New Communication Paradigms in Automotive Networking

Ethernet and CAN FD are the new trailblazers

CAN is the prevailing bus system for communication between electronic control units (ECUs) in today's vehicles. The communication demand has increased dramatically in recent years, and vehicle manufacturers are now reaching the limits of vehicle networking with CAN. Ethernet and CAN FD provide higher bandwidth and are taking over some of the tasks of existing bus systems. The higher bandwidth is one aspect, but also of high relevance is the introduction of new communication paradigms.

A classic CAN message transmits a payload of up to eight data bytes. For reasons of efficiency, it is advantageous to use all eight bytes to maintain the best possible ratio between payload data and protocol overhead. However, the individual data elements (signals) being transmitted, such as wheel speed, are often less than eight bytes in length. Several signals are therefore sent together. Each bit is valuable, and thus the task of defining each CAN message with the signals contained therein becomes very complex.

The communication matrix is defined in a database. For CAN this is done in one of the following formats: DBC, FIBEX, or AUTOSAR System Description. The database (Figure 1) contains not only the messages and their composition but also the associated send and receive relationships. Also defined in the database are the Protocol Data Units (PDU) – an abstraction layer between signals and messages.

CAN FD, the improved CAN bus

The maximum transmission rate of 1 Mbps on the CAN bus is no longer sufficient for all of today's communication demands. A solution to the bandwidth problem is to use CAN with flexible data rate (CAN FD). This enhanced version of the CAN protocol features transmission rates of up to 8 Mbps. This is achieved by two enhancements to classic CAN:

> 1. The payload data bytes are transmitted at a higher bit rate. In order for the other properties of CAN, such as the maximum cable length, to remain the same as possible, the header and trailer of a CAN message are sent at the normal bit rate.
> 2. A CAN FD message contains up to 64 payload data bytes. If an eight times faster bit rate is used and 64 payload data bytes are sent, the transmission time is comparable to a classic CAN message with 8 payload data bytes.

With CAN FD, the previously mentioned complexity of defining the communication matrix is significantly increased once again. To ensure efficient communication in this case, network designers must logically group signals in messages that are up to 8 times larger. Some scenarios also require that PDUs defined previously for CAN are available on CAN FD as well. This would eliminate the advantage of the greater payload data length. Assistance is provided by the n-PDU-to-frame mapping, which allows the sending of multiple PDUs in one message.

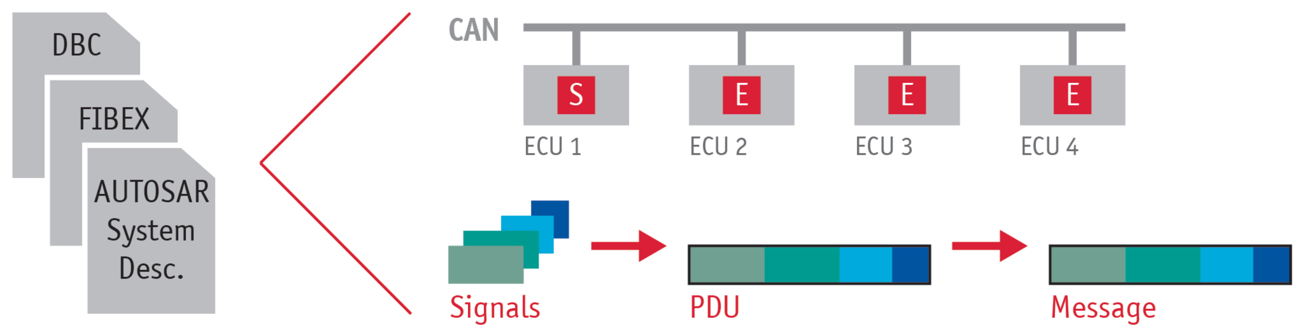


Figure 1: A database contains both system and data descriptions

Exemplary n-PDU-to-frame mapping in a gateway scenario

In a gateway scenario (Figure 2), the bandwidth of a classic CAN bus may no longer be sufficient. Let's assume that three CAN buses are connected to a gateway and all are already being operated at the bus load limit. An ECU that requires much of the data of the other two buses is attached to CAN bus 3. The gateway is responsible for routing the needed data to the ECU. If, due to a change of generation, the ECU on CAN bus 3 has a need for additional communication, the bandwidth of CAN bus 3 is no longer sufficient. CAN FD should therefore be used instead. In addition, there are further requirements:

> Use of all 64 payload data bytes for efficient bus utilization.
> The previously defined PDUs must not be changed.
> The gateway must not become significantly more complex.

