Hot-wiring of the future - exploring car CAN buses

Overview

Modern cars have many interconnected networks (e.g. CAN, LIN, FlexRay, MOST, D2B). CAN (Controller Area Network) is a serial broadcast bus connecting systems and sensors and the de-facto standard for in-vehicle data transmission CANs are used also in satellites, industrial automation, medical equipment and more.

In cars, CANs are used to control power-engine systems, door and roof control units as well as lighting and seat control units, infotainment devices. Different CAN-based networks are connected via gateways (see Figure 1 for CAN networks in a Volkswagen Passat)

CAN bus-equipped vehicles started appearing in model year 2003. As of 2008 all vehicles sold in the US are required to implement CAN as one of their signaling protocols. The Onboard diagnostic system ODB-II (typically near the dashboard, via pin 6 and 14) offers a way to communicate with CANs.

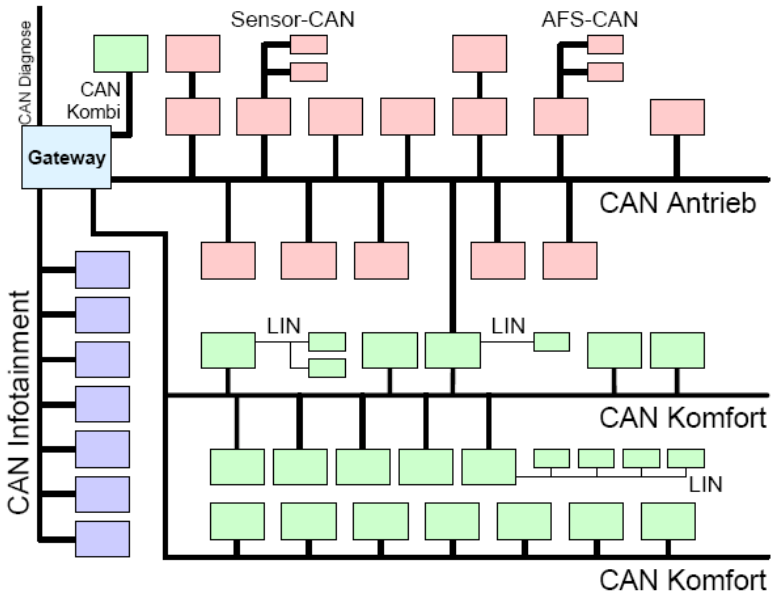

Figure 1: Different types of CAN networks (with different data rates) used for systems: CAN Antrieb is used for powertrain and chassis systems, whereas CAN Komfort and CAN Infotainment are used for body and multimedia/infotainment systems respectively.

Goals

The larger goal of this project is to gain visibility into a car's CAN bus networking, specifically to understand the interconnecting higher level protocols (see Figure 2). Applications that control systems, from windshield wipers to brakes to engine to

entertainment system, rely on these HLPs. Possibilities of successful HLP parsing and subsequent control are nigh endless: You could sync wiper frequency and cleaning fluid squirts to Ravel's Bolero playing on the infotainment system. Or have the headlights morse-code GPS coordinates. Or induce the engine to mechanically self-destruct through carefully generated timed signals[1].

