CAN FD with dynamic multi-PDU-to-frame mapping

Oliver Garnatz / Peter Decker, Vector Informatik GmbH

The AUTOSAR release 4.2.1 introduced a Dynamic Multi-PDU-to-Frame Mapping (PDU: Protocol Data Unit). Primarily this allows an easy migration from existing architectures to CAN FD with a higher data-rate (e.g. packing multiple classic CAN-frames into one CAN FD frame).
But this new functionality allows also a reduction of communication variants across the vehicle model ranges. By assigning each data-element/object an own unique ID (PDU-ID) across the full model range, the communication is getting independent from the frame architecture and CAN ID. Functionalities can be moved within the vehicle without modifying any receiving node.
This approach opens a high flexibility in the variety of vehicle architectures. But there are also drawbacks. The resource demand for each ECU increases. Tools are also concerned. Testing and analyzing requires new functions.
The paper introduces the Dynamic Multi-PDU-to-Frame Mapping with its benefits and its drawbacks. Further it shows the impact on gateways and its influence on SOME/IP. Requirements for analyzing and testing the dynamic PDU concept will finalize the paper.

The Basic Requirement: Bandwidth

The most obvious trend in E/E architectures is the demand for more communication bandwidth to add new functions to a vehicle.

Central Gateway Architecture

When automotive networks mainly consisted of CAN the first solution to gain more bandwidth was splitting up CAN networks. E.g. into functional clusters. Typically these clusters were connected by a central gateway (figure 1).

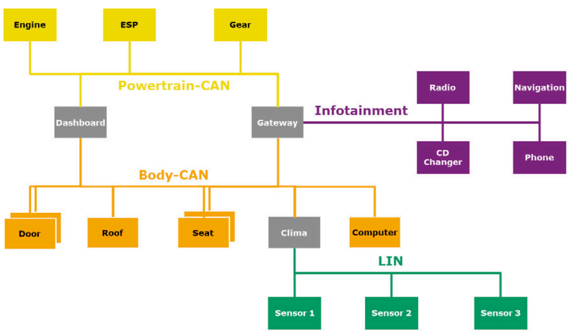

Figure 1: Central gateway architecture

A new network protocol: FlexRay

The next step to address new automotive use cases and to gain bandwidth was the introduction of FlexRay.

FlexRay was designed to cover safety aspects in communication e.g. for x-by-wire applications.
FlexRay comes with a payload of up to 255 bytes and a maximum bit rate of 10 Mbit/s. To cover a high predictability FlexRay introduced a scheduled communication of 64 repeating cycles.
Each cycle with a duration of typically 5ms consist of static slots with a guaranteed timing and dynamic slots to cover the event driven communication.
The slot-id is similar to the CAN-ID and defines the content of the slot.

New architecture: Backbone

FlexRay was an enabler for a new kind of E/E architecture. Domain architectures are used instead of having a central gateway. Each functional cluster is controlled by a domain controller and all domain controllers are interconnected by a FlexRay backbone network. Typical domains are Body, Chassis, Powertrain and Infotainment. The benefit is a more flexible E/E architecture where the slave networks are controlled by their domain controllers. The domain controller is a kind of router. It is forwarding only the information which is required in a certain domain.

A backbone architecture reduces the need for bandwidth in all slave networks. The slave networks benefit from the higher bandwidth capabilities of the backbone data link (figure 2).

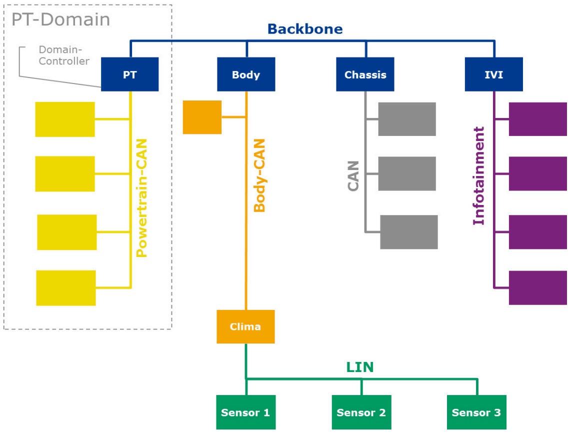

Figure 2: Backbone architecture

Data link vs PDU mapping

In the automotive networking the exchanged information are usually signals which are often very small. Sometimes a signal is only a few bits in size. E.g. the information whether a button is pushed or released only requires one bit. To optimize the payload-to-overhead ratio several signals are mapped to one frame (figure 3).