All of these requirements can be met by the sending of multiple previously-defined CAN PDUs in one CAN FD message by the gateway.

Up to now, the content of a message was identified by the respective CAN identifier (CAN ID). The receiver can no longer use the CAN ID for data identification since a CAN FD message contains multiple PDUs. A possible solution would be to define the content of the CAN FD message statically in a database. In contrast, with n-PDU-to-frame mapping, the database defines the PDUs that can potentially be contained in the CAN FD message. The PDUs that the ECU actually sends are chosen during runtime. The described scenario is presented schematically in Figure 2. Each ECU on CAN bus 1 and CAN bus 2 as well as the gateway itself sends a PDU that is to be transmitted on the CAN FD bus. The gateway fills up the CAN FD message step-by-step with the PDUs up to the time of sending. A small header is put in front of each PDU so that a receiver can extract the individual PDUs from the message. The gateway no longer has to pay attention to the order of the PDUs and is spared the complex task of keeping the defined message layout. This keeps the demand for resources on the gateway as low as possible.

The sending of a CAN FD message is triggered by various events, such as:

> 1. The send buffer of the message is full.
> 2. The timeout defined for the complete message expires.

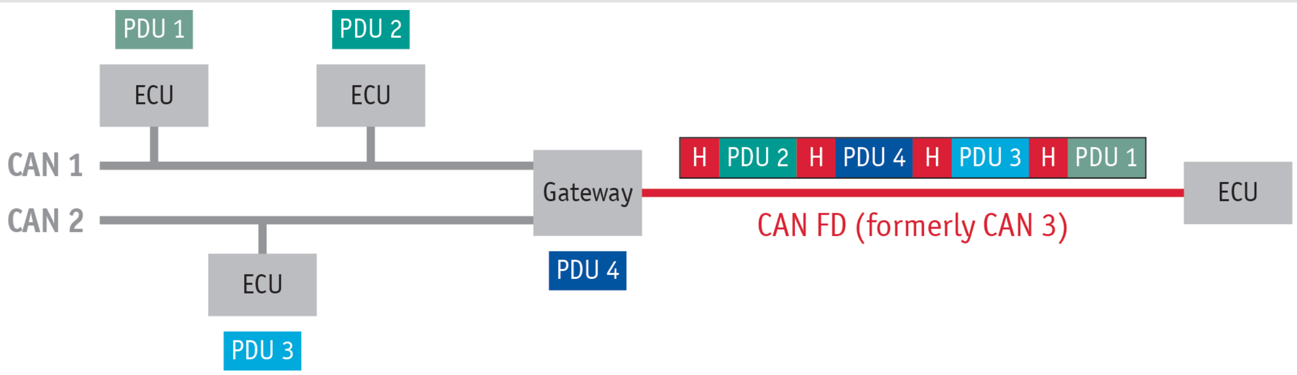


Figure 2: Possible scenario for introduction of CAN FD

> 3. After expiration of the timeout defined for a single PDU, the message containing this PDU is sent.
> 4. A PDU has an additional attribute that triggers immediate sending of the message after the PDU is copied to the send buffer.

The option of sending multiple PDUs in one message has been defined in AUTOSAR since release 4.2.1. The mechanism is specified independent of the network technology in the I-PDU Multiplexer module so that it can be used, for example, for FlexRay or Ethernet in addition to CAN FD. Besides the pure collection of PDUs, the module also supports the different sending conditions and two different PDU header formats. For CAN FD and FlexRay, a 4-byte header is primarily used while an 8-byte header is generally used on Ethernet.

Ethernet provides a much larger payload data length

Compared to CAN, Ethernet provides a maximum payload data length that is more than 185 times larger. A classic definition of hundreds or even thousands of signals within a 1500-byte PDU is not really feasible. For gateway applications, however, it can be advantageous to forward existing CAN or FlexRay PDUs 1:1 on Ethernet. The sending of multiple PDUs within one message, which was described for CAN FD, can be used for Ethernet unchanged. In contrast to CAN FD and FlexRay, AUTOSAR specifies two equivalent approaches for Ethernet. The first approach is the n-PDU-to-frame mapping in the I-PDU Multiplexer, which has already been described for CAN FD. The same PDU collection algorithm is specified in the Socket Adaptor module, which is responsible for linking the TCP/IP stack to the rest of the AUTOSAR architecture. The user can use both approaches. When the Socket Adaptor is used, additional control possibilities for Ethernet-based communication are available.

