IP Concepts on CAN XL

How SOME/IP can be transformed into „SOME/CAN“

Oliver Garnatz, Peter Decker, Vector Informatik

The electronics, sensor assemblies and networks in automobiles are currently undergoing fundamental changes. Many new tasks can only be handled through use of modern information technology, including Automotive Ethernet and Internet Protocol (IP). Thus, in addition to classic signal-oriented communication, service-oriented communication has also become an aspect of automotive technology. The coexistence of classic and IP-based networks has resulted in an overlapping situation in the transmission speed range of 10 Mbit/s. While a suitable system is available from above in the form of 10 Mbps Ethernet (10BASE-T1S), the newest CAN development „CAN XL“ is pushing into the 10-Mbit/s domain from below. The two technologies are characterized by opposite operating principles, and for this reason the conditions under which CAN XL can handle service-oriented communication tasks – in addition to signal-based communication – is the subject of the current investigation. The following discussion focuses primarily on networks in automobiles.

Currently, the development departments in the automotive industry are, for the most part, concentrating on the challenges posed by the transformation in mobility. The focus is on assistance systems (ADAS – Advanced Driver Assistance System), autonomous driving, electric mobility and continuous connection to the Internet or to the Cloud. High-performance sensor systems such as radar, laser scanners and video cameras in the vehicle are an indispensable prerequisite for autonomous driving. They generate amounts of data that were unknown in the automotive industry only a few years ago. The challenge is how to transmit and process this exploding data volume in real time. To meet this challenge, the industry has introduced Automotive Ethernet for fast transmission of data, covering primarily the bandwidths of 100...1000 Mbit/s (100BASE-T1, 1000BASE-T1) and used initially in the ADAS area. At the lower end of the Ethernet networks, development is focused on 10BASE-T1S, with a transmission speed of 10 Mbit/s.

Classic Automotive Bus Systems: Perfect for Control Tasks

In contrast, the classic automotive networks such as CAN/CAN FD and FlexRay employ signal-based communication technology. In most applications, CAN operates at a transmission rate of 500 kbit/s and is used in areas such as engine management and body control. The capabilities of CAN, a pioneer in automotive networks, are extended upwards by FlexRay and CAN FD, whose transmission rates range from 1...10 Mbit/s. These newer systems are predestined for time critical applications in engine management, body control and chassis control, where they are used, for example, in the brake system. Lastly, MOST, which is responsible for infotainment applications, covers the 25...150 Mbit/s domains.

Given the rise of automotive Ethernet and in view of the growing number of communication systems, a consolidation appears reasonable to limit complexity and costs. Since the fields of application of FlexRay and MOST can also be sufficiently covered by Ethernet, these systems will likely be replaced in the medium term. This would leave CAN and Ethernet, with Ethernet now handling infotainment, ADAS, telematics and connectivity at 100...1000 Mbit/s. CAN and CAN FD operate in the range of 0.5...5 Mbit/s and are responsible for engine management, and body control.

CAN XL or 10BASE-T1S could be used in the future for chassis control systems.

The Automotive 10 Mbit/s Domain

Against this backdrop, it is interesting to examine the 10 Mbit/s limit of CAN and Ethernet more closely. While Ethernet with 10BASE-T1S is positioning itself exactly here from above, the new CAN XL, coming from below, is expanding into the 10 Mbit/s domain. Considering that about 90 % of all network nodes communicate at speeds below or up to 10 Mbit/s, the 10 Mbit/s domain covers a wide field of application. It extends from audio applications to radar and ultrasonic sensors all the way to chassis control. From the technical standpoint, the initially mentioned applications focus on streaming and serializing of data as well as the principle of service orientation. In contrast, for applications in chassis control, signal-oriented communication dominates.

Both 10BASE-T1S and CAN-XL domains could frequently operate as network branches under a 100BASE-T1 domain. Coupling of 10BASE-T1S to 100BASE-T1 is possible without problems through use of a switch. In contrast, to connect CAN and CAN XL branches, a gateway is required. With their different approaches, both models have advantages and disadvantages, and theoretically could exist in parallel to each other. The only question is whether this is what vehicle manufacturers have in mind or whether a decision in favor of one of the two systems is envisioned. With this in mind, it is necessary to clarify the extent to which CAN XL is also suitable for IP applications and what are the prerequisites for this. Accordingly, it makes sense to first look more closely at the technical principles underlying both systems.

