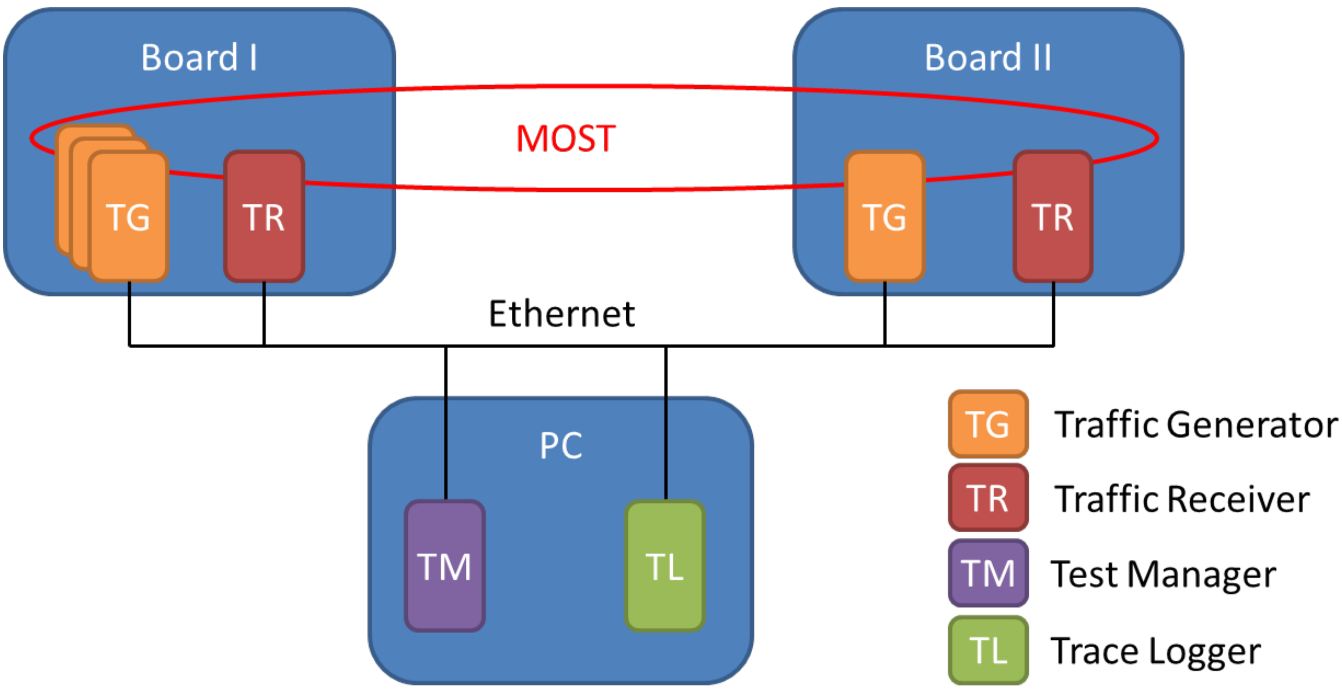
A Performance Evaluation Framework for MOST Ethernet Supported by Virtual Prototyping Technology

MOST offers numerous dedicated communication channels for the transmission of multimedia content in various formats and data rates. In addition to these dedicated channels, there is the Asynchronous Channel, over which MOST Ethernet offers a service referring to Ethernet, the well-known standard in computer networking. This service allows for a TCP/IP or UDP/IP architecture on top, which brings a vast range of applications into the automotive domain. These applications are of increasing significance, not only when thinking of on-board Internet access or connected cars.

Challenge

The usage of these applications not only requires a suitable protocol architecture, but further requirements concerning bandwidth or communication latency have to be fulfilled. In most cases, these requirements cannot simply be guaranteed by the communication protocol specification, but highly depend on the concrete implementation and communication infrastructure, including the operating system and the communication primitives used by the specific application. The goal of the assessment at FZI was to evaluate the performance of MOST Ethernet from an application perspective. The framework developed for this task had to be flexible in terms of use case coverage, offer a high degree of automation to support large numbers of measurements, and ensure reproducible experiments.