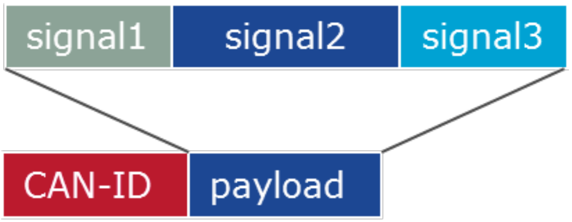

Figure 3: Signal to PDU mapping

To get a better grouping of signals, which are related to each other, AUTOSAR defined a new abstraction between signal and the data link frame. Signals are not mapped directly on a frame any more but to a PDU (Protocol Data Unit). Later the PDU is mapped to a data link frame.
This allows using the same PDU in different frames and to propagate the PDU on different networks.
Classic CAN does not really make use of this concept due to its limited payload of 8 bytes. AUTOSAR just defines a classical CAN Frame as a single PDU. The CAN-ID defines the content of the frame.

Static PDU-to-Frame mapping

With FlexRay the PDU-mapping to the data link frame became meaningful due to FlexRay's payload of up to 255 bytes (in practice up to 48 bytes are used). A frame can contain multiple PDUs each mapped to a fixed position (figure 4).

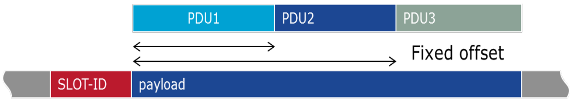

Figure 4: Static PDU mapping

Several CAN frames represented as PDUs can thus be mapped to a single FlexRay frame.
But the time-triggered behavior of FlexRay does not suit very well the event-driven behavior of CAN. It can happen that update cycles of CAN PDUs are different from the FlexRay cycle. Therefore, the same data is transmitted twice or one PDU update is lost.

The CAN evolution: CAN FD

With CAN FD the next network with larger payload (up to 64 bytes) and higher bit rate has been introduced. CAN FD has been designed to close the gap between classical CAN, that usually works on vehicle networks with 500 Kbit/s, and FlexRay with a common bit rate of 5Mbit/s.
In first vehicle projects CAN FD will work with 2Mbit/s in its data phase and 500 kBit/s in the arbitration phase. This leads to an average bit rate of 1.5 Mbit/s when using in average a payload of 32 bytes. Having the higher payload of CAN FD Dynamic Multi-PDU-to-Frame Mapping becomes meaningful. In contrast to FlexRay a more dynamic approach is followed.

Dynamic multi-PDU-to-frame mapping

The motivation is a dynamic construction of the frame during runtime. This means the position of a PDU within a frame is not static. Also the length of a PDU can be dynamic. Therefore a header identifying the PDU is required to enable the receiver to extract the single PDUs. This header consists of an ID and a DLC (data length code) ([1], [2]). To abstract the data link frame for the Multi-

PDU-to-Frame Mapping a data link frame is modeled as a Container-PDU (figure 5).

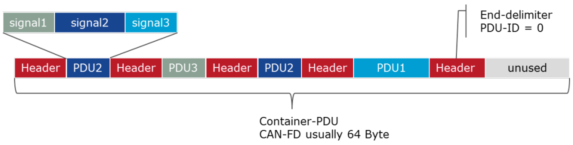

Figure 5: Container-PDU

Transmission behavior of Container-PDUs

With the dynamic construction of a frame the triggering for transmission must also be determined at runtime.
Many different triggers were defined to control the data transmission. This allows utilizing as much as possible of the payload without violating the communication timings.
When a PDU is added to a frame a transmission is triggered (figure 7):

1) after a certain container timeout
2) after a certain PDU timeout
3) immediately after a adding a PDU
4) after reaching the container threshold
5) when the container size will exceed