10BASE-T1S: Bus Topology with Controlled Network Access

This refers to the 10 Mbit/s version 10BASE-T1S[1] (S – Short Distance OR: Short Range), which is intended explicitly for automobiles and covers short distances of up to 25 meters. It should not be confused with 10BASE-T1L (L – Long Distance), which provides ranges of up to 1000 meters for industrial applications.

An unshielded twisted wire pair serves as physical layer for 10BASE-T1S („T1“). In contrast to today‘s other switched-Ethernet versions, the topology for 10BASE-T1S is a bus. All participants are connected to a common Ethernet cable (multi-drop bus topology) by short stubs of max. 10 centimeters in length. This immediately raises the question of network access: In the Ethernet-PHY, a round-robin approach is implemented that allows collision-free network access via PLCA (Physical Layer Collision Avoidance). This guarantees deterministic response times for each network participant and provides real-time capability in the application. Collision-free access further allows complete use of the entire bandwidth of 10 Mbit/s. 10BASE-T1S offers only half-duplex operation, for which only one PHY per ECU is required instead of two per connection.

With these characteristics, 10BASE-T1S is basically also suitable for applications found in classic automobile networks. The decision as to which communication system will play a predominant role in the future always depends, of course, also on cost considerations. As will be shown below, however, the situation is somewhat more multi-layered in this case.

CAN XL – Newest and Fastest CAN

CAN XL[2] is a further development of CAN and CAN FD, and operates largely on the same principle. A CAN message can be divided into arbitration and data phases. While CAN XL uses low transmission speeds of 500 kbit/s to 1 Mbit/s in the arbitration phase, the speed in the data phase is scalable over a wide range of 500 kbit/s to 10 Mbit/s. Also scalable is the useful data length, which for CAN XL can have a variable length from 1...2,048 bytes.

---

[1] *[1]*

[2] *[2]*

The access method used is CSMA/CR (Carrier Sense Multiple Access/Collision Resolution), which resolves competing write access through bit arbitration. In this way, CAN follows a strict priority concept that allows the more important message to be transmitted with no delays. In contrast to 10BASE-T1S, CAN XL offers the ability to use more complex topologies with a star and long stubs. For this reason, proven, existing CAN solutions cannot be replaced one to one with 10BASE-T1S networks with their considerably more restrictive bus topology. On the other hand, nothing stands in the way of upgrading from CAN/CAN FD to CAN XL in this regard, since a great deal of know-how and development time has already been invested in wire routing and carefully designed cable harnesses.

It is precisely this backward compatibility that makes CAN XL interesting for those automakers who focus primarily on compact and midsize cars. In this mass market, autonomous driving will not be found for some time, at best you will find simple assistance systems that have already been in common use for years, for example anti-lock brake systems. Without radar sensors, high-resolution cameras, etc., there is no need for an Ethernet network; instead, the classic systems will predominate, above all, CAN, of course. For such vehicles, CAN XL offers the ideal platform for further development on the basis of the existing vehicle architecture. No redesigns of cable harnesses, controllers, and protocol stacks are necessary. The simpler protocol stack for CAN compared to that for IP allows use of smaller and thus lower-cost controllers. One goal for CAN XL would be to continue this tradition. Having said that, the rest of this article will be a neutral, strictly technical presentation.

Signal-Based Communication

A more in-depth understanding of the technology can be gained by comparing the two different communication forms: signal-based communication vs. service-based communication. For typical control tasks, the signal-based approach has been tested and proven for almost three decades. Together with the priority principle used with CAN, the system ideally satisfies the necessary real-time requirements. A major feature of signal-based communication is the predefined static communication matrix. Signals such as temperatures, pressures, speeds or revolutions always represent the same fixed quantity, which is mapped to an established CAN frame and sent to ECUs. In addition, so-called PDUs have been introduced, which form an intermediate layer and make communication more flexible[3]. PDUs can be represented on a frame statically or dynamically. While in the static version a PDU is represented directly on a frame in the same way as the signal was above, in the dynamic version the PDU header contains length data in addition to the identifier. As a result, PDUs are no longer bound to a static layout in a frame.

