IP concepts with CAN XL

(Source: Vector)

The authors take a look on IP concepts with CAN XL and the transformation of Some/IP towards Some/CAN. Additionally, this article sets out several options for implementing service-oriented communication with CAN XL.

The networking of future generations of automobiles will be based primarily on use of the various Ethernet and CAN variants. While Ethernet dominates with its service-based IP communication and Some/IP middleware at the level of assistance systems, signal-based CAN networking will continue to exist long into the future in the drive and chassis sector. The new CAN XL should assume an important role in enabling these two fundamentally different concepts to co-exist and to work alongside one another. This raises questions as to whether the CAN XL ECU (electronic control unit) participates in service-based communication, and about the possible ways that might exist for enabling this to happen.

In broad terms, the key technical parameters for CAN XL have been defined: The new CAN variant offers data speeds of up to 10 Mbit/s and, with a variable length of user data in a range of 1 byte to 2 048 byte, it is also capable of transporting complete Ethernet frames inside CAN XL frames. As you would expect, CAN XL is otherwise broadly speaking reverse-compatible with Classical CAN or CAN FD and with the concept of signal-based communication. This is of particular benefit for the further development of electronic architectures for small and compact automobiles for which, at this time, no Ethernet high-performance communication is required for advanced driver assistance systems (ADAS) and autonomous driving. CAN XL makes it possible to continue using existing networking concepts and wiring harnesses without the need for big modifications.

Best of both worlds with CAN XL

Nonetheless, the future will inhabit the realm of service-oriented IP communication with Ethernet, according central significance to Some/IP (scalable service-oriented middleware over IP) middleware with its Some/IP-SD service discovery process. On board vehicles, Some/IP enables dynamic links to be established between providers (data sources) and consumers (data sinks). For modern applications, it does not matter who is supplying data or services.

Service-oriented communication is also in charge of the transmission of dynamic data structures. The volume of data to be transmitted, for example in the case of sensor data fusion applications, is generated only during the runtime of the application. Such data cannot be mapped statically in the way typical of signal-based communication. Instead, the communication system must serialize the data dynamically. The SomeIpXf module is responsible for serialization in the Autosar Classic Platform. Since it is part of the middleware level of Autosar, its functionality can be used to serialize dynamic data for CAN XL as well.

Dynamic link connection

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. A few benefits are associated with the establishment of dynamic communication on all protocol ▷

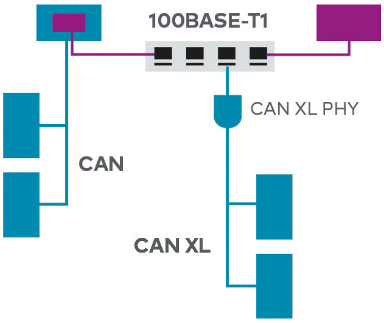

Figure 1: Ethernet CAN network with CAN PHY (Source: Vector Informatik)

levels: A service can be relocated within the network to any other desired node without requiring modifications to the ECU. The same applies to MAC and IP addresses. When required, consumers and providers may make multiple use of the Address Resolution Protocol (ARP) to determine their respective MAC and IP addresses.

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. This method also establishes dynamic communication on the service level during runtime, but allows communication to start faster because it can do so without ARP. With a semi-dynamic link connection, services can also be moved arbitrarily because all IP and MAC addresses are known. 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 works with every Ethernet variant, regardless of whether it only involves switched networks or ones with network topology.

Service-oriented communication on CAN XL

To do justice to its role as the link element between Ethernet and CAN domains, CAN XL should also be able to participate in service-oriented communication. It is therefore of great interest to the designers of future E/E architectures to establish which options CAN XL can offer to this in technical terms. At the same time, users continuously strive to find the most cost-effective solution. A crucial factor in this is to establish which requirements individual solutions impose on the software stack and on the ECU hardware.

