Demo Proposal: Prototyping Next-Generation In-Car Backbones Using System-Level Network Simulation

Till Steinbach, Philipp Meyer, Stefan Buschmann, Franz Korf, and Thomas C. Schmidt
Department of Computer Science, Hamburg University of Applied Sciences, Germany
{till.steinbach, philipp.meyer, stefan.buschmann, franz.korf, t.schmidt}@haw-hamburg.de

***Abstract*—We show a network simulation environment for assessing Ethernet-based concepts and technologies of next generation in-car networks, as well as their protocols, and possible deployment in topologies. Among others, the simulation models contain the core concepts of AS6802 and AFDX, Ethernet AVB and IEEE 802.1Q as well as legacy fieldbus technologies like CAN and FlexRay and automotive gateway designs to interconnect the technologies. All modules can be flexibly configured and combined or used as a foundation for the implementation of new ideas. System-level network simulation allows us to design and evaluate backbone architectures and develop protocols and configurations that comply with the rigid real-time requirements of in-car communication. The shown toolchain is open source and can be downloaded for experiments and reviews of published simulation studies at http://core4inet.realmv6.org**

***Index Terms*—Automotive Network Simulation, Real-time Ethernet, Fieldbus, Automotive Gateways, Performance Evaluation**

I. Introduction & Motivation

Over the last years, the number of electronic systems in cars heavily increased. Especially in the areas of info- and entertainment as well as driver assistance systems the demand for faster communication technologies for the in-car network is growing. This trend won't cut off with upcoming topics such as automated and finally autonomous driving and new high bandwidth sensors, such as cameras, laser scanners, and radar, being added to the car. Todays heterogeneous in-car network architectures, mainly consisting of fieldbus technologies such as Controller Area Network (CAN), FlexRay, or Media Oriented System Transport (MOST) with limited bandwidth and shared collision domains won't cope with the upcoming challenges.

Ethernet is the most promising candidate for the next-generation in-car networks. It offers a flexible physical layer, high bandwidth and scalability due to its switch-based topology with isolated congestion domains on its links. There are two shortcomings of standard switched Ethernet to overcome: The first is the physical layer. For the cost efficient use in the automotive domain unshielded cables are preferred. The BroadR-Reach technology by Broadcom that is currently being standardized by the IEEE under PAR 802.3bw, offers 100 Mbit/s over a single unshielded pair of twisted wires while passing the stringent automotive electromagnetic emission requirements. The second shortcoming are the limited quality of service capabilities of standard switched Ethernet. Ethernet must be extended to enable hard real-time communication. Several real-time Ethernet extensions and shaping strategies are currently under investigation. Examples are Ethernet AVBs credit based shaping or scheduled traffic as in AS6802.

Designing Ethernet based in-car network architectures at this early stage is complex. Where todays fieldbus configurations are designed based on the experiences gained in decades of development and deployment, the design of switched real-time networks with a multitude of configuration parameters poses new challenges. Moreover, gateways translating messages between todays fieldbus technologies and the real-time Ethernet backbone are required to transparently integrate legacy hardware and preserve the investment in fieldbus based systems.

We propose system-level network simulation for the early design stage of real-time Ethernet based in-car networks. We present a simulation environment consisting of flexibly interconnectable models of real-time Ethernet protocols and shapers, legacy fieldbus technologies, realistic traffic sources, and gateway concepts. Our open source models are based on the established OMNeT++ discrete event based network simulator. In our demonstration we show how network simulation at system-level can help to develop and evaluate new real-time Ethernet protocols and traffic shaping strategies, assess network and gateway configuration parameters, and predict hardware requirements for future electronic control units (ECUs). Typical network metrics that can be obtained using system-level network simulation are for example latencies, jitter, buffer sizes, bandwidth utilization, or synchronization precision.

II. Background & Related Work

The demonstrated simulation environment consists of several modules for the simulation of real-time Ethernet technologies, legacy fieldbusses, and automotive gateways. It bases on the open source OMNeT++ network simulator and can be downloaded for simulation experiments and reviewing at http://core4inet.realmv6.org.

