Demo Proposal: Tele-Operated Support over 4G/5G Mobile Communications

Grigorios Kakkavas*, Maria Diamanti*, Matthias Gabriel†, Charbel Lahoud†, Phanindra Akula†,
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: {matthias.gabriel, charbel.lahoud, phanindra.akula, klaus.moessner}@etit.tu-chemnitz.de

***Abstract*—This demonstration presents the design and implementation of a remote driving (tele-operated) support service over mobile communications with an actual vehicle, in the framework of the 5G-HEART project. Also, it evaluates the performance of existing cellular technologies, which is essential for better understanding and planning remote driving through 5G.**

***Index Terms*—4G/5G mobile communications, tele-operated support, remote driving.**

I. Introduction

Remote driving and tele-operation refer to the remote control of a vehicle using the available communication infrastructure. A human operator located remotely sends control commands to the vehicle over the network. At the same time, information about the vehicle's state and its surroundings is properly transferred and visualized back to the operator. The inclusion of a human with advanced perceptual/cognitive skills in the control loop of the vehicle alleviates the need for sophisticated algorithms and complex/expensive subsystems performing demanding real-time tasks like object identification and path planning required in fully autonomous driving.

Tele-operation can be seen as an additive service to the autonomous driving, increasing its reliability and effectiveness in complex situations, where the Autonomous Vehicle (AV) cannot efficiently maneuver an obstacle. In such cases, a remote driver can assume control of the vehicle through telecommunication networks. Apart from a first cost-feasible and transient step for the realization of the broader vision for purely automated driving, tele-operation can also offer a feasible solution for realizing lower-cost professional or industrial transportation and operating public support vehicles in harsh and hazardous environments. Finally, remote operation can be enabled on demand or in cases of a driver's health emergency.

Remote driving relies on the mobile network infrastructure and hence, strict requirements regarding the delay, loss, reliability, availability, and channel security are imposed. The most critical Quality of Service (QoS) parameters include (i) latency (regarding both the execution of the operator's commands and the transmission of vehicle's sensor information back to the operator), which can significantly affect cognitive functions such as spatial cognition, sense of presence, and awareness, and (ii) throughput/bandwidth, which is directly related to the ability of the operator to accurately perceive the vehicle's environment and state. For better understanding and planning the transition to 5G, it is essential to first identify the performance of the existing mobile network technologies.

In this demonstration, we present the end-to-end design and complete implementation of a Tele-operated Support (TeSo) service over mobile communications, aiming to assess its feasibility. To that end, we focus on identifying the network requirements for a tele-operated vehicle, exploring the related network challenges and analyzing the suitability of 5G for such services, by using 4G performance as a baseline.

II. Overview

The end-to-end system architecture for the provision of a TeSo service is illustrated in Fig. 1. The main components are a remotely operated vehicle, the mobile network infrastructure and a Remote Control Center (ROC) located at the edge of the network. The vehicle is equipped with the appropriate sensors and actuators to measure and control, respectively, its speed, acceleration, steering angle and brake position. Four cameras are mounted on each side of the vehicle (i.e., at the front, back, right and left side) to allow for video streaming to the ROC. An On-Board Unit (OBU) that interfaces with the sensors and the cameras is charged with capturing operational and ambient data and thus, enables their use by other hardware components that are integrated in the vehicle. The ROC Gateway (ROC-GW), in turn, is responsible for the processing and aggregation of the data extracted by the OBU and their final transmission over the 5G network to the ROC. Considering the opposite direction of the communication, the control commands transmitted over 5G from the ROC to the vehicle are received by the ROC-GW and forwarded to the OBU, where their appropriate conversion for the vehicle's actuators takes place. Hence, the OBU provides an interface with the vehicle's sensors and actuators, while the ROC-GW serves as an intermediate point for the communication between the vehicle and the ROC.

This work was supported by the European Commission in the framework of the H2020-ICT-19-2019 project 5G-HEART (Grant Agreement No. 857034).

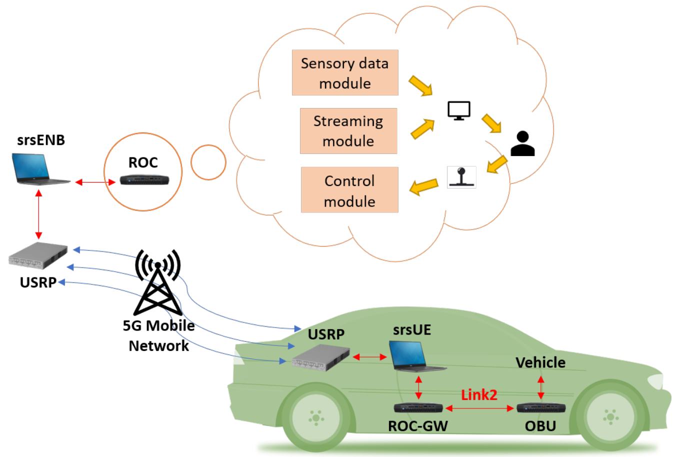

Fig. 1. Overview of the end-to-end architecture of TeSo.

