

Requirements on Real-Time-Capable Automotive Ethernet Architectures

Philip Axer, Daniel Thiele, and Rolf Ernst
Technical Univ of Braunschweig

Jonas Diemer, Simon Schliecker, and Kai R. Richter
Symtavision GmbH

ABSTRACT

Ethernet is the hottest candidate for future in-car communication architecture, promising much higher bandwidth, flexibility and reduced costs. In the coming years, Ethernet will likely evolve from a separate communication medium for special applications like surround-view cameras and infotainment to a central communication infrastructure as a backbone technology. To make this transition, many difficult design decisions have to be made in order to make the technology suitable for the stricter time and safety requirements of today's and future cars. There are a lot of potential real-time effects that must be taken into account.

To guide these design decisions, it is necessary to analyze the various architecture concepts with respect to load, performance and real-time capabilities. In this paper, we present different design space axes of Ethernet and propose a methodology of assessing and comparing them. This includes a formal worst-case timing analysis approach and a set of metrics that make the timing analysis results comprehensible and comparable across different design options. Using this foundation, we explore the design space for a typical automotive scenario. Specifically, we compare different topologies and switching strategies and their effect on the performance of the network. This provides valuable insights and design guidelines, but also shows the need for further exploration and reduction of the design space by deeper standards for the automotive industry.

CITATION: Axer, P., Thiele, D., Ernst, R., Diemer, J. et al., "Requirements on Real-Time-Capable Automotive Ethernet Architectures," *SAE Int. J. Passeng. Cars – Electron. Electr. Syst.* 7(2):2014, doi:10.4271/2014-01-0245.

INTRODUCTION

Ethernet is currently added to the existing set of automotive communication standards. Due to its superior bandwidth and flexibility and the promise of sharing cost of ownership with other industrial segments, Ethernet appears ideal to address the high demands of new functions in infotainment and advanced driver assistance systems (stereo camera, surround view, etc.) or to reduce ECU flashing and updating cost. While the first generation of cars will use Ethernet as additional communication medium, it is strongly considered as a powerful future backbone technology also carrying traffic originating in CAN(-FD) or other bus subsystems. An exemplary Ethernet backbone use-case is depicted in Figure 1.

This is no easy task since Ethernet has been developed for applications which are neither time nor safety critical, but focus on bandwidth. Cars have a far richer set of real-time requirements including quality-of-service (QoS), guaranteed end-to-end timing (i.e. latency), guaranteed delivery, guaranteed bandwidth or best-effort. Furthermore, Ethernet differs from CAN(-FD) and FlexRay in many ways: Ethernet is

a packet switched network with point-to-point links and switches, which add delays to the end-to-end latency. Ethernet only has (at most) 8 priority levels compared to 2^{11} / 2^{29} on CAN. Frames can even be dropped in the switches without any ECU noticing it, requiring countermeasures on higher protocol layers. There are a lot of potential real-time effects that must be taken into account.

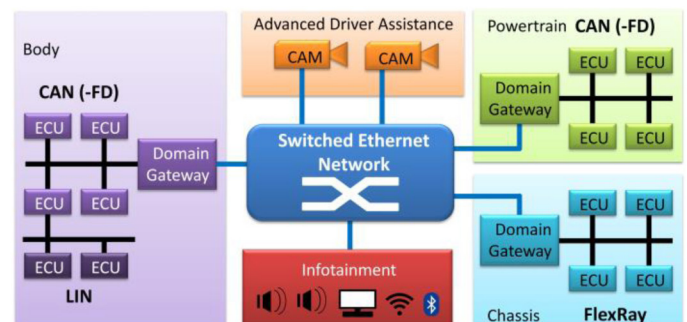


Figure 1. Typical Ethernet backbone architecture

Being able to analyze architecture concepts, load, performance and real-time capabilities of such Ethernet networks is one of the top-priority requirements. Such analyses are established standards for CAN and FlexRay networks already today, and we will need them more urgently for Ethernet-based E/E architectures because the real-time behavior will become more complex. At present, first solutions exist but there is only little experience with Ethernet-in-the-car, the existing standards do not cover all relevant aspects, and we can expect several “explosive issues” under the hood. Furthermore, the design space for Ethernet is (still) huge, which could lead to OEM-specific solutions that drive costs on all sides with no added value. To avoid this, a carefully planned joint approach would be highly beneficial for the automotive industry.