Service-Oriented Communication

Service-oriented communication goes hand in hand with Ethernet and IP technology. Applications need data and services. It does not matter who provides them. However, a dynamic link connection between data sink (consumer) and data source (provider) is required. As intermediary between consumer and provider within vehicles, the middleware SOME/IP (Scalable service-Oriented MiddlewarE over IP) with its service discovery process SOME/IP-SD is used. In principle, both the slower but more secure transmission control protocol (TCP) and the faster but not secure User Datagram Protocol (UDP) can be used for service-oriented communication.

The ability to transmit dynamic data structures is another major advantage of service-oriented communication. The amount of data to be transmitted, for example in the case

[3] *[3] [4]*

of sensor fusion applications, is generated only during the runtime of the application. Such data cannot be mapped statically; instead, the communication system must serialize the data dynamically. Responsible for serialization in the AUTOSAR Classic Platform is the SOMEIPXF module. Since it is part of the middleware level of AUTOSAR, its functionality can be used to serialize dynamic data for CAN XL as well.

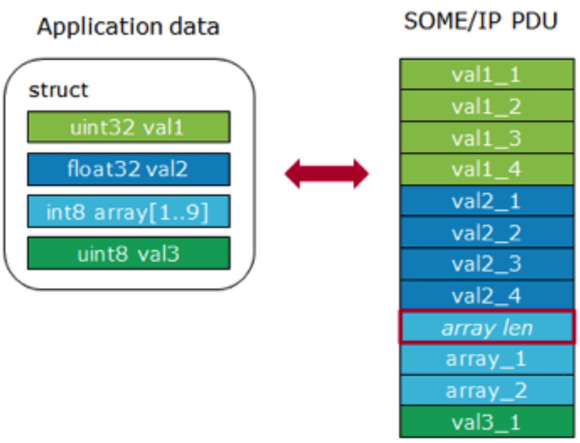

Figure 3: Dynamic data serialization

Fully Dynamic Link Connection with SOME/IP-SD

SOME/IP supports both fully dynamic and semi-dynamic link connections. The fully dynamic link connection is used when the network nodes do not know each other‘s IP and MAC addresses. In the example, the three ECUs Eth1, Eth2 and Eth3 are connected to a switch via Ethernet. Eth2 knows that it needs Service1, which is available on Eth1. However, Eth2 knows neither the IP nor the MAC address of Eth1, nor does it know who offers Service1. Eth1 now offers Service1 via multicast or broadcast with SOME/IP as middleware. The SOME/IP-SD offer message contains the IP address of Eth1 and the port for Service1. Thus, Eth2 is able to subscribe to an event group or call up a method for Service1. To this end, the consumer determines the MAC address of Eth1 using ARP (Address Resolution Protocol), so that it now knows the IP address and MAC address. However, ARP must run once again for a response from Eth1, since ARP is unidirectional.

There are several benefits associated with this method, which establishes dynamic communication on all protocol levels during the runtime: thus, Service1 can be moved to any other nodes in the network without changes being necessary on Eth2. The same holds for MAC and IP addresses; they can be changed, and again no changes are required on Eth2.

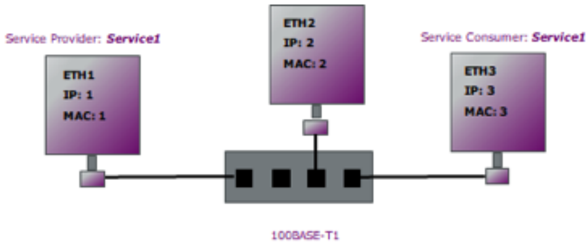

Figure 4: Fully Dynamic Communication Setup

Semi-dynamic Link Connection with SOME/IP-SD

Likewise, there are reasons in favor of a semi-dynamic link connection with static IP addresses and MAC addresses. Each ECU has a mapping table in which the IP and MAC addresses of the other network nodes are saved. In the example, Eth2 again needs Service1 offered by Eth1 via multicast or broadcast and SOME/IP-SD. Eth2 can now subscribe directly to the event group of Service1 without invoking ARP.

