Ethernet is a driver of innovation in vehicles. The two main reasons for this are the available bandwidth as well as the range of functions and the flexibility of this network technology. Based on these premises, this is a good time to take a closer look at three technological trends: The modeling of service-oriented architectures and Automotive Ethernet networks; the use of Automotive Ethernet in POSIX-based ECUs; and reliable communication using Time Sensitive Networking.

The growing use of Ethernet in vehicles is resulting in a number of factors which influence one another. Automotive requirements encourage technological innovations and advances, for example mechanisms for the transfer of time-critical data (Time Sensitive Networking). At the same time, Ethernet is permitting changes to the E/E structure in vehicles and is making it easier to use service-oriented architectures.

What is happening in the field of Automotive Ethernet? In the IT industry, service-oriented architectures (SOA) have already been used for many years to describe and structure distributed systems. However, service-oriented design is now also growing enormously in importance in the automotive industry.
Modeling of Service-Oriented Architectures and Automotive Ethernet Networks

An SOA creates a service landscape that is characterized by clearly defined interfaces (Figure 1). Ideally, these interfaces should be correctly described both syntactically and semantically. In addition, in an SOA, the software components are connected via middleware known as a “service bus”. The middleware mediates between the two roles that participate in a service: the service provider and the service consumer. The SOA provides a loose coupling between the service roles in a system. It is therefore more flexible than monolithic software architectures. Two other positive aspects are the increased reusability and better ease-of-integration of new service components. The middleware regulates communication between service provider and service consumer as well as the establishment of this communication. It also usually defines the serialization of data in the Ethernet frame. The serializer determines how the data is converted into a serial bit stream and is then deserialized again after reception.

Nowadays, there are various description languages for interfaces, known as interface description languages (IDL). These typically have technology-independent components as a formal description of a service. In addition, they can also have protocol-specific elements. Technology-independent properties include available methods, fields and events, including the data types used. The middleware-specific startup behavior, on the other hand, can be regarded as a technology-dependent characteristic. These include service discovery (SD) and the use of data serialization and protocol-specific identifiers (ID). IDLs are particularly well suited for the formal description of individual interfaces and automatic code generation. They are mostly present in hierarchically structured text files. However, text files are not particularly well suited for designing complete SOAs. As a result, model-based design tools are becoming more popular. One strength of modeling is that the data is managed centrally and that the maintenance effort is therefore greatly reduced. Model-based design also guarantees that the data is formally correct at the time of design. The clear definition also makes it possible to structure the services. In addition, larger systems usually use layer architectures that make the dependencies of services understandable and represent clear design rules. When services build on each other, the necessary design is called “service choreography”. In a model-based outline design, the data communication can be partially derived based on the modeling of the service. At the same time, model-based design makes it possible to manage complexity considerably better than in the IDL-based approach.

The use of an SOA permits the dynamic establishment of communication relations at runtime. However, in current automotive systems that already use SOA, service communication is often still statically designed due to limited resources. Thanks to the use of middleware, the connection between the service provider and service consumer is established at runtime – and not at the time the system is designed, as in the past. In the future, we can assume that fully-dynamic service communication will predominate, especially in systems with high-performance hardware. Information and driver assistance systems, in particular, are driving this trend forward. External communication with online map services or OTA (over-the-air) updates is also supporting this trend. Service communication requires much more protocol information at the network level than a signal-based approach. Due to the required bandwidth, Ethernet is therefore the main network technology in the vehicle.

Automotive Ethernet in POSIX-Based ECUs

The modeling of service-oriented E/E architectures and the lower-level Ethernet networks as described in the section above is already relevant for many ECUs of different performance classes. The spectrum extends from intelligent sensors such as camera, radar and LIDAR systems through to central high-performance computers that implement sensor fusion and the higher-level driver assistance applications. ECUs in the lower and medium performance classes are often based on a conventional AUTOSAR software architecture.

However, other operating systems are necessary for high-performance ECUs. On the one hand, support is needed for the corresponding hardware accelerators, for example for graphical units. On the other hand, there is a need for greater flexibility in application handling and the updating of data. This is why POSIX-based operating systems are mostly used in this field. The best-known software solutions in the automotive environment include Linux – for safety-relevant applications under Linux, a hypervisor such as BSD socket API and adapted Ethernet driver.

