An Analytical Study of Communication Protocols Used in Automotive Industry

Conference Paper · February 2014

CITATION
1

READS
1,327

3 authors:

Rohit Mathur
Institute for Plasma Research
15 PUBLICATIONS **134** CITATIONS
SEE PROFILE

Ritesh Saraswat
JIET Group of Institutions
1 PUBLICATION **1** CITATION
SEE PROFILE

Gunjan Mathur
JIET Group of Institutions
6 PUBLICATIONS **19** CITATIONS
SEE PROFILE

An Analytical Study of Communication Protocols Used in Automotive Industry

Rohit Mathur[1], Ritesh Saraswat[2], Gunjan Mathur[3]
Department of Electronics and Communication
Jodhpur Institute of Engineering and Technology
Jodhpur, India
[1]r4rohitmathur@gmail.com, [2]ritesh.saraswat@jietjodhpur.com, [3]er_gunjan_mathur@yahoo.com

***Abstract*— Communication protocols in-vehicle networking is a method for transferring data among distributed modules via a serial data bus or by mean of wireless. The use of systems for communications between the electronic control units (ECU) of a vehicle in production cars dates from the beginning of the 1990s. The specific requirements of the different car domains have led to the development of a large number of automotive systems such as Local Interconnect Network (LIN), J1850, CAN, TTP/C, FlexRay, media-oriented system transport (MOST), IDB1394, etc. This paper presents the study on character and performance of various protocols which is widely used in automotive industry and automation systems. Then, a comprehensive review of the most widely used automotive systems is discussed.**

***Keywords*— Electronic Control Unit, CAN, VAN, FlexRay, LIN**

I. INTRODUCTION

Electronic control units (ECUs) have been equipped almost in all the vehicles and the interaction between these subsystems has increased the need for real time communication within a vehicle. It is impossible to use dedicated signal wires because of cost, reliability and repair problems. In-vehicle schmoosing (also known as multiplexing) is a method to solve this problem. This can decrease dedicated wires required for each function and reduces the size of the wiring harness. The cost of embedded system, its weight along with the reliability, serviceability and installation of the system, can be improved by using various communication protocols. Sensor data like vehicle swiftness, engine heat, etc. are available on the network making data shared on the network which eliminates the need for redundant sensors because of effective communication protocol.

II. AUTOMOTIVE COMMUNICATION SYSTEMS: PAST AND PRESENT

A. From Point-to-Point to Multiplexed Communications

At the beginning stage of automotive electronics each new task was implemented as an individual electronic control unit (ECU), interpreted as a microcontroller based subsystem. This approach was not appropriate and sufficient to cater the need for function distribution and exchange over several ECUs. Consider a case where the vehicle speed predicted by the engine controller or by wheel rotation sensors has to be identified to adapt the steering energy, to control the suspension, or to choose the right wiping speed. "In modern luxury cars, up to 2500 signals (i.e. elementary information such as the speed of the vehicle) are exchanged by up to 70 ECUs" [1]. Until the beginning of the 1990s, exchange of data was done by point-to-point links between ECUs, which required an amount of communication channels, was not able to manage the increasing use of ECUs because of the weight and cost problems, reliability and complexity caused by the wires and the connectors. These issues encouraged the use of networks where the communications are shared over a medium, this requires defining rules or protocols for managing communications and granting bus access.

B. The Vehicle Domains and Their Evolution

Since all the functions embedded in vehicles have different performance or safety needs, different Quality of Services (e.g. response time, jitter, bandwidth, redundant communication channels for tolerating transmission errors, efficiency of the error detection mechanisms etc.) are sought from the communication systems.

The control of chassis components with regard to steering or breaking and driving conditions (ground surface, wind etc.) is done by functions such as ABS, ESP, ASC (Automatic Stability Control), 4WD (4 Wheel Drive) gathered by the chassis domain. Requirements for communication in this domain are very much similar to those for the powertrain but, as they have a high effect on the vehicle's stability, dynamics and agility, the chassis functions are more critical from a safety standpoint. Furthermore, the "x-by-wire" technology, currently used for avionic systems, is now being introduced to execute steering or braking functions. "X-by-wire" refers to the replacement of mechanical or hydraulic systems by fully electrical/electronic systems, leading to new design methods for their development and understanding the interferences between functions[2],[3].