The physical setup, as shown in figure 1, consists of multiple Freescale i.MX6 developer boards and a controlling PC. The boards are each equipped with a MOST150 PhyBoard, through which they form the nodes of a MOST ring. Furthermore, they are connected over switched Gigabit Ethernet with the controlling PC. The nodes are powered by the Linux operating system, in which MOST Ethernet is accessible via the virtual Ethernet interface 'meth0'. This virtual network interface can be used by applications running on the node just as any other interface, e.g. for TCP/IP or UDP/IP socket communication.



Solution

On each of the MOST nodes, an evaluation application consisting of multiple modules is run: a set of traffic generators, a universal traffic receiver and a configuration client. The configuration client receives test case descriptions as XML documents from the controlling PC and equips the traffic generators to produce the desired communication data flow profile on the MOST Ethernet network. The receiver module acts as sink for incoming data and records traces, which are sent back to the controlling PC, where they are stored for off-line analysis. The traffic generators offer numerous parameters, set by the test case description:

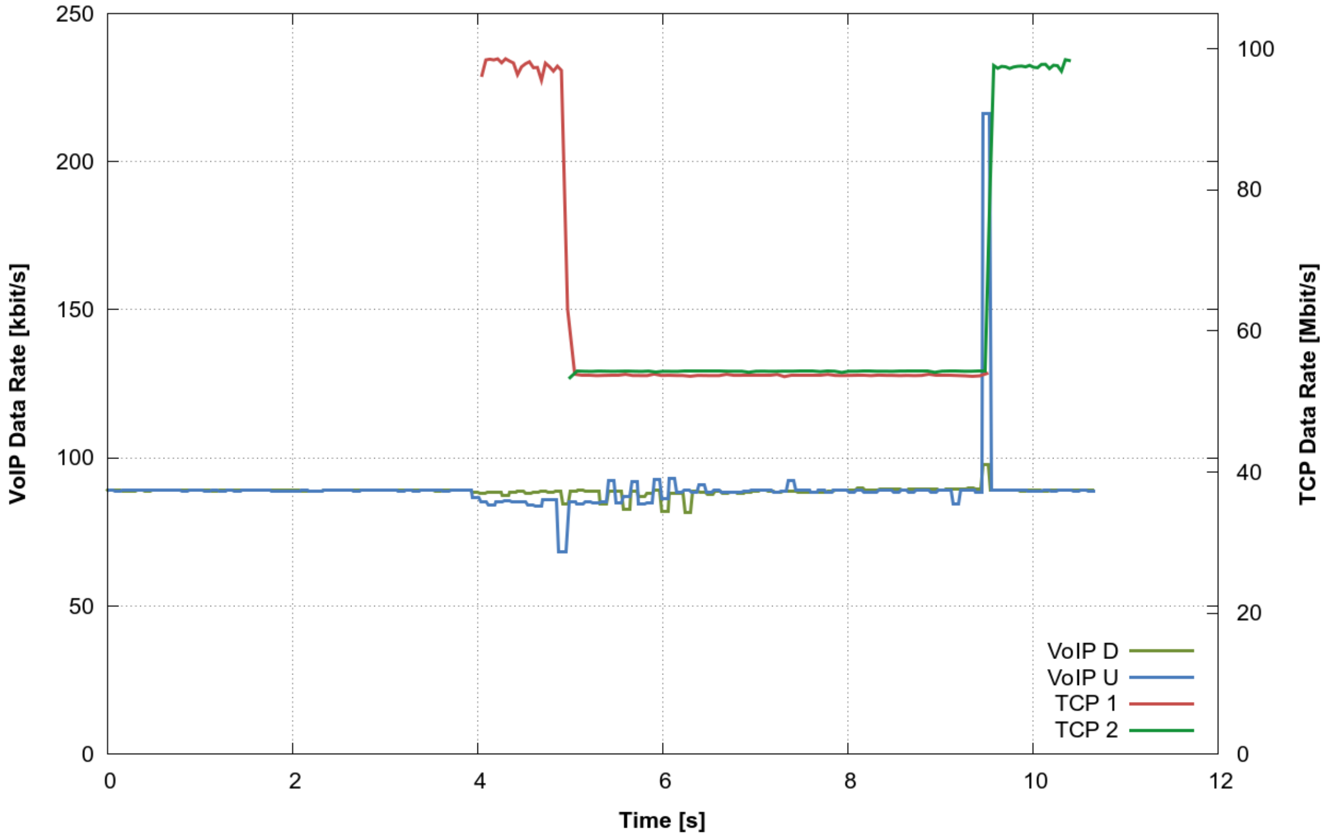
- **Transfer Type** one-shot, periodic
- **Period Length** (periodic case only)
- **Offset** communication startup delay
- **Length** amount of payload data
- **Transport Protocol** TCP, UDP or ICMP
- **Targets** list of receiver nodes

Using these profiles as building blocks it is possible to model a large number of use cases that have been identified in a prior study, including those shown in table 1.