Different network topology with Ethernet

Besides the payload data length, Ethernet differs significantly from other network technologies in regard to network topology and addressing (Figure 3). CAN uses a classical bus topology while FlexRay can be physically implemented with a star topology, from a logical standpoint it behaves like a bus, too. In the case of both network technologies, a message is sent to all nodes. Each network node decides independently whether a message is relevant to it and will be further processed. This is done using the CAN ID for CAN and the Slot ID for FlexRay. In either case, the ID already classifies the content of the message. There is no option of selectively sending a message to only one receiver. For this reason, CAN and FlexRay are broadcast media. At the beginning of its success story, Ethernet was also a bus system. The cabling was implemented with coaxial cables and T-elements. Today, Ethernet is, in the vast majority of cases, an actively switched network with tree topology in which collisions cannot occur due to point-to-point connections.

Every Ethernet node has a MAC address, which is used for unique addressing in the network. A node processes a message when its MAC address is specified as the destination. In other words, the sender must know the receiver and enter the corresponding destination address in the message. This 1:1

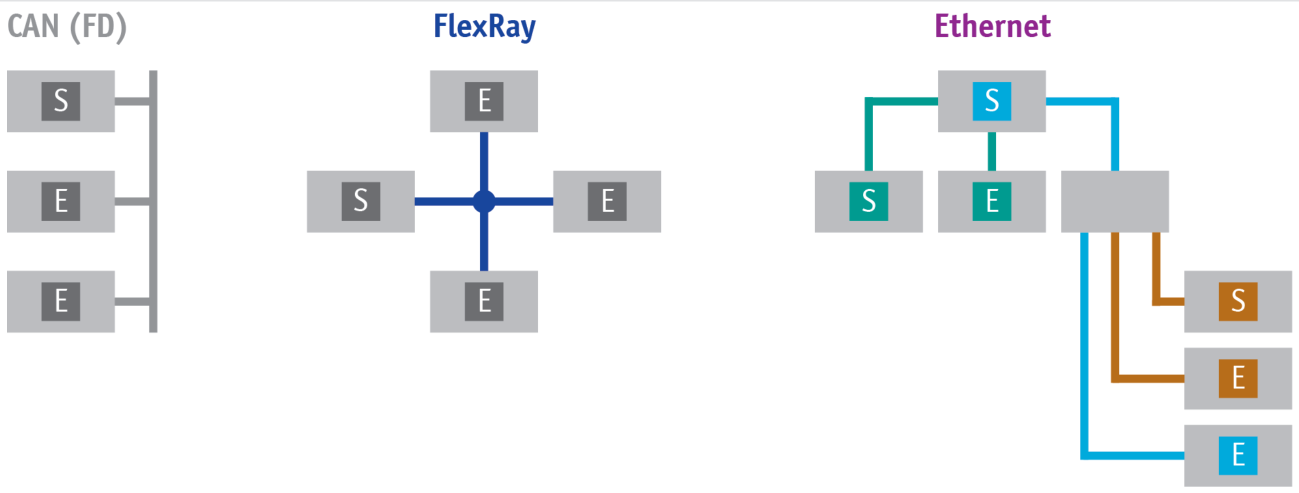


Figure 3: Comparison between network topologies of CAN (FD), FlexRay and Ethernet

communication relationship uses so-called unicast addressing. If a unicast-addressed message is sent to all network nodes, only one receiver processes the packet. All other receivers discard it. To prevent flooding the network unnecessarily, the switch as an active network connection element was introduced. After a brief learning phase, a switch forwards messages only to the connection where the addressed destination can be reached. The available bandwidth is thus used efficiently. Furthermore, a switch enables multiple unicast-addressed messages to be sent simultaneously on the network. Figure 3 shows an example Ethernet network with two switches: One in the ECU at the top of the figure, the second in the unlabeled ECU. Highlighted in color are parallel, non-interfering communication paths through the network. All connections support up to 100 Mbps full-duplex data transmission. As a result, the bandwidth of 100 Mbps is multiplied by the number of parallel communication paths. Ethernet provides 1:n (multicast) and 1:all (broadcast) communication as well. If these addressing methods are not used skillfully, the described advantages of Ethernet are lost.

In the case of unicast addressing, the intelligence migrates on the network from the receiver to the sender. The sender must know which receivers in the network are interested in which data (PDUs) and must assemble the messages appropriately using n-PDU-to-frame mapping. It is possible for PDUs or messages to be sent multiple times if there are multiple receivers for the same PDUs. This receiver-specific data fan-out can be implemented most easily in AUTOSAR with the Socket Adaptor.