Since all the nodes do not require a large bandwidth, therefore design of low-cost networks such as Local Interconnect Network (LIN) and TTP/A is done. On these networks only master node possesses an accurate clock and drives the communication by controlling the slaves periodically. The combination of diverse communication needs within the body domain lead to a hierarchical network architecture. Integrated mechatronic subsystems based on low-

cost networks are interconnected through a CAN backbone in a hierarchical network architecture. Body functions are mainly activated according to the driver and passengers' solicitation (e.g., opening a window, locking doors etc.). Telematics functions are becoming more and more numerous: hands-free phones, car radio, CD, DVD, in-vehicle navigation systems, rear seat entertainment, remote vehicle diagnostics etc.

III. VEHICAL NETWORK AND COMMUNICATION PROTOCOLS

The steadily increasing need for bandwidth and the diversification of performance, costs and dependability requirements lead to a diversification of the networks used throughout the vehicles. The Society for Automotive Engineers (SAE) classified automotive communication protocols [5], [6] as per data transmission speed and functions distributed over the network

A. SAE Classification

SAE Vehicle Network for multiplexing and data communication committee has defined basic categories of in-vehicle networks based on network speed and functions:

- Class A Low Speed (<10kb/s); Convenience features (entertainment, audio, trip computer, etc.)
- Class B Medium Speed (10kb/s to 125 kb/s); General information transfer (instrument cluster, vehicle speed, legislated emission data, etc.)
- Class C High Speed (125kb/s to 1M b/s or greater); Real-time control (powertrain control. Vehicle dynamics, break by wire, etc.)
- Class D higher speed (>1M b/s); Applied to more strictly real-time control and multimedia system.

B. Priority Buses Protocols

To ensure the relevancy of the exchanged data at runtime and the timely command delivery to actuators, it is vital ensure bounded response times of frames using Medium Access Control (MAC) protocol. To ensure this, MAC scheme that grants bus access according to the priority of the messages is developed. This serves two purposes: giving priority for transmission and allowing message filtering upon reception.

The two main representatives of such "priority buses" are CAN and J1850.

1) CAN

CAN is the most widely used in-vehicle network. CAN communication network was designed by Bosch (1980) for multiplexing communication between various ECUs in vehicles. This decreases the overall wire harness including wire lengths and numbers, "e.g. the number of wires has been reduced by 40%, from 635 to 370, in the Peugeot 307, which embeds two CAN buses with regard to the non-multiplexed Peugeot 306" [12]. CAN also allows to share sensors among different ECUs.

Currently, CAN is used as an SAE class C network for real-time control in the chassis domains and powertrain (at 250 or 500 kb/s). It can also be used as an SAE class B network for the electronics in the body domain, at a data rate of 125 kb/s. A CAN frame is characterized by an identifier, transmitted within the frame Fig. 1, whose numerical value determines the frame priority. There are two versions of the CAN protocol differing in the size of the identifier: CAN2.0A (or "standard CAN") with an 11-b identifier and CAN 2.0B (or "extended CAN") with a 29-b identifier. For in-vehicle communications, only CAN 2.0A is used, since it provides a sufficient number of identifiers (i.e., the number of distinct frames exchanged over one CAN network is lower than 211).

