Overview of Automotive Network Protocols

Orlin Stoyanov[1], Georgi Krastev[2], Aleksandar Stoyanov[3]

Abstract – **This paper describes some of the different network protocols used primarily in the automotive industry but also throughout the aeronautical and various industrial markets. The first paragraph is dedicated to the Controller Area Network (CAN), the system which is often acting, as the "backbone" of a set of other networks, and intends to familiarize the reader briefly with the current state of the CAN protocol. The second paragraph begins with a detailed description of the LIN bus, generally described as a sub-type of CAN. At the end, several other protocols operating under X-by-Wire application fitted in motor vehicles are considered.**

Keywords – **Automotive control, High Speed Protocol, Vehicle network.**

I. Introduction

During the last few years there is an observed increase in the number of electronic systems in the automobiles. The large quantity of sensors built in the automobiles requires each new function to be controlled by ECUs. The field of multiplexed buses is constantly being enlarged in line with the car manufacturers which are making developments in the order to improve the safety, comfort and reliability. With the increasing number of electrical equipment in the automobile the need of efficient networking system is rising too. The use of such in-vehicle network technologies is expected to reach new frontiers in the near future.

Some of the important in-car technologies and protocols divided into two general classes, CSMA (Carrier Sense Multiple Access) and TDMA (Time division multiple access), are described as follows:

II. In-vehicle Network Protocols

Prior to begin brief introduction of this protocols, let's have a general overview to some of them. Fig. 1 illustrates the performance/cost ratio and the relative importance and uses of each protocol. It is not surprising that the faster the system, the higher the price to be paid. Of course, the relative costs shown on the horizontal axis are given for guidance only, with the usual reservations.

A modern vehicle includes a large number of intersection and interlacing layers of links for connecting multiple electronic control units, on-board systems and entertainment applications. To take more accurate idea of the situation, the diagram on the Fig. 2 shows architecture of a high class vehicle.

The CAN bus (Controller Area Networking) was defined in the late 1980 by Bosch, initially for use in automotive applications, but which is coming into use for linking distributed controllers and sensors in other fields. The official CAN specification has been released by ISO as 11898-1 (CAN data link layer protocol) [1].

CAN is a CSMA/CD protocol. It supports speeds of up to 1Mb/s so is an SAE class C protocol, suitable for real time control applications.

The CAN bus is a broadcast type of bus. This means that all nodes can receive the transmitted messages. They are not addressed to intended recipients, but message itself include the sender's identifier. This provides local filtering so that each node may react only on the messages with the correct identifier. Messages' identifiers give the priority of the message, so the priority of messages is decided at the design stage. There are two standards for CAN 2.0, called A and B. Part A describes the most common "standard" CAN frame. This frame supports 11 identifier bits. Part B is intended to describe the CAN frame in its "extended format". Because it was insufficient for some applications, the identifier value was changed from 11 to 29 identifier bits. The CAN 2.0A and 2.0B versions were designed to provide compatibility with any earlier version of the protocol. In full CAN, the CAN devices add filtration of the messages, and will only pass messages with specified identifiers on to its associated controller, so a controller is only interrupted by those messages the CAN terminal passes, that is those of interest to that controller [7,9].

The bus uses non-return to zero (NRZ) coding with bit-stuffing. The modules are wired to the bus: if just one node is driving the bus to a logical 0, then the whole bus is in that state regardless of the number of nodes transmitting a logical 1.

The CAN standard defines four different message types. The messages use a bit-wise arbitration to control access to the bus, and each message is tagged with a priority.

The scope of the CANbus protocol covers the physical and data link layers of the ISO/OSI model [9].

[1]Orlin Stoyanov,Faculty of Electrical and Electronic Engineering and Automation, Studentska St, 7017, Ruse, Bulgaria, E-mail: ostoyanov@uni-ruse.bg.

[2]Georgi Krastev,Faculty of Electrical and Electronic Engineering and Automation, Studentska St, 7017, Ruse, Bulgaria, E-mail: GKrastev@ecs.ru.acad.bg.