III. Innovation

A. Hardware Components

The central part of the employed setup is an experimentation vehicle that is extended to support a variety of research options (e.g., sensors, software algorithms, actuators, Human-Machine Interface (HMI)). The sensors, HMI, and actuators form the fundamental layer of the vehicle's architecture, while the DRAIVE Link2 middleware [1] acts as an interface layer between the hardware and software, i.e., it receives the sensor data via vendor specific hardware interfaces, processes them into a common format and forwards them to the other layers. As seen in Fig. 1, all sensors of the vehicle are connected to a central computer, i.e., the OBU. This central computer has the Link2 middleware installed and is connected to a local network via Ethernet. Another crucial hardware component of the experimentation setup, which is directly related to the implementation of the mobile communications, is the Universal Software Radio Peripheral (USRP) device. Specifically, the mobile communication is realized using srsRAN [2], an open-source platform written in C language, which captures the signal over-the-air using a USRP N321 [3] and processes it through the 4G/5G stack. Automobile multi-band antennas are used on the USRPs, which cover the LTE and 5G frequency range (i.e., from 1.7 to 6 GHz).

B. Software Components

1) ROC-GW Node: The ROC-GW node is constructed using the DRAIVE Link2 framework. In order to transport a message from a sender (publisher) to the receiver (subscriber), a publish-subscribe communication pattern is used. Output pins assemble the desired data into messages and send them to other nodes. Input pins receive these messages and decompose them into data objects for further processing. A subscription is defined by mapping data fields of the publishers' offers to corresponding data fields of the demand of each input pin. The implemented ROC-GW node acts both as a subscriber and a publisher, having seven input pins and three output pins. According to the configured subscriptions, the input pins receive the respective table data objects accompanied with the respective timestamps. These objects include: (i) the JPEG encoded frames of the video streams from the front, back, right, and left cameras, (ii) the vehicle state corresponding to the vehicle velocity in meters per second, (iii) the automation state including the current steering wheel angle in radians, the throttle percentage value and the brake percentage value, and (iv) the GNSS position data. To forward these objects to the ROC over the 5G network, the publish-subscribe pattern of the ZeroMQ [4] asynchronous messaging library is employed, where the ROC-GW is the publishing endpoint and the ROC the subscribing endpoint. The output pins push the table data objects that they receive from ROC (i.e., the desired throttle percentage value, brake percentage value, and wheel angle in radians) to the mesh, where the OBU has subscribed. Once again, the ZeroMQ publish-subscribe messaging pattern is employed for the transmission of these remote-control commands from ROC to ROC-GW.

2) ROC Application: The ROC Graphical User Interface (GUI) application has been implemented using the Qt5 [5] framework and following a multi-threaded design. The main thread is responsible for the construction of the GUI's main window with the appropriate widgets for obtaining the remote operator's input that is sent as remote-control commands to the vehicle's ROC-GW. The signals and slots mechanism of Qt is used for displaying the vehicle's data sent from ROC-GW. In particular, separate threads are used for the reception of each data type through ZeroMQ (i.e., four video streams, vehicle state, automation state, and GNSS position), and, after the necessary processing and data type conversions take place, suitable signals connected to the appropriate display slots of the main window's widgets are emitted. Additionally, a Two-Dimensional (2D) map displayed at the upper right corner of the GUI's window visualizes the received GNSS data.

IV. Relevance

In this demonstration, we will present the complete end-to-end operation of the described TeSo service in a real pilot, including the full communication setup. To gain more insight about the functionality of the employed software and hardware as a whole, the standalone inspection of each component will be provided, considering its individual role and its integration to the complete architecture. This concludes the development and deployment phase of a typical TeSo service, required to further proceed to actual performance measurements.

References

[1] Draive link2. [Accessed: May 20, 2021]. [Online]. Available: https://draive.com/docs/link2/

[2] srsRAN: Your own mobile network. [Accessed: May 20, 2021]. [Online]. Available: https://www.srsran.com/

[3] Ettus Research USRP N321. [Accessed: May 20, 2021]. [Online]. Available: https://www.ettus.com/all-products/usrp-n321/

[4] P. Hintjens. (2011) Ømq - the guide. [Accessed: May 17, 2021]. [Online]. Available: https://zguide.zeromq.org/

[5] L. Z. Eng, *Qt5 C++ GUI programming cookbook*. Packt Publishing Ltd, 2016.