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 develop and then incorporate a CAN XL PHY between the port to which the CAN XL network is attached and the CAN XL network. The CAN XL PHY should be able to copy all of the Ethernet frames to CAN XL frames and vice versa – depending on the direction of communication. It is needed only at the Ethernet switch, while at the CAN XL nodes commonly used transceivers suffice. Of course, it is also always possible to use a conventional gateway (figure 1).

The demands on the CAN XL stack are considerably higher. As soon as Ethernet frames can be incorporated in CAN XL, a common TCP/IP stack will also be 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, while the

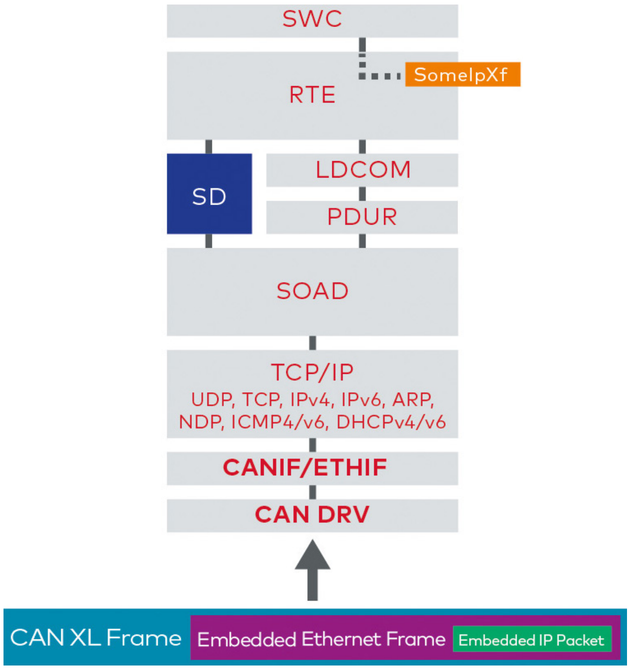

Figure 2: Stack for embedded Ethernet communication via CAN XL (Source: Vector Informatik)

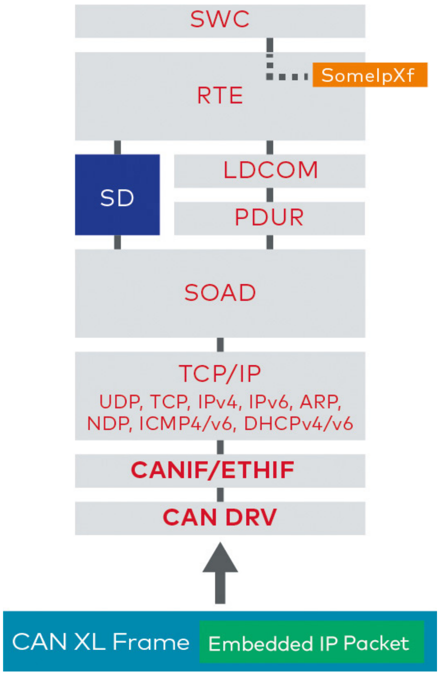

Figure 3: Stack for embedded IP communication via CAN XL (Source: Vector Informatik)

Ethernet part unpacks the IP frame. In addition, each CAN XL node requires a virtual MAC address. The CAN XL PHY then just requires one CAN frame for further transmission of the frames received by the Ethernet network to CAN XL, and for every CAN XL node to have another CAN frame for response data. 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 2)

Routing IP frames

A second possibility is to use a suitable gateway and to route IP frames instead of Ethernet frames to CAN XL. The task ▷