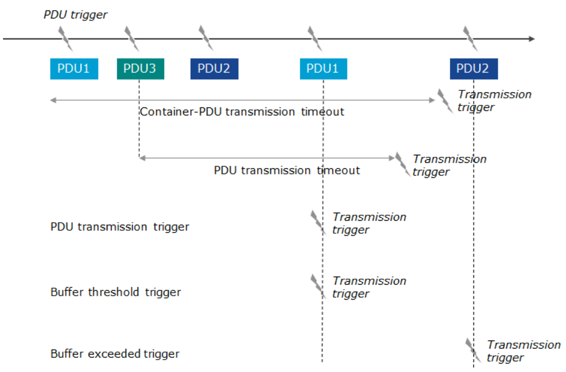

Figure 6: Transmission triggers

Furthermore the communication patterns last-is-best (figure 8) and queued (figure 9) are supported. The queued paradigm supports a FIFO transmitting each PDU update.

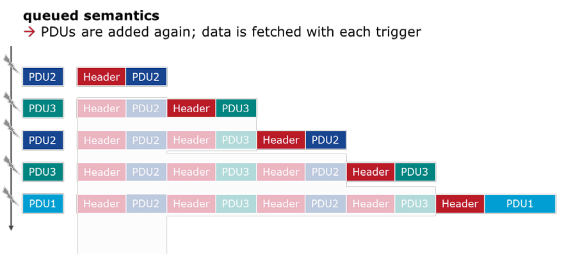

Figure 7: Queued communication

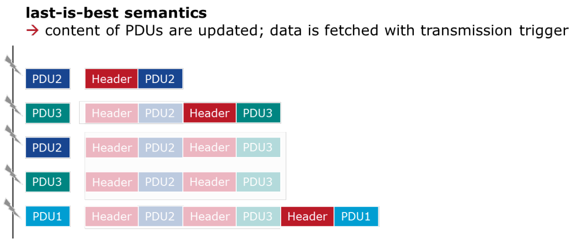

Figure 8: Last-is-best communication

This dynamic mapping can especially improve the variant handling of ECUs.

Benefits: Variety of the vehicle architecture

Today, using one ECU in different car lines usually requires different software variants dependent on the actual communication matrix.
With the new Dynamic Multi-PDU-to-Frame Mapping the communication gets more independent of the network design. As a consequence the sender can be moved within the vehicle without modifying any receiving node.
Unknown PDUs are skipped by the receiving node. Due to this, PDUs can be added to the vehicle without the need to update not affected ECUs.

Drawback: High resource demand

This approach opens a high flexibility in the variety of the vehicle architecture. But the drawbacks are higher interrupt and CPU load in each ECU. The reason for the higher load is that each ECU needs not only to receive frames but also has to analyze them whether relevant PDUs are contained. The receiving algorithm needs to check if the first PDU is relevant for reception. Then it needs to jump to the next PDU header checking if this PDU is relevant. This continues until the end of the frame is reached.
As a consequence the vehicle network design needs to consider this aspect and keep irrelevant messages away from the ECUs. The domain architecture with its domain controller could take care to forward only relevant messages. Hardware filters cannot be used anymore to reduce the interrupt load.

Gateway routing

Higher bandwidth causes higher loads in gateway ECUs, due to more data which needs to be processed and due to faster networks causing higher interrupt loads. The Dynamic Multi-PDU-to-Frame Mapping concept is relieving the gateway by copying the PDU as a whole in comparison to signal-routing the gateway (figure 10).

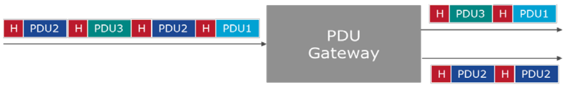

Figure 9: PDU routing

The signal routing (figure 11) requires de-serializing the PDU on reception and serializing a new PDU for transmission.