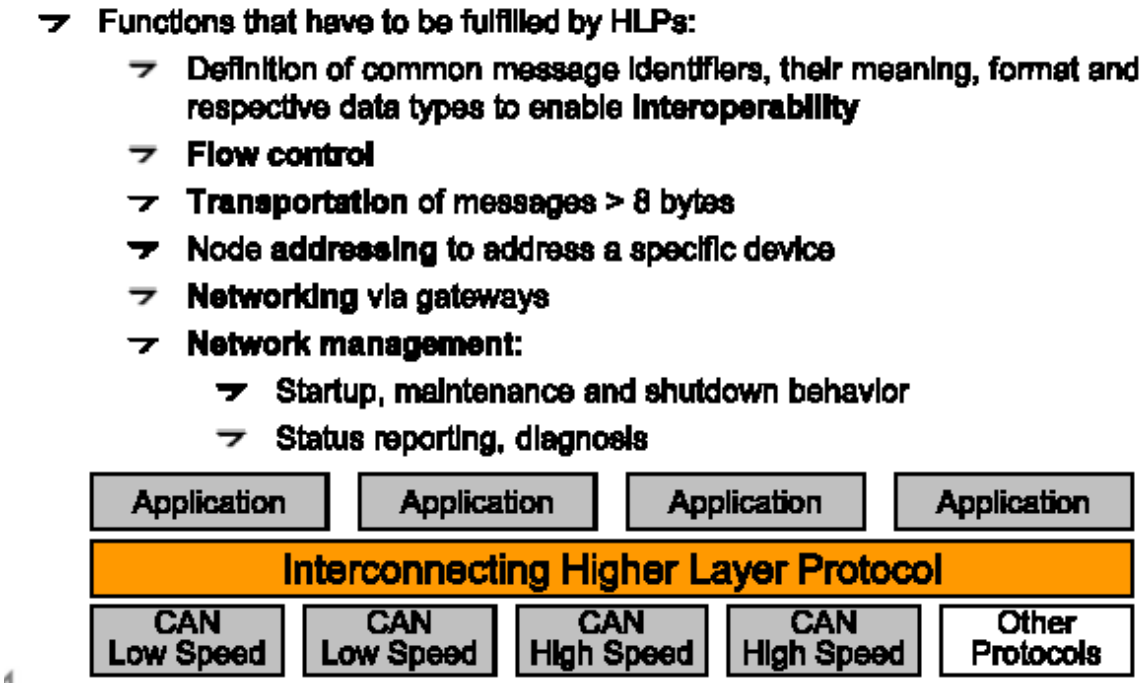

Figure 2: CAN high-level protocols

Deliverables

For the project, we'll use a GoodFET board connected to the ODB-II to communicate with CAN (connector and GoodFET will be provided and students can keep the board and connector after the project)

Deliverables are

1. **Step1: Software** Students will use the provided board to first read out data frames, then reverse engineer HLP and develop software that can inject, record, replay, then inject/evade higher and lower level data frames.
    a. One deliverable is a protocol analyzer module that can integrate with the open-source Wireshark Protocol Analyzer.
    b. Second deliverable is software to inject protocol data frames into the CAN networks.
2. **Step 2: Hot-wiring of the Future:** In consultation with sponsors and advisors, students should develop a PoC demonstration of the software CAN

[1] I know you like that one: Checkout what some say was the PoC to Stuxnet, 2007 Aurora power generator https://www.youtube.com/watch?v=fJyWngDco3g

bus protocol and HLP protocols capabilities by controlling one or more of the car's systems. This could be as simple as injecting false sensor readings to more elaborate schemes like discussed above

3. **Final report:** A final report documenting your steps. Proposed format https://www.dropbox.com/s/zlmkzqvhopir5yt/Report%20Format.pdf

Technical liaisons are Project sponsors from Siege Technologies, Drs. Joe Sharkey and Daniel Bilar, Dr. Sergey Bratus from Dartmouth CS, GoodFET / bus hacking expert Travis Goodspeed.

Equipment

- **Required:** Need vehicle with ODB-2 support for CAN communications
- **Useful:** Car system firmware dumps

Knowledge Areas Needed for Project

- **Required**: Networking course, Software Engineering
- **Useful**: Digital Signal Processing, basic hardware experience

Proprietary Information and Confidentiality Reqs

Software will be BSD licensed. http://www.linfo.org/bsdlicense.html
US citizenship required.

References

[1.] Literature review IN-VEHICLE COMMUNICATION NETWORKS http://alexandria.tue.nl/repository/books/652514.pdf

[2.] Car and Driver "Can your car be hacked?" http://www.caranddriver.com/features/can-your-car-be-hacked-feature

[3.] Bratus "What Hackers Learn that the Rest of Us Don't" http://www2.computer.org/cms/Computer.org/ComputingNow/homepage/0808/WhatHackersLearnThatTheRestOfUsDont.pdf

[4.] GoodFET http://goodfet.sourceforge.net/

[5.] Wireshark https://www.wireshark.org/

[6.] Course on Vehicle Networks
https://www.dropbox.com/sh/brh4u3rcuaqzwin/SYMfzRbWKU