New communication paradigm: Service-oriented communication

The properties of Ethernet as well as the desire of vehicle manufacturers for more flexibility and controllability of the networking complexity led to the introduction of service-oriented communication in the automotive environment. This familiar communication paradigm from the Internet world has been transferred to vehicle networking. However, specific protocols optimized for automotive applications are being used: Service Discovery (SD) and Scalable Service-oriented Middleware over Internet Protocol (SOME/IP). Up to now, we spoke of senders and receivers in the vehicle network. With service-oriented communication there is one ECU that provides a service (the server) and ECUs that use this service (the clients).

Servers use the Service Discovery protocol to announce which services they provide at runtime and how these services are addressed. Clients invoke methods of the server (Remote Procedure Calls) or subscribe at the server, in order to subsequently receive data updates automatically. In the latter case, the server provides certain data elements (called events) and sends their values to all subscribed clients. The application of the server triggers the sending of data updates. The n-PDU-to-frame mapping enables multiple events to be sent within one message. Figure 4 graphically represents the two principles of service-oriented communication.

For method invocations and data updates, the length of the data to be transmitted can vary significantly. The support for this variable data length is a main feature of SOME/IP. It serializes structured and complex application data so that the remaining basic software of an ECU only has to take care of sending or receiving a byte stream. For this reason, the exact message layout is no longer defined in a database.

AUTOSAR fully supports SD and SOME/IP. Because of the platform independence of SD and SOME/IP, the protocols are also used for data exchange between AUTOSAR ECUs and other platforms.

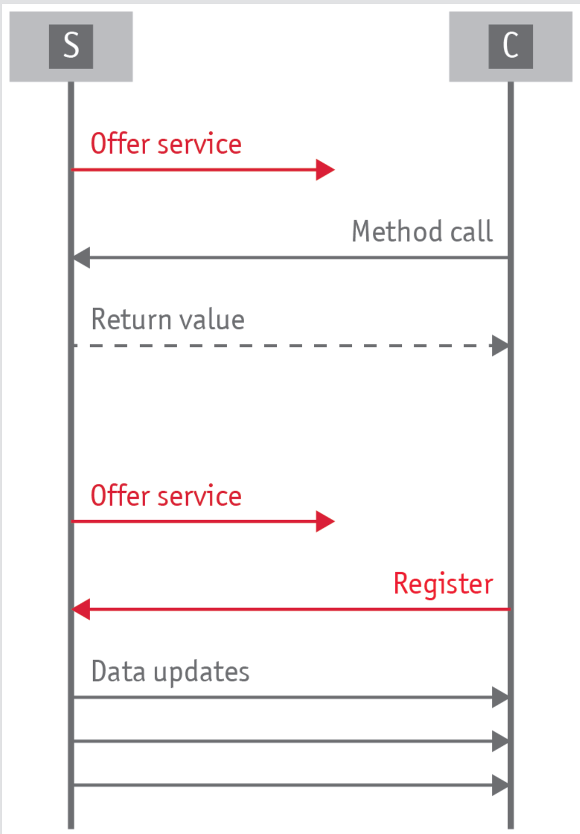


Figure 4: Operating principles of service-oriented communication using Service Discovery

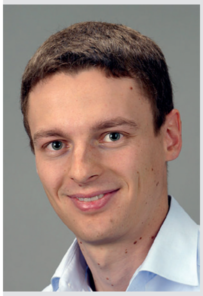
Conclusion

With their greater payload data length, CAN FD and Ethernet provide new data transmission possibilities. They also create new challenges for vehicle manufacturers and their suppliers – for example, when creating the system description. Assistance is provided by n-PDU-to-frame mapping, which packs multiple PDUs in one CAN FD or Ethernet message. In the case of Ethernet, however, there is not only an increase in the user data length but also in available bandwidth. Furthermore the switched network and associated new addressing method represent a small revolution in the automotive environment. New mechanisms for data distribution are necessary. Building on this, data exchange is becoming more dynamic by means of service-oriented communication. AUTOSAR 4.2.1 already supports all of the presented mechanisms and thus facilitates implementation of the new communication paradigms. Basic software implementations of this AUTOSAR version are already available on the market. An example is MICROSAR from Vector along with the development tool CANoe which offers convenient analysis and testing of CAN FD and Ethernet networks.

A shortened version of this article was published in the German magazine Automobil Elektronik, issue 7/8 2015.

All figures:
Vector Informatik GmbH

Links:
Website Vector: vector.com
Product information MICROSAR: www.vector.com/microsar
Product information CANoe: www.vector.com/canoe



M.Eng. Marc Weber
is responsible for product management of the Ethernet stack for the Embedded Software product line of Vector Informatik.

>> Contact information of the Vector Group:
www.vector.com/contact