.08	0.019104	58513.41	9906.71	1450811.49	0
Throttle Control	25323.93	4785.54	250.86	0	34960.6	15674.22	119.9	0
Brake Control	30505		16.86	0	22777	814	22.75	0
Wheel Control	24725.95	6298.37	9474.29	0	34858.02	7068.63	9628.72	0
Vehicle State	27979.52	7610.99	4205.58	0	45884.5	13516.55	4245.01	0
Autom. State	28691.97	7748.15	4941.89	0	51312.72	12040.21	6354.12	0

* For the scenarios with only one transmitted brake control command, there is no corresponding jitter value.

- [5] G. Kakkavas, M. Diamanti, *et al.* "Demo proposal: Tele-operated support over 4G/5G mobile communications," in *2021 IEEE International Mediterranean Conference on Communications and Networking (MeditCom)*. IEEE, Sep. 2021, pp. 1–2.
- [6] 5G-HEART project. [Accessed: June 10, 2022]. [Online]. Available: <https://5gheart.org/>
- [7] J. M. Marquez-Barja, D. Naudts *et al.*, "Designing a 5G architecture to overcome the challenges of the teleoperated transport and logistics," in *2022 IEEE 19th Annual Consumer Communications & Networking Conference (CCNC)*. IEEE, Jan. 2022, pp. 264–267.
- [8] S. Neumeier, C. Corbett, and C. Facchi, "A secure and privacy preserving system design for teleoperated driving," in *Advances in Intelligent Systems and Computing*. Springer International Publishing, 2021, pp. 478–496.
- [9] C. Schuler, T. Gebauer, M. Patchou, and C. Wietfeld, "QoE evaluation of real-time remote operation with network constraints in a system-of-systems," in *2022 IEEE International Systems Conference (SysCon)*. IEEE, Apr. 2022, pp. 1–8.
- [10] M. Iwabuchi, A. Benjebbour, *et al.*, "Evaluation of coverage and mobility for URLLC via outdoor experimental trials," in *2018 IEEE 87th Vehicular Technology Conference (VTC Spring)*. IEEE, Jun. 2018, pp. 1–5.
- [11] M. Kutilla, K. Kauvo, *et al.*, "A C-V2X/5G field study for supporting automated driving," in *2021 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, Jul. 2021, pp. 315–320.
- [12] S. Baskaran, S. Kaul, S. Jha, and S. Kumar, "5G-connected remote-controlled semi-autonomous car trial," in *2020 IEEE International Conference on Machine Learning and Applied Network Technologies (ICMLANT)*. IEEE, Dec. 2020, pp. 1–6.
- [13] G. Pastor, E. Mutafungwa *et al.*, "Qualifying 5G SA for L4 automated vehicles in a multi-PLMN experimental testbed," in *2021 IEEE 93rd Vehicular Technology Conference (VTC2021-Spring)*. IEEE, Apr. 2021, pp. 1–3.
- [14] J.-M. Georg, J. Feiler, S. Hoffmann, and F. Diermeyer, "Sensor and actuator latency during teleoperation of automated vehicles," in *2020 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, Oct. 2020, pp. 760–766.