Network Trace Collection in a Moving Car

Bachelor’s Thesis

Supervisor: Prof. Dr. Alexander Pretschner
Advisor: Thomas Hutzelmann
Email: {alexander.pretschner, t.hutzelmann}@tum.de
Phone: +49 (89) 289 - 17830
Starting date: immediately (September 2018)

Context

In the automotive domain formerly isolated, embedded systems are now extended with modern computers and connected to the Internet. However, old components and structures are left mostly unchanged in the car to fulfill the basic driving functionalities. The combination of legacy and modern systems inhibits several questions and problems.

Especially the network communication between these systems has unique requirements, e.g. safety properties and a strictly limited budget on the used hardware. For insights about realistic network traffic, car manufacturers are using special logging devices. However, these devices are brand/model specific, very expensive and require physical changes inside the car implying a special license to operate such a modified car on the streets. This makes it problematic for academia to collect a broad variety of traces.

Luckily, all vehicles manufactured after 2008 require a special maintenance port, the so-called OBD-port, that has to be easily accessible from inside the car and enables access to the CAN bus [2]. Hence, we will use cheap, Arduino based hardware supporting that port [3] as an alternative logging-solution. In this bachelor’s thesis, we will extend this basic hardware and the corresponding open source software [4] to fulfill the requirements of scientific trace collection. Additionally, we want to conclude with some simple analysis of the collected data.

Goal

Build a working OBD-Dongle with plug-and-play logging capabilities and use the collected traces for some basic analysis.

Working Plan

1. Familiarize yourself with programming on Arduino and car networks
2. Customize the embedded logging software for the OBD-dongle
3. Collect Traces from moving cars containing
   (a) All raw packages accessible to the OBD-port
   (b) GPS trace of the drive
   (c) Meta-Information (e.g. VIN, data, time, duration)
4. Analyse the traces, e.g. you can
   (a) give insights about the amount of available information
   (b) build your own speed-o-meter
   (c) learn profiles for different drivers/cars
   (d) locate the car by speed and steering
   (e) pursue your own ideas

Deliverables

- Source code of the implementation and the working OBD-Dongle directly usable in car
- Technical report with comprehensive documentation of the implementation, i.e. design decision, architecture description, API description and usage instructions.
- Final thesis report written in conformance with TUM guidelines.

References

[1] OBDII data logger design for large-scale deployments, by Smith et al.

Remark: All required hardware, i.e. the dongle and development equipment is provided by our chair. A “suitable” car or a driving license is not mandatory, but could be helpful.