This method also establishes dynamic communication on the service level during the runtime, but allows a faster start of communication without ARP, as would be expected. With a semi-dynamic link connection, Service1 can also be moved arbitrarily, for instance to Eth3. Since Eth2 already knows the IP and MAC addresses of Eth3, no changes are needed here either. The drawback here is that IP/MAC addresses can now no longer be changed. In this case, the mapping tables in all ECUs involved would also have to be updated. What is important to know is that, depending on the vehicle or model series, the industry sometimes uses fully dynamic and sometimes semi-dynamic link connections. SOME/IP functions on 10BASE-T1S in the same way as on 100BASE-T1 networks; that is, it does not matter whether it is running on a bus topology or purely on switched networks.

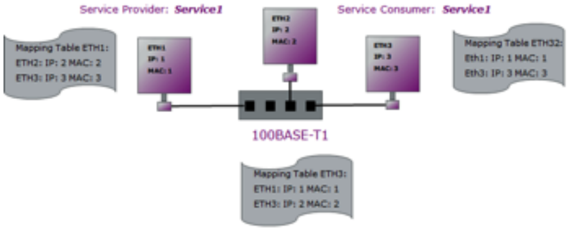

Figure 5: Semi-Dynamic Communication Setup

How can service-oriented communication be represented on CAN XL?

Let us return to the initially presented question as to whether and how CAN XL can be incorporated into an Ethernet/IP environment? The primary goal is to have CAN XL ECUs able to participate in service-oriented communication. This question can basically be answered with „yes", but there are various ways to achieve this. These will now be presented below.

Routing Ethernet Frames

The first possibility is to route Ethernet frames on CAN XL. To this end, a standard Ethernet switch can be used. On the hardware side, it is necessary to insert a still to be developed CAN XL PHY between the port to which the CAN XL network is attached and the CAN XL bus. It will be able to copy all of the Ethernet frames to CAN XL frames and vice versa – depending on the direction of communication. The CAN XL PHY is needed only at the Ethernet switch, while at the CAN XL nodes commonly used transceivers suffice.

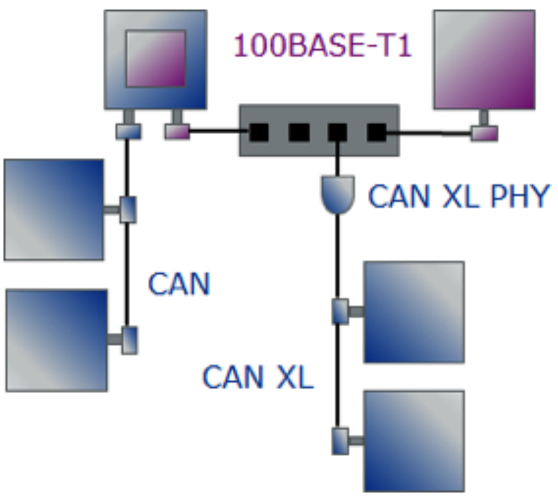

Figure 6: Ethernet switch with CAN XL PHY

The demands on the CAN XL stack are considerably higher. While Ethernet frames can be incorporated into CAN XL, a common TCP/IP stack is also required in the CAN XL ECU. Keep in mind: Embedded in the CAN XL frame is an Ethernet frame, which also contains an IP packet. The interface layer, in turn, must be able to accommodate the behavior of CAN as well as that of Ethernet. The CAN part of the interface layer unpacks the Ethernet frame, the Ethernet part unpacks the IP frame. In addition, each CAN XL node requires a virtual MAC address. For the CAN XL PHY, only one CAN Tx ID is needed and for each CAN XL node one CAN Tx ID each is needed. Filter functions can be performed on the basis of the MAC address embedded in the frame. Under these conditions, SOME/IP, SOME/IP-SD and ARP function exactly as in a pure Ethernet network.