Application	**periodic**	**Transport Protocol**	**Payload Size**
HTTP Download	no	TCP	big
Mail Transfer	no	TCP	varying
VoIP Call	yes	UDP	small
A/V Stream	yes	UDP	big
VPN Download	no	UDP	big
Latency Measurement	yes	ICMP	small

They also allow for modeling various communication topologies, including Unicast (one sender, one receiver), Multicast (one sender, multiple receivers) and Concast (multiple senders, one receiver). The two latter options are required when thinking of automotive passengers accessing Internet functionality through an in-vehicle mobile broadband gateway connected over MOST Ethernet. The ability to instantiate multiple generator modules using the same profile allows for the simulation of multiple applications at the same time. After executing the measurement defined in the test case description, FZI performs an off-line analysis in the trace sent to the control PC. In this analysis, performed by an automated sub-part of the framework, they determine data rate, transfer volume, packet loss and latency over time.

Figure 2 shows an example plot of transfer data rates over time, collected from a test case aimed at modeling the concurrent transfer of two files over a gateway (TCP0, TCP1) while a VoIP call (UDP0, UDP1) is in progress. One can see that the initially constant data rate of the VoIP flows experiences a short period of fluctuation as the first file transfer is initiated, but recovers quickly. Furthermore, it can be seen that the two TCP connections share their part of the available bandwidth almost equally while running in parallel, and make use of the entire available bandwidth left over from the UDP flow when running exclusively.



From this behavior, the authors conclude that both the bus arbitration mechanisms and the packet scheduling mechanisms of the operating system running on the nodes work effectively. Furthermore, flow control and traffic shaping mechanisms implemented by the TCP/IP stack seem to be working on the virtual MOST Ethernet device without any modification of parameters. The fact that the network interface used by the traffic generators is virtual and transmits data over the MOST Asynchronous Channel seems to have no impact on socket communication, allowing applications to be brought into the automotive domain without modification of the communication primitives. Apart from two corner cases with modified MOST bus parameters, the service provided by MOST Ethernet, from a user perspective, can be seen as a replacement for Fast Ethernet.

In order to allow for the evaluation of larger networks without the need for possessing the according number of developer boards, the team designed their evaluation framework to be easily embeddable in simulation environments, especially those based on SystemC. For this purpose they have implemented a set of SystemC modules that encapsulate the functionality of the previously described software modules running on the nodes, providing an interface to the simulated MOST bus. The software running on the controlling PC could even be kept unmodified, as it can communicate with the simulated nodes over the *loop-back* interface of the simulation host.