A. OMNeT++ Network Simulator

OMNeT++ [1] is a discrete event based simulation platform mainly focussing on the simulation of networks and multiprocessor systems, but designed to be as general as possible. It is a perfect base for a simulation toolchain for automotive communication. We developed our framework as an extension of the popular OMNeT++ INET-Framework [2] that provides the implementation of the physical layer as well as protocols and applications above layer 2, such as IP, TCP or UDP.

B. CoRE4INET Real-time Ethernet Models

The CoRE4INET (Communication over Real-time Ethernet for INET) framework provides simulation models for the real-time Ethernet extensions AS6802 (time-triggered and rate-constrained), AFDX, Ethernet AVB as well as strict prioritization according to IEEE 802.1Q. It was developed based on the specifications and carefully validated using both analytical models as well as measurements using prototype hardware [3].

The CoRE4INET framework was developed with the goal to evaluate real-time Ethernet protocols as well as to develop new concepts. Thus it provides building blocks that can be flexibly interconnected to implement new behavior. These blocks include models for:

- Realistic oscillators with parametrizable clock drift
- A central system scheduler for timers and time-triggered events
- Buffers with configurable behavior and size, such as double buffers and queue buffers
- Traffic shapers with different behavior, e.g. time-triggered or credit based shaping
- Clock synchronization

All modules can be flexibly parametrized or used as stubs for the implementation of new protocols and systems.

C. FiCo4OMNeT Fieldbus Models

The FiCo4OMNeT (Fieldbus Communication for OMNeT) models implement the popular legacy automotive fieldbus technologies CAN and FlexRay. The models were implemented according to the latest specification and evaluated [4] using analytical models and established commercial tools like CANoe. Similar to the CoRE4INET framework, FiCo4OMNeT features oscillators and a scheduler. The implementation of CAN contains all features required in automotive networks, such as CAN A and B mode, error and remote frames and error handling. The FlexRay model provides the network synchronization as well as communication in the dynamic and the static segment.

D. Models for Signals and Automotive-Gateways

The third part of our automotive system-level network simulation environment provides simulation models for typical signals of automotive networks as well as the tools to simulate complex gateways between (real-time) Ethernet and fieldbus technologies. Even complex gateway configurations can be achieved using a XML configuration. The models contain building blocks for the routing of frames between different busses and networks, transformation modules between the different fieldbus and real-time Ethernet technologies, and frame aggregation strategies to enable the compressed transmission of multiple small messages in larger Ethernet frames.

III. Setups & Use-cases

The demonstrated simulation environment can be used for several use-cases. In the following we present examples of network analyses that were done using the toolchain.

A. Traffic Shaper Concepts

Using the demonstrated simulation environment we evaluate new traffic shaping concepts such as a time-aware shaper that merges Ethernet AVBs credit based shaper with the time-triggered concepts of AS6802 [5] similar to the standardization efforts under IEEE PAR 802.1Qbv. We were able to show a significant impact of the schedule design of the time-triggered message class on the asynchronous AVB streams. With the demonstrated simulation environment it is possible to combine traffic classes of different real-time Ethernet variants and analyze their interference without the need of prototype hardware. The simulation allows to precisely analyze timing such as end-to-end latency or jitter and thereby compare different shaping concepts.

B. Gateway Strategies

In todays automotive gateways usually traffic from one fieldbus is forwarded to another fieldbus of the same technology. When deploying an Ethernet based in-car backbone with edge networks using legacy technologies such as CAN or FlexRay, gateway designs become more complex. Transparently tunneling messages transmitted between ECUs attached to the fieldbusses without wasting bandwidth on the Ethernet core requires trade-offs whose influences on the timing have to be carefully assessed. A central feature of CAN-Ethernet gateways for example is the aggregation of multiple CAN messages into one Ethernet frame. There are several strategies to schedule the transmission of aggregated messages and assign messages to pools that share one Ethernet frame. The aggregation adds additional delays to the messages. By using system-level network simulation we can analyze the influence of those strategies.