First solutions exist, showing that we can technically handle it. But what is price? On key aspect is that currently, there are very many variants of automotive Ethernet under discussion; this increases the total cost of ownership because someone has to provide the necessary equipment: switches, transceivers, lab equipment, tools, etc. When variants go up, re-use goes down, and prices increase. Hence, we need “deeper” industry-wide standards as soon as possible. Only then, we can reduce cost for components, processes, and tools. We expect an increasing awareness of this issue among experts and decision makers at the car manufacturers world-wide.

The paper is focuses on the issues on ISO/OSI layer 2 and is organized as follows: The next section provides some technical background and identifies key dimensions of the overall Ethernet design space: topology, protocol, and switching options. Then, we explain, how such Ethernet designs can be analyzed with respect to timing. We summarize key analysis techniques, and we present practically relevant metrics for the representation of the results. Next, we apply these analysis and metrics to an example Ethernet network and we compare the different strategies. Finally, we summarize lessons learned and draw our conclusions.

ETHERNET BACKGROUND

Topology Design Space

In this section, we will look into the topology design space, explain key mechanisms, possible pitfalls, and their influence on the overall performance.

Traditional busses like CAN or FlexRay are shared busses where many nodes are connected to the same physical wire. Switched Ethernet, on the other hand, can only connect two nodes directly in a peer-to-peer fashion. It must be noted, that Switched Ethernet is fundamentally different to traditional Ethernet which uses a shared medium (i.e. shared coaxial cable) and CSMA/CD for arbitration. Today, the concept of Ethernet over a shared medium is considered as obsolete.

Automotive Ethernet networks are established by connecting several ECUs to one another with one or more switches in between. This enables many different topologies, see [Figure 2](#). A thorough analysis is required to choose the best fitting topology for a network. In this section four commonly used topologies are discussed.

The *star topology*, where all ECUs are connected to one big central switch, is well-known from office Ethernet networks. This topology only requires a single switch and reduces network contention, by minimizing shared links. On the other hand, a star topology can result in expensive wiring costs, depending on the location of the ECUs.

The *line topology* strives to reduce overall cable length. Here, each ECU comprises a 3-port switch to which it is connected as well. This increases the cost of the ECUs as each unit must contain a switch, and complicates the network configuration. Most importantly, contention increases as links are shared between different ECUs.

Clustered topologies represent a mixture between star and line, allowing a trade-off of the individual strengths and weaknesses. Typically, ECUs exchange most of the data locally in their system domains (i.e. body control). Thus, to reduce contention, one switch per domain is introduced. This reduces latency for intra-domain communication without excessive wiring cost.

In case the number of ECUs exceeds the number of ports per switch, a *tree Ethernet* topology can be chosen. This reduces the number of ports per switch. Special care must be taken with respect of the depth of the tree and location of ECUs in this topology as the latency strongly depends on the number of hops in the network. Note that in most automotive networks, ECUs will integrate the switch to decrease the number of physical nodes. In this sense [Figure 2](#) shows the logical topology and not the physical setup.

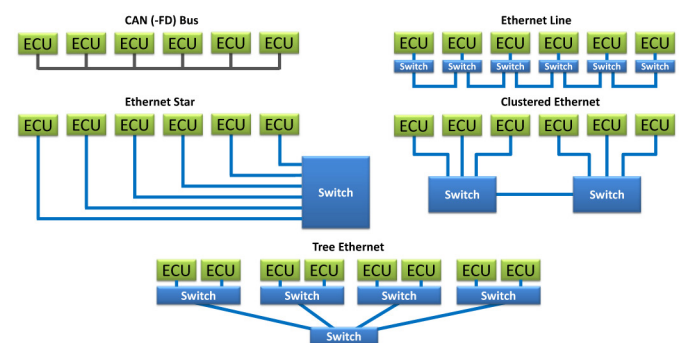


Figure 2. Typical topology candidates

Switching Strategy

The Ethernet switch obviously is the most important part of the entire network design. The switch receives frames on its ingress ports and forwards them to the proper egress ports. This is conceptually shown in [Figure 3](#).

There are two sources of delays: The static switching delay is caused by the store and forward architecture. An Ethernet frame is fully received, written to the output buffer(s) of the destination output port(s) and then transmitted. Static delays can be considered fairly easy and are mostly in the order of a few clock cycles of the switch (i.e. mostly internal memory access). The dynamic queuing delay occurs when frames arrive faster than they can be forwarded. This typically happens if frames arrive on multiple ingress ports which must be forwarded to the same egress port. The queuing delay is complex to predict and strongly depends on the interfering traffic pattern, used scheduling policies, shaper settings and available buffer sizes.