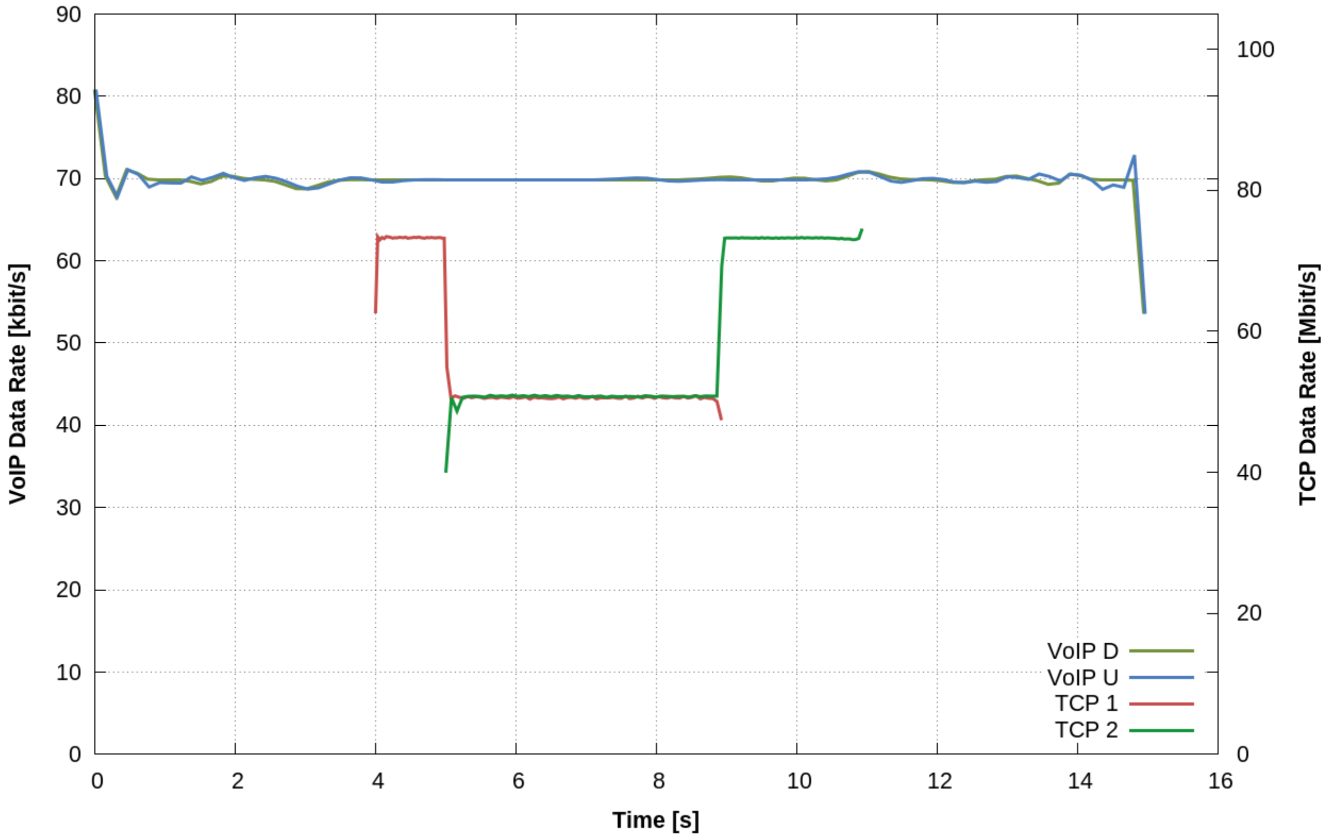


Figure 3 shows the data rate evaluation plot resulting from the analysis of a virtual measurement run equivalent to the example test case described earlier, in analogy to figure 2. It can be seen that the simulation, compared to the physical run, shows a constant offset of the achieved data rate, resulting from some minor shortcomings of the bus model. Nevertheless, it shows all the traffic characteristics depicted in figure 2, especially the TCP bandwidth sharing effects.

Conclusion

The team at FZI has designed and implemented a framework for the evaluation of user applications performing socket communication that allows for the evaluation of various communication properties, including bandwidth and latency. Applied to MOST Ethernet, it showed that common applications using these communication abstractions can be brought to the automotive domain without modification of the communication interface, and that MOST Ethernet, from an application perspective, can serve as a replacement for Fast Ethernet. Using Virtual Prototyping Technology, FZI's framework provides a highly flexible basis for large-scale analysis without the limitations of possessing the according amount of physical hardware.

Authors:

Prof. Dr. Oliver Bringmann is professor at the Wilhelm-Schickard-Institute for Computer Science at the University of Tuebingen, Germany.

bringman@informatik.uni-tuebingen.de

Dipl.-Inform. Alexander Koch works as a research scientist in the department Intelligent Systems and Production Engineering (ISPE/SiM) at the FZI Research Center for Information Technology in Karlsruhe, Germany.

Dipl.-Inform. Sebastian Reiter works as a research scientist in the department Microelectronic System Design (SiM) at the FZI Research Center for Information Technology in Karlsruhe, Germany.

sreiter@fzi.de

Prof. Dr. Wolfgang Rosenstiel is professor at the Wilhelm-Schickard-Institute for Computer Science at the University of Tuebingen and director at the FZI in Karlsruhe, Germany.

rosenstiel@informatik.uni-tuebingen.de

Dr. rer. nat. Alexander Viehl is department manager of the department SiM at the FZI in Karlsruhe, Germany.

viehl@fzi.de