[3]Aleksandar Stoyanov, Faculty of Automotive and Transport Engineering, Studentska St, 7017, Ruse, Bulgaria,

E-mail:astoyanov@uni-ruse.bg

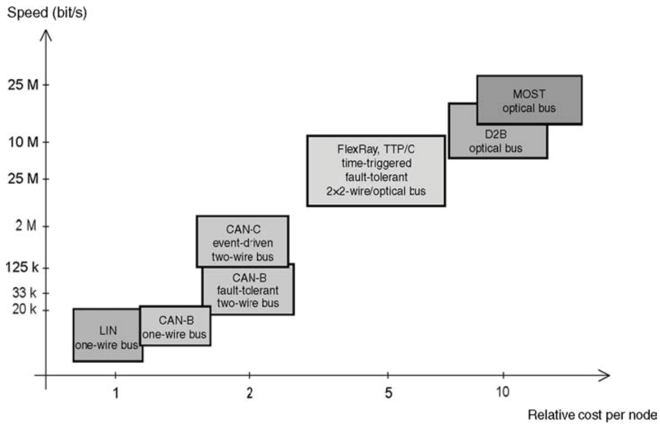

Fig.1Relative cost per node of automotive networks

LIN stands for Local Interconnect Network and is a low cost field bus. It is mainly intended to support the control of “mechatronic” elements found in distributed systems for motor vehicle applications, but of course it can be applied in many other fields. The LIN protocol concept is a multiplexed communication system whose level and associated performances fit below CAN's functionality. The primary and original purpose of LIN is therefore to provide a “sub-division” for CAN, with reduced functionality and lower costs.

The major developer of the LIN concept was the Motorola Company. A consortium was created, in March 2000, including the car manufacturers Audi, BMW, Daimler Chrysler, Volkswagen and Volvo Car Corporation, as well as Motorola Inc. and Volcano Communication Technologies AB.

LIN has single master/multiple slave architecture, therefore no need for arbitration. Speed is 20Kbit/s and it is specified to be most appropriate for SAE class A applications, the speed is actually at the lower end of class B. As it is time triggered, message latency is guaranteed. [7,8].

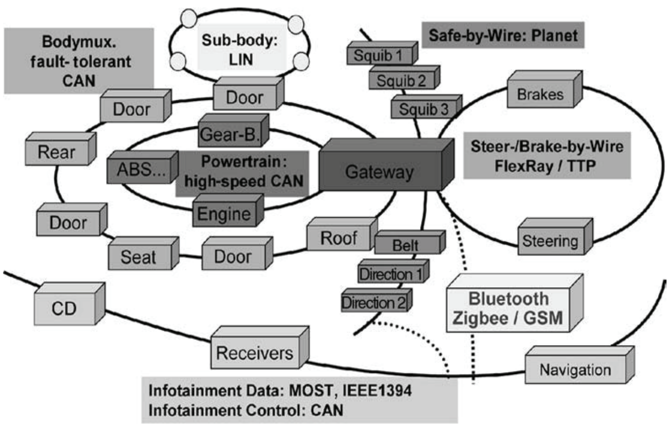

Fig.2 Conventional architecture of a high class vehicle

The increasing complexity of onboard systems arouses a demand for protocols providing “real-time” responses, deterministic solutions and high security. TTCAN (time-triggered CAN) was proposed by the CAN in Automation (CiA) group and the Bosch company at the beginning of 2002, which allowed CAN to be used for time triggered messages and increased determinism, reliability and synchronization over CAN.

TTCAN is a protocol layer at a higher level than CAN itself, without any modification of the data link layers (DLL) and physical layers (PL) of the latter. TTCAN is located primarily in the session layer of the OSI/ISO (International Standardization Organization) model. The aim of TTCAN is to keep the latency of each message to a specified value, independently of the load on the CAN bus itself. This protocol can be implemented on two levels:

- level 1 is limited to the transfer of cyclic messages;
- level 2 supports what is known as a ‘global time’ system.

TTCAN is based on a deterministic temporal exchange, based on a time window of a predetermined operating cycle, whose global operation can be represented in the matrix of rows and columns, which summarizes the general operating principle of this protocol.

All the messages travelling in the network between the CAN nodes are organized as elements of an X_Y matrix. This matrix time system consists of time windows organized in “basic cycles”, identical time values (represented as the totality of each row of the matrix) and numerous time intervals (windows) during which transmissions are authorized (represented by the columns of the matrix). Thus, it defines the relationship between the time windows and the presence of messages in the network.

The TTCAN operating principle is based on the fact that one of the nodes of the network is responsible for organizing the time division and allocation involved. This is because, when the system starts up, one of the nodes allocates the reserved time phases to each of the others. The system thus becomes deterministic, as each node has the right to transmit at a precise moment known to it, for a closely specified period. Clearly, this does not constitute a real-time system at all, but if the complete cycle is executed quickly enough, there is a rapid return to the same node, and to all the participants this appears to be “real-time” network access [2,7].