The switch pipeline consists of two units: an internal switch fabric and the egress stage. The switch fabric contains the address resolution logic, which resolved the output port from the frame's MAC address and forwards the frame to the correct egress port. The egress stage contains one or multiple FIFO queues, one for each service class, as well as the output scheduling logic. If frames are pending on multiple queues, the scheduler decides on the order of transmission.

The original Ethernet standard (IEEE 802.3), initially indented for best-effort transmission, does not specify any particular scheduling policy, but standard implementations typically implement one FIFO queue per port. Such an implementation makes it difficult or even impossible to make any timing predictions for different traffic classes. Because queues are shared, misbehaving or bursty streams easily over-occupy the buffer space and lead to dropped frames (tail-drop). This can significantly degrade the performance and reliability.

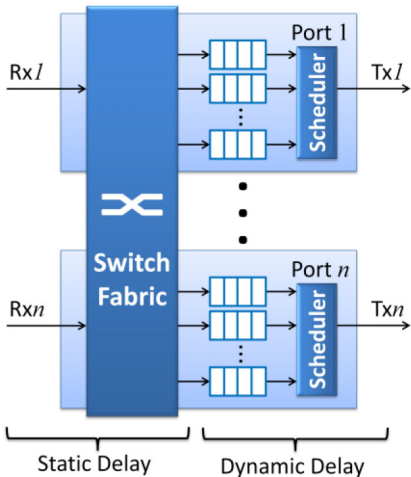


Figure 3. Internal switch architecture.

IEEE 802.1Q VLAN adds the concept of traffic classes and the switches can have up to 8 separate FIFO queues per port, each with a distinct *strict priority* (SP). In such switches, higher-priority traffic has always precedence over lower-priority traffic, just like on CAN buses. However, CAN has 2¹¹ (base) or 2²⁹ (extended) priority levels, while VLAN supports a maximum of 8 priority levels which requires merging the fine grain priority

structure of CAN onto few separate communication streams. This entails a lot of potential real-time effects (e.g. latency) that must be taken into account. Also note, that most switches do not implement all 8 priority levels due to the required memory footprint, this leaves even less design freedom.

The recent *Ethernet AVB* Standard [1] extends the strict-priority scheduler by traffic shapers in the egress stage that facilitate bandwidth reservation. This way, higher-priority traffic classes can be prevented from blocking lower priority traffic for a long duration, while still meeting their bandwidth constraints.

Other switches provide a *weighted round-robin* (WRR) policy [2], where bandwidth is distributed according to weights that are associated with each traffic class. *Time-Triggered Ethernet* (TTEthernet/SAE AS6802) [3] uses a different approach based on a time-triggered schedule that is globally synchronized. It thereby strives to avoid collisions by design instead of resolving them on the fly. Here (just as with FlexRay), the guaranteed latency times come at the cost of reduced flexibility and efficiency. Finally, there is a new standard on the horizon: *AVB Generation 2* or also called *time-sensitive networks* (TSN). This standard borrows concepts from the existing ones to combine the “best of all worlds”, eventually resulting in a real improvement - if done right, and not too late, and not too costly.

Frame Forwarding Logic

Each incoming frame must be forwarded to one or multiple output ports. The Ethernet frame does not contain any intrinsic route information that can be used by switches for the forwarding process, as it only contains a source and a destination MAC (Media Access Control) address. Thus, either a switch dynamically learns the topology by listening to the traffic or it must be statically configured during design time. The possible number of MAC addresses (2⁴⁸) is too large to provide forwarding entries for all possible addresses. Thus a caching technique is used in which destination addresses compete for a limited number of physically available forwarding entries. Forwarding entries (cf. Figure 4) are organized in a Content Addressable Memory (CAM) hardware structure. The number of entries is in the order of a few thousand entries. A consumer grade unmanaged five-port switch (BCM5325) supports up to 2k entries according to the product brief.

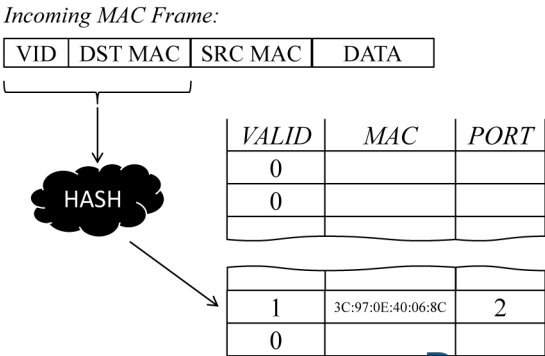


Figure 4. Forwarding logic: CAM table

Documents
Document Sharing Hub