Implementing Next Generation Automotive Communications

Thomas Nolte and Hans Hansson
Mälardalen Real-Time Research Centre
Department of Computer Science
and Engineering
Mälardalen University
Västerås, SWEDEN

Lucia Lo Bello
Department of Computer Engineering
and Telecommunications
University of Catania
Catania, ITALY

Abstract

In-car electronics plays an important role in many automotive applications, such as, for example, steer-by-wire and brake-by-wire, and is expected to gradually replace mechanical or hydraulic means to control these systems. The number of electronic components in a car has therefore significantly grown up, thus leading important implications on the vehicle engineering process. In particular, in-car networks used to interconnect electronics equipments are a key point. While in the past extensive use of wiring was a common design practice, nowadays, for the sake of reducing the vehicle weight and fuel consumption, in-car bus networks are largely adopted.

This paper points out current automotive communication standards, i.e., CAN, LIN, Byteflight and MOST, together with upcoming automotive communication standards, namely, TT-CAN, TTP and FlexRay. The work focuses on discussing open issues with the current FlexRay implementation.

1. Introduction

The automotive system is nowadays a complex distributed system with various demands of networking capabilities. Most automakers share common subcontractors, and in a modern automotive application more and more applications developed by different subcontractors needs the possibility of interaction. One automotive application consists of one or several Electronic Control Units (ECUs). In an automotive system consisting of several automotive applications more than 70 ECUs might need to distribute more than 2500 signals. This makes the automotive system complicated in terms of networking. To manage this complexity the automotive industry has these last years set up several big consortiums in order to agree on a common scalable electric architecture (e.g., AUTOSAR [1]) and a common scalable communication system (e.g., FlexRay [2]) to support the automotive systems of tomorrow.

In the near future hydraulic automotive systems, such as steering and braking, will be replaced by communication networks (wires). The former is offered by subcontractors today, and the latter has been shown to work in several prototype cars. These new steer-by-wire and brake-by-wire solutions are commonly called X-by-Wire (XBW) systems.

In this paper, in Section 2, we discuss automotive communication requirements, together with the main automotive communication technologies currently in use as well as the most promising for tomorrow. Then, in Section 3, we discuss candidates for X-by-wire applications. Based on this, we present some open issues when implementing FlexRay-based automotive systems in Section 4 and we summarise the paper in Section 5.

2. Automotive Communication Requirements

The requirements of an automotive communication network come from the applications it has to support. Major automotive applications that involve networking and the usage of fieldbuses are XBW, chassis systems, powertrain systems and infotainment or multimedia. Looking at XBW systems, the two most important requirements are dependability and fault-containment. However, for chassis systems determinism is more important whereas high bandwidth is needed in powertrain systems. Multimedia devices require high bandwidth and plug-n-play capabilities.

Today several different fieldbuses are used to address these various communication demands. A big issue that the automotive system producers deal with is that too many fieldbus technologies exist today. To interconnect these systems there is a need for high bandwidth together with flexibility and determinism. It is desirable to move from using too many technologies to fewer more general. To reduce the complexity evolving in a modern automotive system it would be good to commit on a set of networking protocols that can be used in most of the applications typically found in an automotive system.

2.1. Technologies used today...

Some of the more common fieldbus technologies interconnecting ECUs today are the Controller Area Network (CAN), the Local Interconnect Network (LIN), Byteflight and the Media Oriented Systems Transport (MOST).

CAN - The Controller Area Network (CAN) [3] was initiated in 1981 and developed by Bosch. In 1994 it became an ISO standard [4] and today it is the most widely used vehicular network. CAN transmits message frames in an event-triggered fashion. Frames can be transmitted at speeds of up to 1 Mbps, although 500Kbps is the more common choice for, e.g., engine control, ABS systems and cruise control. Speeds of less than 125 Kbps are used for comfort electronics (e.g., seat and window control).

LIN - The Local Interconnect Network (LIN) [5] is a time-triggered master-slave fieldbus. With LIN, message frames are sent at speeds of 20 Kbps. The physical medium is a single wire. LIN is an open standard (standardised in 2000) inexpensive network that is used in automotive systems to control devices such as seat control, light sensors and climate control. LIN is often used together with CAN, as LIN complements CAN being much cheaper and simpler still providing the communications required for many automotive applications.

Byteflight - In 1996 BMW started to develop Byteflight [6], which is a Flexible Time Division Multiple Access (FTDMA) network typically using a star topology (although bus and cluster topologies are possible as well). Messages are sent in frames at 10 Mbps, using a minislotting concept (explained in Section 4.1). The physical medium used is plastic optical fibre. One of the strengths of the Byteflight protocol is its support for event-triggered message transmission, although both static and dynamic message frame transmissions are possible.