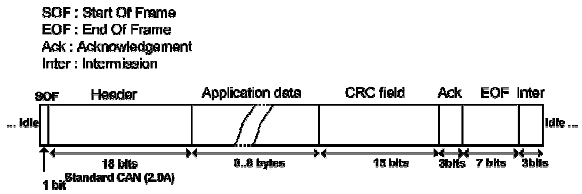

Fig. 1. Format of the CAN2.0A data frame

CAN uses non-return-to-zero (NRZ) bit representation with a bit stuffing of length 5, to establish a bit time (the time between the emissions of two successive bits of the same frame), stations need to resynchronize themselves after a certain period, and this process requires boundaries on the signal. Bit stuffing is referred as an encoding method that enables resynchronization when using NRZ bit representation. Edges are produced into the outgoing bit stream in the way to restrict the transmission of excess number of consecutive equal-level bits (five for CAN). The receiver employs the inverse procedure and de-stuff the frame. CAN requires the physical layer to implement the logical "and" operator: if at least one node is transmitting the "0" bit level on the bus, then the bus is in that state regardless if other nodes have transmitted the"1" bit level. For this reason, "0" is termed the dominant bit value, while "1" is the recessive bit value.

The standard CAN data frame may contain up to 8 bytes of data for an overall size of at max 135 bits which includes all the protocol overheads such as the stuff bits. The CAN frame has following sections:

The header field containing frame identifier, the Remote Transmission Request (RTR) bit responsible for distinguishing between data frame (RTR set to zero) and data request frame (RTR set to 1)with the data length code (DLC) which informs about the number of bytes of the data field. The data field with a maximum length of 8 B. The cyclic redundancy check (CRC) field of 15-bits for ensuring the integrity of the data transmitted. The Acknowledgment field (Ack) which enables the sender to that data has been received by at least one station. The end-of-frame (EOF) field and the intermission frame space, which is the minimum number of bits separating consecutive messages.

CAN contains several mechanisms for error detection. For example, it is checked that the CRC transmitted in the frame is similar to the CRC computed at the receiver end, that the structure of the frame is valid, and that no bit-stuffing error occurred. Each station which detects an error transmits an "error flag," which is a particular type of frame consisting of

six consecutive dominant bits that allows all the stations on the bus to be aware of the transmission error. The corrupted frame automatically re-enters into the next arbitration phase, which make it to miss its deadline due to the additional delay. The error recovery time, is expressed by the time from detecting an error until the possible start of a new frame is 17–31 bit times. The main drawback is that a node has to diagnose itself, which can lead to the non-detection of some critical errors. For example, a faulty oscillator can cause a node to transmit constantly a dominant bit, which is one appearance of the "babbling idiot" fault. Also, other faults such as the subdividing of the network into several sub networks may stop all nodes from communicating due to bad signal reflection at the edges. Without additional fault-tolerance facilities, CAN is not appropriate for safety-critical applications such as some future x-by-wire systems. For illustration, a single node can trouble the functioning of the whole network by sending messages outside their specification (i.e., length and period of the frames). Many mechanisms were proposed for increasing the dependability of CAN-based networks, if each proposal solves a particular problem, they have not been considered to be combined. Moreover, the fault theories used in the design of theses mechanisms are not essentially the same, and the interactions between them remain to be studied in a formal way.

The CAN standard only support the physical layer and data link layer (DLL). Several higher level protocols have been proposed, for example, implementing data segmentation, standardizing start up procedures, or sending periodic messages. Other higher level protocols standardize the content of messages in order to comfort the interoperability between ECUs. This is the case for J1939, which is used for instance, in Scania's trucks and buses [15].

2) Vehicle Area Network (VAN)

VAN is similar to CAN (in case of frame format, data rate) but possesses some additional features that are beneficial from a technical point of view (no need for bit stuffing; in-frame response: a node being asked for data answers in the same frame that contained the request). VAN was used in production cars by the PSA (Peugeot–Citroën), French carmaker in the body domain, but, as it was not adopted by the market, it was abandoned in favor of CAN.

3) The J1850 Network

The J1850 is an SAE class B priority bus [9], adopted in the United States for communications with no strict real-time requirements, such as the control or diagnostics of body electronics. J1850 are defined in two formats: 10.4-kb/s single-wire version and 41.6-kb/s two-wire version. The trend in new designs appears to be the replacement of J1850 by CAN or a low-cost network such as LIN.