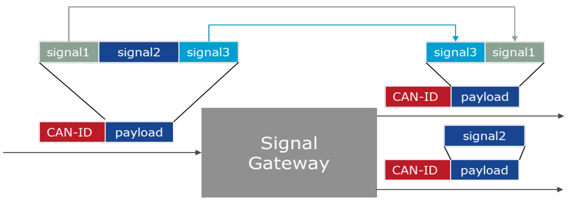

Figure 10: Signal Routing

Classic CAN vs multi-PDU concept

For classic CAN to classic CAN routing the most efficient routing is the message routing. The whole frame is just copied in one piece (figure 12). Optionally the CAN-ID could be modified.

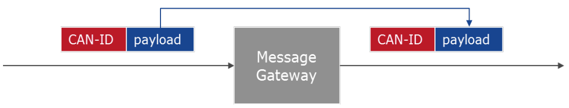

Figure 11: Message routing

Which routing from classic CAN to other networks is used depends on the transmitted data. In case identical PDUs are used the PDU routing could be applied (figure 13). In case a different PDU is used a signal routing needs to be applied.

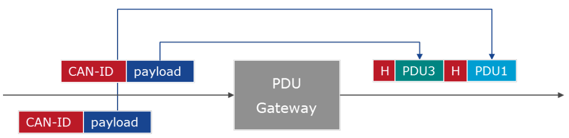

Figure 12: CAN to CAN FD routing

A new approach: Automotive Ethernet

The payload of one Ethernet frame is up to 1500 bytes. However the absolute overhead increases dramatically as well compared to CAN or FlexRay. This is why it is even more important for Ethernet to transmit as many data as possible within one frame. For this the Dynamic Multi-PDU-to-Frame Mapping concept is applicable analog to CAN FD.
Additionally Ethernet is a switched architecture which performs best with a peer to peer communication. The data is usually not broadcasted by the sender but it is sent individually to each receiver. This means that the Dynamic Multi-PDU-to-Frame Mapping should be done for each receiver separately.
Besides the classic communication of signals and PDUs a service oriented approach is available for Ethernet networks called SOME/IP.

SOME/IP

SOME/IP ([3]) was invented as a lean middleware covering the automotive needs. SOME/IP includes an efficient serialization pattern for complex data and Remote-Procedure-Calls (RPCs).
With Service Discovery ([4]) SOME/IP added a further communication principle: The publish-subscribe pattern. This allows creating the communication relations at runtime (figure 14). The notification events (data updates) are only transmitted to those receivers which subscribed to the corresponding publisher.

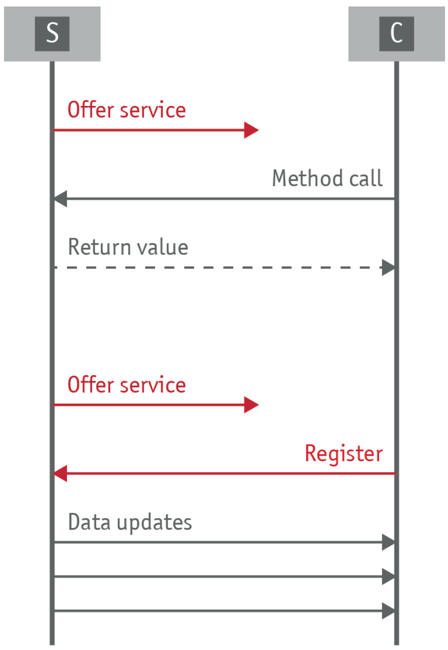

Figure 13: Operating principles of service-oriented communication using service discovery

One notification event is represented as one PDU of dynamic length. If multiple notification events are sent to a subscriber the publisher will typically make use of the Dynamic Multi-PDU-to-Frame Mapping.
To ensure interoperability the Dynamic Multi-PDU-to-Frame Mapping PDU Header was designed to match the first part of the SOME/IP header (figure 15).