Byteflight guarantees deterministic (constant) latencies for a bounded number of high priority real-time messages. Moreover, it is possible to send non-real-time messages in a prioritised manner thanks to the Minislotting mechanism. Clock synchronization is provided by a dedicated master node, achieving clocks synchronized with a precision in the order of 100ns. Any Byteflight node can be configured as the master node. It is possible to avoid babbling idiots using a star coupler.

Typical applications supported by Byteflight are, for example, airbag systems and sear-belt tensioners, as they feature fast response-time requirements and short mission-times.

MOST - To support communications for multimedia application, MOST, or Media Oriented Systems Transport [7] is commonly used. MOST was initiated in 1997 and supports both time-triggered and event-triggered traffic with predictable frame transmission at speeds of 25 Mbps. Moreover, MOST will be available with speeds of 150 Mbps in the near future. MOST is using plastic optical fibre as communication medium.

Typical applications where MOST is used are, for example, the interconnection of telematics and infotainment, such as video displays, GPS navigation systems, active speakers and digital radios. MOST was the first networking technology that was approved by DVD-CCA to carry protected digital video streams.

2.2. ...and technologies for tomorrow

XBW systems need fault-tolerant communication with deterministic message transmissions and low jitter. This is traditionally solved using Time Division Multiple Access (TDMA) protocols due to their deterministic nature. Three of the more common TDMA-based fieldbuses for automotive applications are TT-CAN, TTP and FlexRay. However, as we will see, they differ in their bandwidth and fault-tolerant capabilities.

TT-CAN - TT-CAN, or Time-Triggered CAN [8], is a time-triggered session layer on top of CAN. This provides a hybrid TDMA on top of Carrier Sense Multiple Access (CSMA) allowing for both time-triggered and event-triggered traffic. TT-CAN is standardised by ISO [9] and intended for XBW systems, although it does not provide the same level of fault-tolerance as the other two XBW candidates, i.e., TTP and FlexRay (presented below).

TTP - TTP/C (TTP), or the Time-Triggered Protocol, is part of the Time-Triggered Architecture (TTA) by TTTech [10, 11]. The first TTP communication controller was released in 1998. TTP is a pure time-triggered TDMA protocol.

Using TTP, frames are sent at speeds of 5-25 Mbps depending on the physical medium. Research is going on to reach speeds of 1 Gbps using an Ethernet based star architecture. TTP uses both twisted pair and optical bus medium.

FlexRay - In 1998 BMW and Daimler-Chrysler analysed the current available automotive networks (e.g., CAN, TTP, MOST and Byteflight) and found out that none of those technologies would fulfil future needs of next generation automotive systems, especially when the automotive industry will take the next step towards XBW. Office-oriented communication protocols can not be used either, since they are not automotive qualified in terms of operating temperatures and electromagnetic requirements.

As a response to this, the FlexRay consortium [2] was formed with the goal to develop a new protocol, called FlexRay [12]. This new protocol should be the solution for the introduction of XBW systems as well as the replacement of some of fieldbuses currently used, thus reducing the total number of in-car networks.

FlexRay is expected to be the de-facto communication standard for high-speed automotive control applications interconnecting ECUs in future automotive systems. A special area of interest will be high-speed safety-critical automotive systems such as XBW and advanced powertrain applications.

FlexRay provides both time-triggered and event-triggered message transmission. The event-triggered message transmission is done in the same way as in Byteflight using minislotting. Messages are sent in frames using either single or dual channels (the latter for redundancy or doubling the bandwidth) at 10 Mbps and there are no limitations to increase this speed due to protocol mechanisms. At the physical layer, both electrical and optical solutions are adopted. Using FlexRay, the ECUs are interconnected using either a passive bus topology, the way most ECUs are connected today, or an active (multiple) star topology.

FlexRay complements CAN and LIN being suitable for both powertrain systems (high bandwidth and event-triggered traffic) and XBW systems (dependability and fault-containment).

3. Comparative assessments of candidates for XBW

In a modern automotive system there is a need for both time-triggered and event-triggered traffic. CAN is very good in handling event-triggered traffic, and today it can be found on many places in a car, ranging from engine-control and ABS-systems to body and comfort systems. CAN has also been extended with a time-triggered session layer, called TT-CAN, to support time-triggered traffic. This provided a smooth transition using CAN in the first generation "wet" XBW where a hydraulic backup is used. However, since it relies on the CAN lower layers, it is lacking fault-tolerant mechanisms and bandwidth capabilities. Hence, CAN/TT-CAN is not commonly suggested for XBW solutions today.