C. In-Car Backbone Design

System-level network simulation allows us to design backbone architectures (see Figure 1) and develop configurations that comply with the rigid real-time requirements of in-car communication. We successfully utilized the demonstrated toolchain to simulate a real-world prototype car that was equipped with a real-time Ethernet backbone [6]. Thanks to the opportunity to simulate several parameter sets in advanced the bring up time of the prototype could be significantly reduced. In the RECBAR research project we use the simulation environment to work on new backbone architectures. By using the traffic patterns of a current series car we can precisely predict the achievable network metrics of proposed architecture variants. The network metrics obtained in the simulation with realistic traffic flows are directly transferrable to the real-world prototype. This allows us to evaluate different parameter sets prior to the deployment in the car and thus saves a significant amount of setup time. Also errors in the configuration can be found faster as the simulation allows a deep view into the system that can be only achieved with significant effort in the real network.

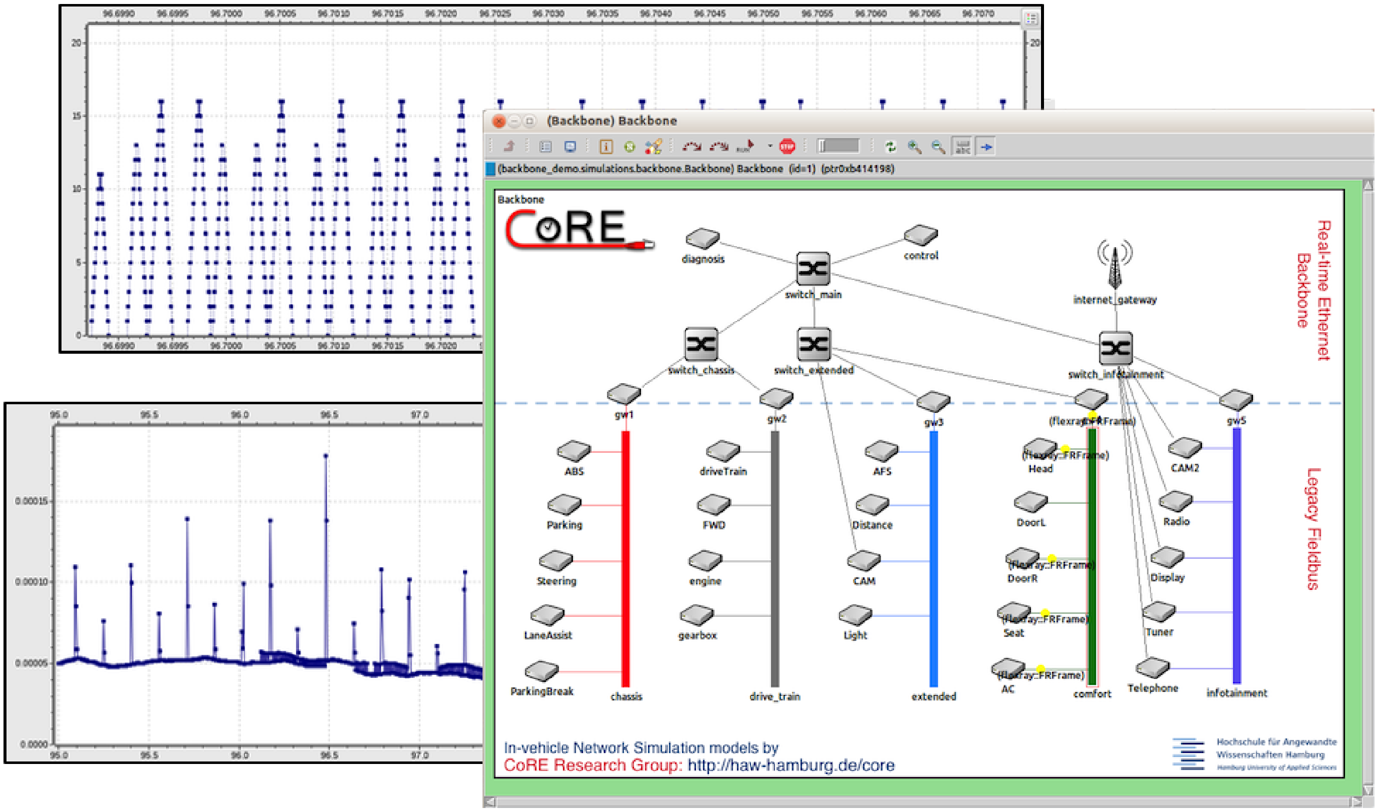

Fig. 1. Screenshot of the simulation of an in-car backbone architecture consisting of a real-time Ethernet core and attached legacy fieldbusses

IV. Conclusion & Future Work

In-car communication will slowly transition from todays fieldbus technologies to switched networks. Network simulation on system level will help to support this process by providing a framework to evaluate real-time and application protocols, develop new shaping strategies, assess architectures, and predict hardware requirements long before first real world prototypes are being realized. We show a simulation environment that can be used for these upcoming challenges and provide the building blocks to evaluate new ideas in the domain of in-car networking.