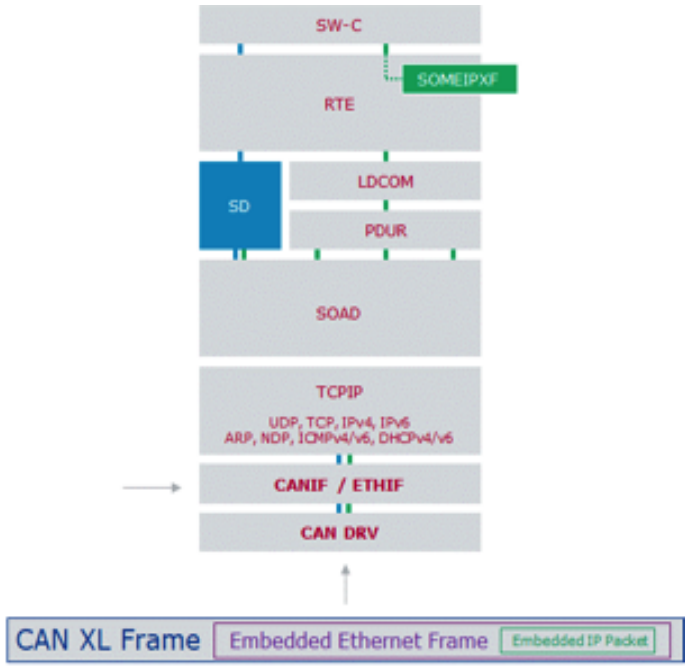

Figure 7: CAN XL Stack for Ethernet Routing IP Frames

A second possibility is to use a suitable gateway to route the IP frames to CAN XL instead of the Ethernet frames. The task of the gateway is to unpack the IP frame from the Ethernet frame. With the aid of a suitable routing table in the gateway, the gateway recognizes the embedded IP address as a packet to be packed in a CAN XL frame and routed to the CAN XL bus.

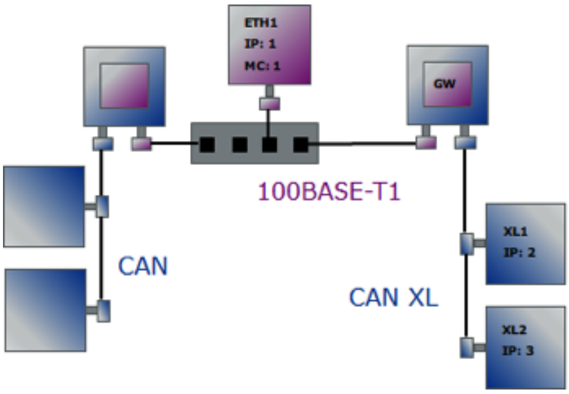

Figure 8: CAN XL with IP Gateway

Here, too, a TCP/IP stack is needed on the hardware side of the CAN XL ECU. For the CAN interface, only minor changes are needed in the implementation, the same is true of the TCP/IP stack. Each CAN XL node has a CAN Tx ID, including the gateway. The IP address embedded in the frame can be used for filtering. In this scenario as well, SOME/IP and SOME/IP-SD function exactly as in a pure Ethernet network.

Figure 9: CAN XL Stack for IP

The first two possibilities presented can be summarized as follows: SOME/IP functionality can be achieved in both cases. Routing of Ethernet frames only requires a CAN PHY at the standard switch, while for the software stack on the CAN XL ECUs a new intermediate layer is required. When routing IP frames, changes in the software stack are also needed, but to a lesser extent than when routing Ethernet frames. Logic changes are not needed in the software modules; only the implementation needs to be modified. Hardware filtering would be useful in both cases, but would require quite a bit of effort.

Service Orientation on CAN XL without TCP/IP

A solution that dispenses completely with the TCP/IP stack is presented as a third possibility. The motivation for this is that about 50-100 kB of ROM storage capacity can be saved in each CAN XL ECU, allowing use of smaller and lower-cost controllers. Here, a newly introduced „SOME/CAN“ layer replaces the TCP/IP (TCPIP) and socket adapter (SOAD) modules in the software stack. The SOME/CAN layer is a suggestion proposed in this paper and would need to be specified and implemented in greater detail if there is market interest.