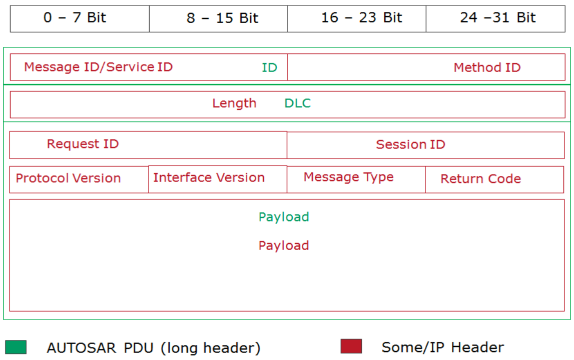

Figure 14: SOME/IP header – PDU header

Testing and analyzing

ECUs that make use of Dynamic Multi-PDU-to-Frame Mapping require tools that allow to test and analyze such ECUs on the PDU level. At the moment tools work mainly on the data link layer. This means that tools focus on frames. Therefore, analyze windows and test scripts usually show and analyze the communication based on the frames of the used data link protocol. As the different data link protocols highly differ in frame format and communication principles (e.g. event-triggered, time-triggered or switched) the tool architecture is kept separated (figure 16).

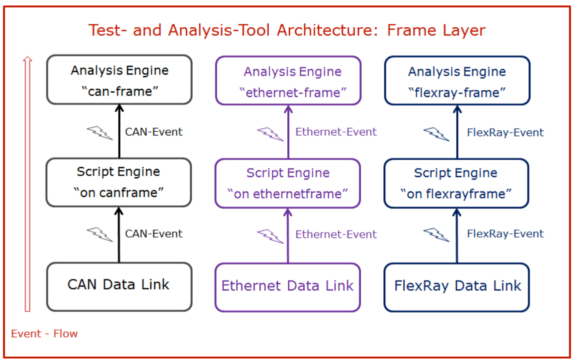

Figure 15: Frame based architecture

With the Dynamic Multi-PDU-to-Frame Mapping the task of the data link layer is more or less reduced to transport PDUs.

That means for a tool that a new communication layer must be implemented that is independent from the data link protocol and works the same way for CAN FD, Ethernet or FlexRay. An independent PDU layer has the advantage that users can concentrate on the communication data. This allows to write test scripts that work the same way e.g. for CAN FD and Ethernet. Also the representation of the data in analysis windows is the same and requires no deep knowledge of the used data link protocol.
This requires a tool architecture that provides new communication principles such as dynamic PDUs and that hides the data link layer from the user (figure 17).

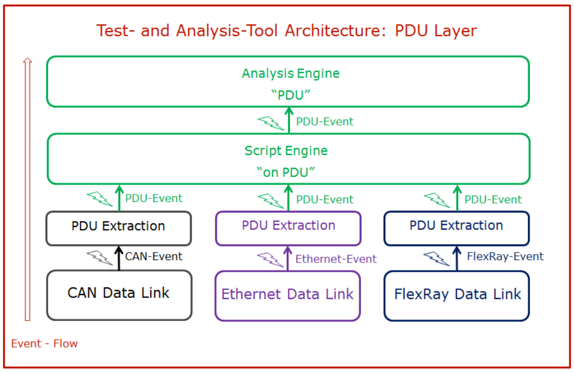

Figure 16: PDU based architecture

Summary

CAN FD and Ethernet require an abstraction between the signal and the data link layer to introduce new communication principles and to keep vehicle architectures flexible. The Dynamic Multi-PDU-to-Frame Mapping concept offers an ideal platform for these use cases. Even if there are drawbacks concerning performance, the concept convinces with its high flexibility especially in event driven environments. Also gateways benefit from the concept as it is mainly required to copy PDUs and not de-serialize signals any more.
For testing and analyzing such environments also new approaches within tools are necessary.

Oliver Garnatz
Vector Informatik GmbH
Ingersheimer Straße 24
70499 Stuttgart
Germany
Phone +49 711 80670 3615
E-mail oliver.garnatz@vector.com
Website www.vector.com

Peter Decker
Vector Informatik GmbH
Ingersheimer Straße 24
70499 Stuttgart
Germany
Phone +49 711 80670 4805
E-mail peter.decker@vector.com
Website www.vector.com

References

[1] Specification of I-PDU Multiplexer AUTOSAR Release 4.2.1
[2] Specification of PDU Router AUTOSAR Release 4.2.1
[3] Specification of SOME/IP Transformer AUTOSAR Release 4.2.1
[4] Specification of Service Discovery AUTOSAR Release 4.2.1