On the other hand, TTP is a highly fault-tolerant network developed and intended for safety-critical systems such as XBW and avionics. TTP is implementing fault-tolerant mechanisms, such as atomic broadcast using membership service, distributed clock synchronization and bus guardians. TTP ensures that there can be no single point of failure. TTP is very deterministic at the cost of being less flexible in terms of message transmissions compared with, e.g., FlexRay. A weak point here is that due to its inflexible message transmission and high cost, the use of another fieldbus would be needed for other applications in the car, such as the powertrain, where high bandwidth together with event-triggered capabilities are needed.

While TTP does not directly support event-traffic, FlexRay does, as it combines TDMA message transmission and the FTDMA of Byteflight, thus allowing for both time-triggered and event-triggered message transmissions. Moreover, FlexRay was developed with safety-critical applications in mind, just like TTP. Hence, using FlexRay it is possible to develop both XBW systems and powertrain systems, reducing the need for several fieldbus technologies.

Among the three protocols FlexRay has the biggest potential for becoming the next generation automotive network for safety-critical fault-tolerant applications, mainly because it is heavily backed up by industrial partners and pushed by most major industrial automakers where several have moved from being TTP-supporters to being FlexRay-supporters [13].

Although FlexRay is developed and mainly intended for safety-critical communications such as XBW, it is possible to use it in several other automotive applications. For this reason here we propose the implementation of higher-layer extensions on top of FlexRay, which will make it suitable for a broader class of automotive applications. The proposed extensions are especially devised for better supporting dynamic traffic scenarios.

4. FlexRay Communication

The FlexRay frames are sent in what is called a communication cycle. The communication cycle is divided into four parts: static segment, dynamic segment, symbol window and network idle time. In this paper we focus on the static and the dynamic segments.

The static segment contains static slots that are used for the transmission of statically scheduled frames. Multiple slots can be allocated to the same node in the same communication cycle, thus allowing for intra-cycle application level agreement. Note here that this feature is not available in the competing TTP technology, and therefore represents one of the strengths of FlexRay. The static segment is protected by a bus guardian, preventing the babbling idiot problem.

The dynamic segment contains minislots where dynamically scheduled frames are sent. This allows for dynamic bandwidth allocation (also during runtime), either on per-node or on per-channel basis. Note that there is no bus guardian mechanism here to protect the dynamic segment from babbling idiots.

4.1. Minislotting

With FlexRay, the dynamic segment is scheduled in a cyclic manner using a minislotting technique similar to Byteflight. As the dynamic segment is started, the minislotting mechanism is used to schedule the messages. In order for the minislotting mechanism to work, all messages must have unique identifiers (IDs), like in CAN. Moreover, all nodes in the system keep a slot-counter.

As a dynamic segment is started, all slot counters are reset to 0 indicating the start of the first dynamic slot. Whenever a node has a message with an ID matching the slot counter, the node will send its message. Once the message has been sent, then all the nodes in the system will detect a new dynamic slot and increase their slot counters by one. If there is no message transmitted within a short time Δ after the initiation of a dynamic slot (Δ being much shorter than the time needed to transmit a message), a new dynamic slot is detected and all the slot-counters are increased again, etc. In this way messages are scheduled in increasing order based on their IDs, serving the lower value IDs first. Hence, a low value ID gives a high priority to a message and thus a higher chance of being transmitted. A dynamic slot can be of varying size depending on the size of the message transmitted in the dynamic slot.

However, the slot-counters might not reach their maximum value due to the fact that the dynamic segment does not fit all possible messages. As the dynamic segment is of fixed size, depending on the number of dynamic slots used (by messages being transmitted) and their individual lengths, the slot-counter will only reach a certain value. Therefore some messages might have to wait one or more communication cycles in order to be scheduled. It should be noted here that the slot-counters always begin with value 0 for all dynamic segments.

4.2. Open Issues

Looking at the FlexRay protocol as it is defined today, there is a potential for some additions to address a broader class of automotive applications.

Support for dynamic addition and removal of traffic flows With FlexRay all periodic traffic is scheduled using the static segment of the communication cycle. Hence, a TDMA schedule is created for this and applied at system startup.

To allow for dynamic addition and removal of periodic traffic (without a need to reconfigure the static segment of FlexRay) and aperiodic/sporadic traffic, we propose to better exploit the dynamic segment by means of the introduction of admission control, together with software bus guardians (using, for example, the concept discussed in [14] where servers act as software bus guardians). Using the dynamic window of FlexRay in this way allows for open system design, where traffic flows can dynamically be added and removed during run-time. This requires both admission control and bandwidth isolation mechanisms. Bandwidth isolation can be achieved by implementing servers as bus guardians and traffic shapers. Admission control can be implemented in a higher layer, which should be developed on top of FlexRay. As a result, we point out a number of implementation issues to be addressed, such as, how the admission control protocol should be implemented, how to implement bus guardians and traffic shapers suitable for FlexRay, where should these implementations reside, etc.