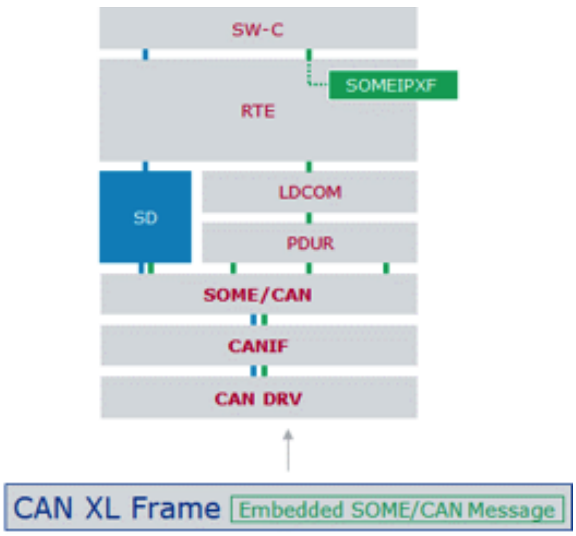

Figure 10: SOME/CAN Stack

Routing and the conversion of SOME/IP into SOME/CAN takes place in the gateway. SOME/IP messages are converted in the PDU router module. SOME/IP-SD messages must be deserialized in the application and correspondingly re-serialized as SOME/CAN messages. In doing so, the gateway replaces the IP address and the port number in the SOME/IP-SD header, for instance, with a CAN ID, and identifies the frame as „CAN XL Type“.

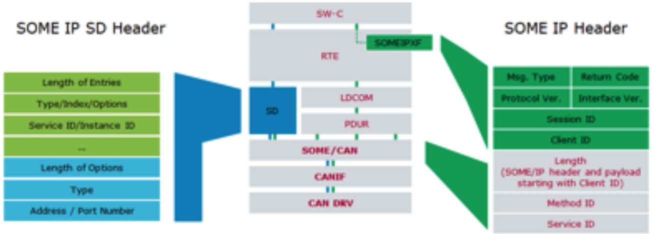

Figure 11: SOME/CAN Header

A CAN XL frame now transports an embedded, modified SOME/IP frame, which consequently should be designated a „SOME/CAN message“.

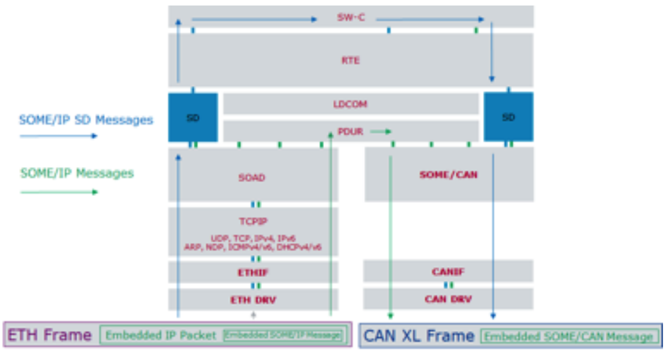

Figure 12: SOME/IP-SOME/CAN Routing

While in Ethernet SOME/IP subscribers listen to a dedicated (UDP) port, SOME/CAN subscribers wait for special SOME/CAN IDs. Whether a SOME/CAN message or service discovery is involved, is indicated in the header by the message value. The message value 0xFFFF 8100 identifies a service discovery message.

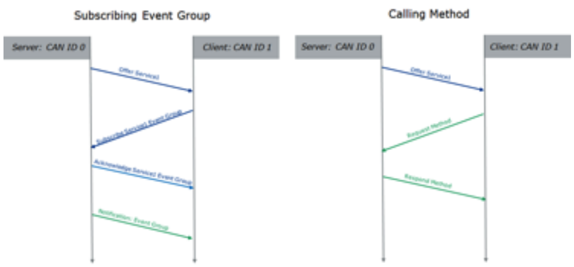

Figure 13: SOME/CAN Communication

SOME/IP communication can be transformed into CAN without a TCP/IP stack as shown. This makes an ID-based SOME/CAN possible, which, however, depends on certain software modules in an appropriately expanded AUTOSAR stack. Filtering via hardware is thus excluded.