C. Time-Triggered Networks

Among communication networks, as discussed before, one distinguishes time-triggered networks, where activities are motivated by the progress of time and event-triggered ones, where activities are motivated by the occurrence of events. Both types of communication have advantages but one considers that, in general dependability is much easier to ensure using a time-triggered bus [4]. This explains that, currently only time-triggered communication systems are being considered for use in x-by-wire applications. In this category, multi-access protocols based on TDMA are particularly well suited; they provide deterministic access to the medium (the order of the transmissions is defined statically at the design time), and thus provide bounded response times.

1) The TTP/C Protocol

The time-triggered protocol TTP/C, which is defined in [11], is a central part of the Time-Triggered Architecture and it possesses numerous features and services related to dependability, such as the bus guardian (components that prevent a node from transmitting outside its specification, for instance, at the wrong time or sending a larger frame), the group membership algorithm (knowledge of the set of stations that are functioning properly), and support for mode changes i.e. specific operational phases of an application. [20]

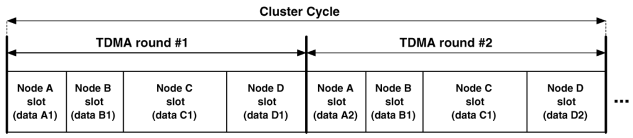

Fig. 2. Example of TTP/C communication cycle with four nodes A, B, C and D

On a TTP/C network, transmission relies on redundant channels and each channel transports its own copy of the same message. Although EMI is likely to affect both channels in quite a similar manner, the redundancy provides some resilience to transmission errors. TTP/C can be implemented with a bus topology or a star topology. The latter topology provides better fault tolerance, since, in the star topology, guardians are integrated into central star couplers and protect against errors that cannot be avoided by a local bus guardian. For instance, a star topology is more resilient to spatial proximity faults (i.e., faults that affect all components located in a given area, such as temperature peaks) and to faults due to a desynchronization of an ECU. To avoid a single point of failure, a dual star topology should be used with the drawback that the length of the wires is significantly increased. At the MAC level, the TTP/C protocol implements a synchronous TDMA scheme: the stations (or nodes) have access to the bus in a strict deterministic sequential order and each station possesses the bus for a constant period called a slot, during which it has to transmit one frame. The sequence of slots such that all stations have accessed the bus one time, is called a TDMA round. An example of a round is shown in Fig. 4. The size of the slot is not necessarily identical for all stations in the TDMA round, but a slot belonging to one station is the same size in each round. Consecutive TDMA rounds may differ according to the data transmitted during the slots, and the sequence of all TDMA rounds is the "cluster cycle" which repeats itself in a cycle.

TTP/C defines three types of frames:

- The "cold start frame," solely used at the initialization of the network.
- The data frame with explicit C-State. The C-State is a field that indicates the internal state of the communication controller: current time, frame being transmitted, current functioning mode of the cluster, membership vector, etc. This information is needed by stations willing to integrate the cluster at startup or reintegrate it at runtime.
- The data frame with implicit C-State. In that case, the C-State is not explicitly transmitted, but the receiver can still detect if it disagrees with the sender on the C-State, since the CRC is computed on the fields of the frame plus the C-State.

A TTP/C frame is composed of a field for indicating mode change requests, of application data, of a CRC and, depending on the frame type, of the C-state. A data frame can carry a payload of up to 240 B [19] but, at the time of writing, the "compatibility layer" specification, which defines the exact format of the frame, is not publicly available for the latest version of the protocol [23].

2) FlexRay Protocol

The FlexRay network is very flexible with regard to topology and transmission support redundancy. It can be organized as a bus, a star, or a multistar. It is neither compulsory that each station possess replicated channels nor a bus guardian, even though this must be the case for critical functions such as steer-by-wire. At the MAC level, FlexRay defines a communication cycle as the concatenation of a time-triggered (or static) window and an event triggered (or dynamic) window.