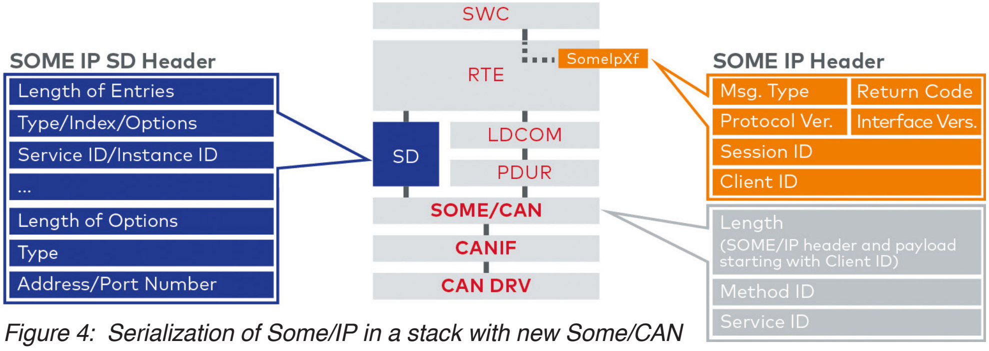

Figure 4: Serialization of Some/IP in a stack with new Some/CAN module (Source: Vector Informatik)

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 network. Here too, a TCP/IP stack is needed by the CAN XL ECU. For the CAN interface, only minor changes are needed in the implementation, but major changes can be expected in the TCP/IP stack. 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 3).

From the first two possibilities presented, we can conclude: Some/IP functionality can be achieved in both cases. Routing of Ethernet frames requires a new CAN PHY at the standard switch, while a new intermediate layer is necessary for the software stack on the CAN XL ECUs. When routing IP frames, changes in the software stack are also needed. 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 be resource-intensive.

A third possible solution is to dispense completely with the TCP/IP stack. The motivation for this is that about 50 KiB to 100 KiB of ROM storage capacity can be saved in each CAN XL ECU, enabling smaller and lower-cost controllers to be used. Here, a newly introduced "Some/CAN" layer replaces the TCP/IP (TCPIP) and Socket Adaptor (Soad) modules in the software stack. The Some/CAN layer is a suggestion by the authors and would need to be specified and implemented in greater detail if there is market interest. Routing and the conversion of Some/IP into Some/CAN takes place in the gateway. Some/IP messages are converted in the PDU (protocol data unit) router module. Some/IP-SD messages need to be deserialized in the application, then to be serialized back into corresponding Some/CAN frames. 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". A CAN XL frame now transports an embedded, modified Some/IP frame, which consequently should be designated a "Some/CAN frames". While in Ethernet Some/IP subscribers listen to a dedicated (UDP) port, Some/CAN subscribers wait for special Some/CAN IDs. Whether Some/CAN frames or service discovery is involved, is indicated in the header by the message value. The message value FFFF 8100_h identifies a service discovery message (figure 4).

Some/IP communication can be transformed into CAN without a TCP/IP stack as shown. This makes Some/CAN feasible, although this does depend on certain software modules in an appropriately expanded Autosar stack. Filtering via hardware is therefore excluded.

If the Some/CAN approach were adequately developed further, hardware filtering could still be practicable. To this end, each user would receive a node address. This node address then enables hardware filtering. In this regard, it is also 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 needs to be implemented. A node address separates filtering and network access. This means that the priority can be changed without having to make a change to a user.

Outlook

This article sets out several options for implementing service-oriented communication with CAN XL. While the routing of Ethernet frames and IP frames can be envisaged, there needs to be a good way to filter hardware in order to be efficient. To implement smaller and more cost-effective controllers, it would be advisable to implement Some/IP communication by dispensing with a TCP/IP stack on CAN XL. The problem of early hardware filtering can be remedied by introducing a node address to avoid interrupt loads. The details of the direction in which service-oriented communication on CAN XL may yet develop are still unclear. The fact that service orientation on CAN XL is needed is demonstrated by the applications that are to be implemented. In overall terms, CAN XL is well equipped for this.

On page 28 (Change in automotive communication systems), the authors take a look on the transformational change of communication systems. CAN XL provides the basis for cooperation between IP technology and signal-based communication. It closes the gap between CAN FD and Ethernet. ◀

Authors

Oliver Garnatz, Peter Decker
Vector Informatik
info@vector.com
www.vector.com

This article originally appeared in Hanser Automotive in German language.