Extensions Toward Hardware Filtering

If the SOME/CAN approach were adequately developed further, hardware filtering could still be practicable. To this end, each participant would receive a node address. The node address, which is part of the data field, now allows filtering via hardware. In this regard, it is further necessary to provide multicast or broadcast addresses for the service offering. Since node addresses are now used for addressing, the gateway must map them statically. For dynamic mapping, an appropriate address resolution solution for CAN node addresses would need to be implemented. Network access and filtering can be decoupled if SOME/CAN is introduced as a protocol type in a further step. The protocol type replaces the semantics of a CAN identifier used for service orientation. This permits pre-filtering based on the protocol type, and the CAN identifier is then only required for network access and prioritizing. This means that the priority can be changed without having to make a change to a participant.

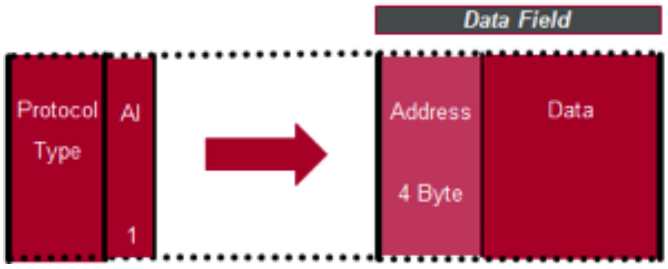

Figure 14: Protocol Type and Node Address

Conclusion

With appropriate extensions, IP concepts and service-oriented communication can also be applied in CAN XL networks. The functions required for serializing dynamic data can also be used by CAN XL. An initial possible solution utilizes a standard Ethernet switch in conjunction with a CAN XL PHY. It copies Ethernet frames to CAN XL frames unchanged. The effort and later the runtime increase in the software stack of the CAN-XL controller, which has to be extended by TCP/IP functionality.

In the second option, IP packets are routed to CAN XL. Logic changes are not required in the software modules, only modifications in the implementation. Modification of the software stack is also unavoidable here, but to a lesser extent than with routing of Ethernet frames.

Hardware filtering must also be taken into consideration. Without it, the interrupt load increases drastically. The use of lower-cost controllers could be jeopardized by this. Filtering by MAC or IP addresses, however, requires an expensive filter in the CAN XL hardware.

Another possibility is to bring SOME/IP communication to CAN XL without using a TCP/IP stack. ID-based SOME/CAN functions, but again does not allow hardware filtering. With the aid of node addresses, SOME/CAN with hardware filtering is possible. SOME/CAN with protocol type and node addresses is conceivable as an extension, which on the one hand, allows hardware filtering, while simultaneously decoupling network access and filtering.

Oliver Garnatz
Vector Informatik GmbH
Holderäckerstr. 36
DE-70499 Stuttgart
www.vector.com

Oliver Garnatz (Dipl.-Ing. (FH)) has been employed at Vector Informatik GmbH since 2000 and serves as Product Manager in the Embedded Systems product line.

Peter Decker
Vector Informatik GmbH
Holderäckerstr. 36
DE-70499 Stuttgart
www.vector.com

Peter Decker (Dipl.-Ing. (FH)) has been employed at Vector Informatik GmbH since 2002 and serves as CAN Product Manager in the Networks and Distributed Systems product line.

References

[1] IEEE P802.3cg™/D3.0, 10BASET1S
[2] CiA 610, CAN XL Data link layer and physical signaling
[3] I-PDU Multiplexer, AUTOSAR Release 4.4.0
[4] PDU Router, AUTOSAR Release 4.4.0
[5] SOME/IP Protocol, AUTOSAR Release 4.4.0
[6] SOME/IP Service Discovery, AUTOSAR Release 4.4.0
[7] SOME/IP Transformer, AUTOSAR Release 4.4.0

All figures by Vector Informatik GmbH