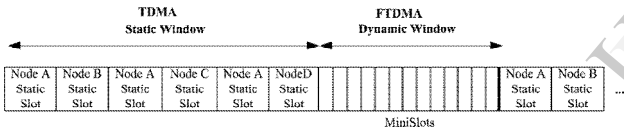

Fig. 3. Example of FlexRay communication cycle with 4 modes A, B, C and D

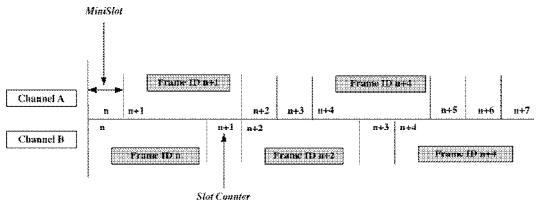

Fig. 4. Example of message scheduling in the dynamic segment of the FlexRay communication cycle.

The time-triggered window uses a TDMA MAC protocol; the main variance with TTP/C is that a station in FlexRay might possess several slots in the time-triggered window, but the size of all the slots is identical.

3) In the event-triggered part of the communication cycle, the protocol is Flexible TDMA (FTDMA): the time is divided into minislots, each station possesses a given number of minislots, and it can initiate the transmission of a frame inside each of its own minislots. A minislot remains idle if the station do not transmit which actually induces a loss of bandwidth. An example of a dynamic window is shown in Fig. 4: on channel B, frames have been transmitted in minislots and while minislot has not been used. It is notable that frame is not received simultaneously on channels A and B, since, in the dynamic window, transmissions are independent in both channels.

The FlexRay MAC protocol is more flexible compared to the TTP/C MAC, since in the static window nodes are assigned as many slots as necessary (up to 2047 overall) and since the frames are only transmitted if essential in the dynamic part of the communication cycle. In a similar way as with TTP/C, the structure of the communication cycle is statically stored in the nodes; however, unlike TTP/C, mode changes with a different communication schedule for each mode are not possible.

The FlexRay frame is made up of three parts: the header, the payload segment, and the CRC of 24 b. The header of 5 Bytes includes the identifier of the frame and the data payload length. The use of identifiers permits to move a software component, which sends a frame, from one ECU to another ECU without altering anything in the nodes that consume frame. It is to be noted carefully that this is not possible when signals produced by distinct components are combined into the same frame for the purpose of saving bandwidth.

3) Time-Triggered CAN (TTCAN) Protocol

TTCAN uses the CAN standard but, in addition, requires that the controllers must have the possibility to disable automatic retransmission of frames upon transmission errors and to provide the upper layers with the point in time at which the first bit of a frame was sent or received [18].

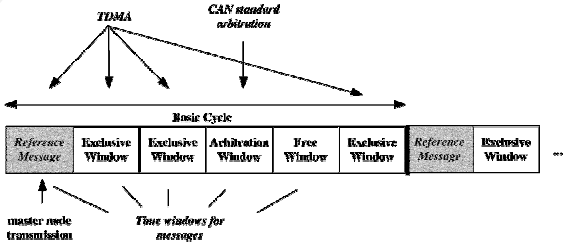

Fig. 5. Example of TTCAN basic cycle

The bus topology of the network, the characteristics of the transmission support, the frame format, as well as the maximum data rate1 Mb/s are imposed by the CAN protocol. Channel redundancy is possible [14], but not standardized and no bus guardian is implemented in the node. The key idea is to propose, with FlexRay, a flexible time-triggered/event-triggered protocol. As illustrated in Fig. 5, TTCAN defines a basic cycle as the concatenation of one or several time-triggered (or "exclusive") windows and one event-triggered (or "arbitrating") window. Exclusive windows are devoted to time-triggered transmissions (i.e., periodic messages), while the arbitrating window is ruled by the standard CAN protocol: transmissions are dynamic and bus access is granted according to the priority of the frames. Several basic cycles that differ by their organization in exclusive and arbitrating windows and by the messages sent inside exclusive windows can be defined.