The TTP/C system is a member of the family of time-triggered protocols. The “/C” indicates that it meets the criteria of class C of the SAE – Society of Automotive Engineers – for the real-time and fault-tolerant aspects of communication in the car industry. It was developed by Professor Hermann Kopetz of the Vienna University of Technology, Austria and subsequently adopted by the TTTech – Time-Triggered Technology Company. It was not originally intended for motor vehicle applications, but for industrial applications in general. TTPTM/C is designed on the principle of TDMA (time division multiple access) to the medium.

All activities are carried out at certain points in time, decided at system design time, rather than network activities being triggered by external events, as in a CSMA protocol such as CAN. As TTP is a TDMA protocol, latency is deterministic. There is a bus guardian that “guarantees" that no node can control communication media outside its transmission slot. This principle can resolve problems of interoperability between CPUs developed independently of each other. [4,7,10].

Volcano might be described as "TTP on CAN" and the Volcano web-site describes the protocol as CAN-based and deterministic. The protocol is used by Volvo on the S80 and V70 cars, and is coming into use on Volvo buses. According to the Volcano Communications Concept, Volcano appears to be a technique in which the CAN network is integrated in such a way as to guarantee the latency of all the messages. It does this by specifying the latency and periodicity of messages at design time. This allows the maximum latencies to be calculated, so the system designer can specify the network set up in such a way as to modify these specifications to guarantee the specified parameters, by avoiding arbitration as far as possible. This seems to imply that the sending of network communication is time triggered rather than event triggered, so the description "TTP on CAN" seems a pretty good summing up.

This apparently means that network loadings can be considerably higher than using CAN conventionally, maybe 60% loading, whereas for latency of lower priority messages to be contained to reasonable limits, CAN loading may need to be around 10%.

FlexRay originated from the formation of a group of companies which conduct a profound technical analysis of the existing networks used in the automobile industry, namely CAN, TTCAN, TCN, TTP/C and Byteflight. The purpose of the analysis was to discover whether any one of them was capable of meeting all the technical and application requirements stated above. It was concluded that this study clearly showed that there is a lot more to be developed in this specific area, leading to the development of a new proposal, called ‘FlexRay’. The findings concerning the existing solutions are summarized below:

FlexRay was designed to provide a communication system in which collisions for access to the medium are impossible; in other words, the nodes are not subject to arbitration on the transmission channel, and collisions should not occur in normal operation. However, collisions may arise during the starting phase of the protocol on the transmission medium. The physical layer does not provide any means of resolving these collisions, and therefore the application layer must take over to handle such problems.

In order to provide the system with the greatest flexibility of application, it is necessary to communicate at a exact known instant for a known length of maximum time (operate in “real time”) with the assurance of being the only station present at that moment on the physical communication medium, making collisions not possible; to allow communication at a variable bit rate if required, and thus to require an unspecified communication time.

Communication in FlexRay takes place with the aid of recurring communication cycles. Each includes a ‘static segment, a ‘dynamic segment’, an ‘optional symbol window’ and a phase in which the network is in idle mode, which is called the network idle time (NIT). This cycle is initialized by the network manager node.

III. Conclusion

The need of high technology automobile networks is critical considering the growing dependence of the automobiles on the smooth functioning of the electronics. The car manufacturers and suppliers are working together with the aim to standardize the in-vehicle network protocols. The growing demand for safety, comfort, reliability and entertainment requires incorporation of multiple protocols. This tendency is expected to continue with the manufacturers’ implementation of more and more electrical devices in the automobile with the purpose of increasing the value of their products and meeting the requirements of the changing industry.

References

[1] CAN in Automation (CiA): CAN protocol specification http://www.can-cia.org/index.php?id=164

[2] CAN in Automation (CiA): Time-Triggered CAN http://www.can-cia.org/index.php?id=166

[3] FlexRay specification: http://www.flexray.com/

[4] Kopetz H., Real-Time Systems: Design Principles for Distributed Embedded Applications. Kluwer Academic Publishers, Boston, 1997

[5] Lawrenz, W. CAN System Engineering: From Theory to Practical Applications., 1997.

[6] Navet, N., F. Simonot-Lion. Automotive Embedded Systems Handbook. Taylor & Francis Group, LLC, 2009.

[7] Paret, Dominique. Multiplexed Networks for Embedded Systems. CAN, LIN, Flexray, Safe-by-Wire. John Wiley & Sons, Ltd, 2006.

[8] Rey, S. Introduction to LIN (Local Interconnect Network). Revision 1.0, 2003.

[9] The can protocol tourhttp://www.kvaser.com/can/protocol/index.htm

[10] TTTech website, http://www.tttech.com