Figure 2: Use of AUTOSAR Classic modules in ECUs with POSIX operating system
as Sysgo’s Pike-OS is also necessary – and Integrity from Green Hills Software and QNX. However, these operating systems do not provide any integrated support for Automotive Ethernet. To meet automotive requirements, a number of specific protocols based on Ethernet and TCP/IP have been defined for vehicle use. These include Scalable Service-oriented Middleware over IP (SOME/IP) for efficient, service-oriented communication, Diagnostic Communication over IP (DoIP) for vehicle diagnoses, User Datagram Protocol-based Network Management (UDPNM), Universal Measurement and Calibration Protocol (XCP) for measurement and calibration tasks, as well as time synchronization as specified in IEEE 802.1AS. Unlike AUTOSAR Classic, modern POSIX systems have no native support for these protocols. However, they do support subsequent integration.

One possibility is to execute proven AUTOSAR Classic modules in a POSIX process (Figure 2). In AUTOSAR Classic, SOME/IP-SD, DoIP, UDPNM, XCP and conventional signal- and PDU-based communication (PDU: Protocol Data Unit) do not directly access the AUTOSAR TCP/IP sockets because these are encapsulated by the Socket Adaptor Module. However, if the socket adaptor uses Berkeley sockets (BSDs) instead of AUTOSAR sockets, then this module – and therefore also the above-mentioned automotive-specific protocols – can be executed within a POSIX process. If we go a step further and also emulate the AUTOSAR operating system, then it is even possible to build a scalable Ethernet network. To do this, hardware resources in the Ethernet controller. TSN makes it possible to build a scalable Ethernet network. To do this, various message classes are scaled in terms of availability, and their latency and priority are categorized. Each message class is assigned a guaranteed bandwidth. In addition, redundant Ethernet systems are supported and security criteria are defined to ensure stable data exchange. The IEEE Audio/Video Bridging Task Group defined mechanisms and protocols to ensure low-latency data exchange and synchronize applications in terms of time. Audio/Video Bridging (AVB) is mainly aimed at infotainment applications. The widespread introduction of driver assistance systems, on the other hand, calls for even stricter rules with corresponding AUTOSAR Classic modules allow the implementation and use of the automotive-specific layer 2 protocols in a POSIX process.

A second possibility is to use AUTOSAR Adaptive. This is a runtime environment that has been standardized for POSIX operating systems and that will be used for many innovative automotive applications in the future. AUTOSAR Adaptive provides automotive-specific extensions for fully dynamic POSIX applications. In the field of Ethernet communication, these are presently SOME/IP and DoIP. Work is currently being undertaken to support further protocols. The BSD sockets of the POSIX operating system are also used for TCP/IP-based communication, thus simplifying integration. For the support of layer 2 protocols in the context of time-sensitive networking (TSN), OS-specific adaptations may be necessary in AUTOSAR Adaptive, since some additional information required is not available via the standard POSIX interfaces. The question of which of the above variants is most suitable for an ECU project depends on various factors. For example, if it is possible to call on existing AUTOSAR software components (SWCs) or diagnostic implementations, then one good solution may be to use conventional AUTOSAR modules under POSIX. However, if the full flexibility of native POSIX applications is needed then AUTOSAR Adaptive is recommended.

Reliable Communication Using Time-Sensitive Networking (TSN)
In vehicles, real-time capabilities are vital for certain applications (Figure 3). To ensure that these are available, it is necessary to have mechanisms that directly access the hardware resources in the Ethernet controller. TSN makes it possible to build a scalable Ethernet network. To do this, various message classes are scaled in terms of availability, and their latency and priority are categorized. Each message class is assigned a guaranteed bandwidth. In addition, redundant Ethernet systems are supported and security criteria are defined to ensure stable data exchange.