In our future work we develop interfaces to interconnect our simulator with established tools of the automotive domain, such as CANoe to ease the development effort for the user. We further work on tools to improve the visualization of network metrics obtained in the simulation, e.g. by adapting gantt charts to visualize delays in the path between sender and receiver. By adding new technologies to the simulation such as frame preemption as currently discussed in IEEE 802.1Qbu, we try to provide the tools to evaluate upcoming trends. A further possible extension is the implementation of CAN with flexible data rate (CAN FD).

Requirements for the Demo

A small table and a power outlet is required. The setup time is approx. 10 minutes. The demo is run on a laptop. Due to luggage limitations we would kindly ask the conference organizers to provide us a computer monitor (e.g. 22-24").

Acknowledgements

This work was supported by the German Federal Ministry of Education and Research (BMBF) under the RECBAR project.

References

[1] A. Varga and R. Hornig, "An overview of the OMNeT++ simulation environment," in *Proceedings of the 1st International Conference on Simulation Tools and Techniques for Communications, networks and systems & workshops*. New York: ACM-DL, Mar. 2008, pp. 60:1–60:10.

[2] OMNeT++ Community, "INET Framework 3.0." [Online]. Available: http://inet.omnetpp.org/

[3] T. Steinbach, H. Dieumo Kenfack, F. Korf, and T. C. Schmidt, "An Extension of the OMNeT++ INET Framework for Simulating Real-time Ethernet with High Accuracy," in *Proceedings of the 4th International ICST Conference on Simulation Tools and Techniques*. New York: ACM-DL, Mar. 2011, pp. 375–382.

[4] S. Buschmann, T. Steinbach, F. Korf, and T. C. Schmidt, "Simulation based Timing Analysis of FlexRay Communication at System Level," in *Proceedings of the 6th International ICST Conference on Simulation Tools and Techniques*. New York: ACM-DL, Mar. 2013, pp. 285–290.

[5] P. Meyer, T. Steinbach, F. Korf, and T. C. Schmidt, "Extending IEEE 802.1 AVB with Time-triggered Scheduling: A Simulation Study of the Coexistence of Synchronous and Asynchronous Traffic," in *2013 IEEE Vehicular Networking Conference (VNC)*. Piscataway, New Jersey: IEEE Press, Dec. 2013, pp. 47–54.

[6] T. Steinbach, K. Müller, F. Korf, and R. Röllig, "Real-time Ethernet In-Car Backbones: First Insights into an Automotive Prototype," in *2014 IEEE Vehicular Networking Conference (VNC)*. Piscataway, New Jersey: IEEE Press, Dec. 2014, pp. 137–138.