It should be noted that these modifications should be done still keeping the composability property, i.e., they should not significantly affect the way the network is allowed to be used from an application point of view.

Increased fairness using cyclic service All the traffic sent in the dynamic segment of the communication cycle is scheduled using minislotting as explained before. This allows for a few high priority messages (i.e., those with lower IDs) to get hard real-time services by being scheduled in every communication cycle, and therefore getting deterministic transmission times with fairly low jitter. However, real-time messages might not have to be necessarily scheduled in every communication cycle, but still their worst-case transmission times should be deterministic.

Since there are a limited number of messages fitting in the dynamic segment of a communication cycle, it is possible for a few high priority messages to cause starvation of lower priority messages.

To remove such a potential starvation problem, here we propose the introduction of a cyclic service mechanism based on three classes of communication: high, medium and low priority messages. In each cycle a given number of messages for each class are allowed to be sent. To control this, we implement a cycle-counter. This cycle-counter is to be used to determine if a specific message is allowed to be sent or not. We need this cycle-counter in order to hold messages for a number of communication cycles.

Whenever a the slot-counter match a message ID, instead of sending the message as traditionally done, the cycle-counter is used together with the class of the message (high, medium or low) to determine whether the message is to be sent or held back.

In this way we can enforce that in each communication cycle there is only a subset of all possible messages competing for message transmission. Therefore we can limit the impact of the higher priority messages on the lower priority ones, still providing deterministic response times.

Complete architectures With the addition of FlexRay as an automotive network it is possible to use the network both for safety-critical applications as well as a backbone of a complete architecture. Within the whole automotive system LIN, CAN and FlexRay (and MOST) can then be combined. As new applications require these systems to be integrated, for example vehicle dynamics control, issues of interconnection arise, such as network gateways LIN/CAN, CAN/FlexRay and possibly LIN/FlexRay. Tunnelling of traffic might be desirable in some cases and in other cases global addressing could be useful. Global addressing can be done in a static way using tools that configure the whole system using the networks as they are, or by adding some higher layer addressing protocol creating global addresses. Ways of designing these new complete architectures needs to be discussed, as are tools for allocation and analysis of the network traffic.

5. Summary

In this paper we addressed the networking protocols that are found today in automotive systems and we have looked into the latest protocols intended for the next generation automotive applications, with particular focus on XBW systems. Among these protocols, FlexRay seems to be the most promising for a number of reasons.

Looking at FlexRay, we have investigated its communication capabilities and proposed some additions to be implemented on top of the protocol in order to make it able to cope with a broader class of communication requirements. The reason for extending FlexRay with more networking functionalities is the industrial wish for fewer networking technologies to be used in the automotive systems of tomorrow.

Another issue discussed is the possible complete architecture using FlexRay. Here the automotive networks need to be interconnected and able to provide communications across several networks. Gateways together with a higher layer addressing protocol could solve this. The design of these complete architectures is an issue which needs to be further elaborated on.

References

[1] AUTOSAR. http://www.autosar.org/.

[2] FlexRay Consortium. http://www.flexray.com/.

[3] Robert Bosch GmbH. BOSCH's Controller Area Network. http://www.can.bosch.com/.

[4] CAN. Road Vehicles - Interchange of Digital Information - Controller Area Network (CAN) for High-Speed Communication. *International Standards Organisation (ISO)*, ISO Standard-11898, Nov 1993.

[5] LIN Consortium. http://www.lin-subbus.org/.

[6] Byteflight. http://www.byteflight.com/.

[7] MOST Cooperation. http://www.mostcooperation.com/.

[8] CiA - Time-Triggered CAN. http://www.can-cia.de/can/ttcan/.

[9] TT-CAN. Road Vehicles - Controller Area Network (CAN) - Part 4: Time-Triggered Communication. *International Standards Organisation (ISO)*, ISO Standard-11898-4, December 2000.

[10] TTA-Group. http://www.ttagroup.org/.

[11] TTTech. Time Triggered Technology. http://www.tttech.com/.

[12] FlexRay Consortium. FlexRay Communications System - Protocol Specification. Version 2.0, June 2004.

[13] X-by-Wire Communications Protocol: Possible FlexRay and TTP Merger. *The Hansen Report on Automotive Electronics*, 14(4), May 2001.

[14] T. Nolte, M. Nolin, and H. Hansson. Hierarchical Scheduling of CAN using Server-Based Techniques. In 3^{rd} *International Workshop on Real-Time Networks (RTN 2004) in conjunction with the* 16^{th} *Euromicro International Conference on Real-Time Systems (ECRTS 2004)*, pages 17–20, University of Catania, Italy, June 2004.