The IEEE Audio/Video Bridging Task Group defined mechanisms and protocols to ensure low-latency data exchange and synchronize applications in terms of time. Audio/Video Bridging (AVB) is mainly aimed at infotainment applications. The widespread introduction of driver assistance systems, on the other hand, calls for even stricter rules with

![Figure 3: TSN is extending Automotive Ethernet for time-critical data in the areas of scalability, reliability and timing guarantees.](image3)

![Figure 4: The TSN bridge only passes on data streams that are registered with it.](image4)
regard to sending and receiving behavior. Therefore, the IEEE Time-Sensitive Networking Task Group continues the work of the Audio/Video Bridging Task Group. The focus of this working group is on deterministic data transmission, further latency reduction and stable and secure transmission in Ethernet networks (Figure 3).

Time synchronization using gPTP as described in IEEE 802.1AS ("Timing and Synchronization for Time-Sensitive Applications") is essential to ensure that Ethernet networks with time-based data analysis are stable. In this process, a Grand-Master sends a global time to all time-sensitive ECUs. In addition, the average runtime of a defined Ethernet frame is measured and taken into account when calculating the synchronized time. In this way, all time-sensitive ECUs can synchronize to a discrepancy of no more than one microsecond.

The sender and recipient then assign this synchronized time to the individual data packages. The data is distributed to the endpoints using a specific send procedure (Shaper). During this procedure, the credit-based shaper that is known from audio/video bridging ensures that the data arrives at the recipient within a defined time (presentation time). By contrast, the time-aware shaper added by TSN ensures that the data is deterministically sent in accordance with a fixed, predefined time grid. In particular, this must be ensured by every switch. For a specific portion of the bandwidth, this results in a time-driven network, similar to the case of FlexRay. The two shaping procedures can coexist without problems in parallel with the regular Ethernet traffic in a network. This is an advantage compared with FlexRay.

TSN also supports mechanisms to favor high-priority Ethernet packets over low-priority Ethernet packets that are currently being sent without waiting for their complete transmission. For this purpose, the latter are preempted at the next possible point in time in favor of the Ethernet packet with higher priority and continued later. This method reduces the maximum latency even further without affecting the rest of the transmission behavior. The Stream Reservation Protocol (Figure 4) is used to monitor the bandwidth. In this protocol, the TSN bridge checks that only data streams that are registered with it are passed on. At the same time, the current message volume for a message class must be in a previously defined range. If this budget has been used up, this data stream cannot be sent at the present time.

In the case of safety-relevant systems, it is also possible to duplicate the packages for redundant transmission over separate network paths. In addition, filter and control mechanisms ensure that only message classes that are actually expected are passed on and that this only takes place with the required frequency. Both increase the robustness of Ethernet communication in the event of an error - whether due to interrupted network structures or unwanted Ethernet traffic.

Many of the TSN mechanisms require hardware support (Figure 5). The TSN specification IEEE 802.1Qbv ("Enhancements for Scheduled Traffic"), which is now stable, is playing a leading role and is already supported by a wide range of hardware. By contrast, in the case of IEEE 802.1Qbu ("Frame Preemption"), it remains to be seen how the interrupt mechanisms on Ethernet will establish themselves. In Gigabit Ethernet, the shorter wait times through to the transmission of a high-priority frame will hardly make any difference. Instead, this will more likely have a significant influence on 10/100 megabit systems.
Outlook

In future vehicle platforms, the dynamic protocols and methods that are familiar from IT are becoming fused with conventional automotive requirements in terms of time response, availability and robustness. Service-oriented communication and safety-relevant and deterministic Ethernet traffic are merging to form complex systems. On the one hand, at design time there is no binding specification about the ECU on which a function is to be implemented and the runtime environment in which it is to be executed. On the other, the network architecture must guarantee reliable communication that meets the needs of the application. With its embedded software, Vector is already able to meet these requirements today. What is more, Vector’s Ethernet design and test tools provide ways of representing and testing dynamic, service-oriented and time-sensitive communication relations. Thanks to Vector’s active work on various boards, the market also benefits from the early availability of solutions for future trends in the automotive industry.

Translation of a German publication in magazine Automobil-Elektronik, issue 11-12/December 2017 (unabridged version)

Image rights: Vector Informatik GmbH