The list of successive basic cycles is called the system matrix, which is executed in loops. Interestingly, the protocol enables the master node to stop functioning in TTCAN mode and to resume in standard CAN mode. Later, the master node can switch back to TTCAN mode by sending a reference message.

TTCAN is built on a well-mastered and low-cost technology, but as defined by the standard does not provide important dependability services such as the bus guardian, membership service, and reliable acknowledgment. It is, of course, possible to implement some of these mechanisms at the application orMWlevel but with reduced efficiency. Probably, carmakers might consider the use of TTCAN for some systems during a transition period until FlexRay technology is fully mature.

D. Low-Cost Automotive Networks

Several fieldbus networks have been developed to fulfil the need for low-speed/low-cost communication inside mechatronic-based subsystems generally made of an ECU and its set of sensors and actuators. Two representatives of such networks are LIN and TTP/A. The low-cost objective is achieved not only because of the simplicity of the communication controllers but also because the requirements set on the microcontrollers driving the communication are reduced (i.e. low computational power, small amount of memory, low-cost oscillator). Typical applications involving these networks include controlling doors (e.g. door locks, opening/closing windows) or controlling seats (e.g., seat position motors, occupancy control). Besides cost considerations, a hierarchical communication architecture, including a backbone such as CAN and several sub networks such as LIN, enables reducing the total traffic load on the backbone.

Both LIN and TTP/A are master–slave networks where a single master node, the only node that has to possess a precise and stable time base, coordinates the communication on the bus: a slave is only allowed to send a message when it is polled. More precisely, the dialogue begins with the transmission by the master of a “command frame” that contains the identifier of the message whose transmission is requested. The command frame is then followed by a “data frame” that contains the requested message sent by one of the slaves or by the master itself

1) LIN

LIN is a low-cost serial communication system used as SAE class a network [7, 8], where the needs in terms of communication do not require the implementation of higher bandwidth multiplexing networks such as CAN.

The LIN specification package includes not only the specification of the transmission protocol [7] (physical layer and DLL) for master–slave communications but also the specification of a diagnostic protocol on top of the DLL. A LIN cluster consists of one “master” node and several “slave” nodes connected to a common bus. For achieving a low-cost implementation, the physical layer is defined as a single wire with a data rate limited to 20 kb/s due to EMI limitations. The master node decides when and which frame shall be transmitted according to the schedule table. The schedule table is a key element in LIN; it contains the list of frames that are to be sent and their associated frame slots, thus ensuring determinism in the transmission order. At the moment a frame is scheduled for transmission, the master sends a header (a kind of transmission request or command frame) inviting a slave node to send its data in response. Any node interested can read a data frame transmitted on the bus. As in CAN, each message has to be identified: 64 distinct message identifiers are available. Fig. 6 depicts the LIN frame format and the period, termed a “frame slot,” during which a frame is transmitted.

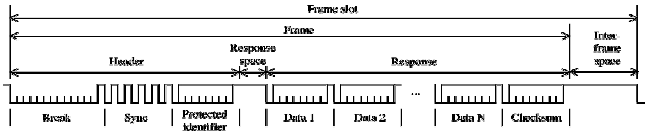

Fig. 6. Format of LIN frame

The header of the frame that contains an identifier is broadcast by the master node, and the slave node that possesses the identifier inserts the data in the response field. The break” symbol is used to signal the beginning of a frame. It contains at least 13 dominant bits (logical value zero) followed by one recessive bit (logical value one) as a break delimiter. The rest of the frame is made of byte fields delimited by one start bit (value zero) and one stop bit (value one), thus resulting in a 10-b stream per byte. The “sync” byte has a fixed value (which corresponds to a bit stream of alternatively zero and one); it allows slave nodes to detect the beginning of a new frame and be synchronized at the start of the identifier field. The so-called protected identifier is composed of two subfields: the first 6 b are used to encode the identifier and the last 2 b the identifier parity. The data field can contain up to 8 B of data. A checksum is calculated over the protected identifier and the data field. Parity bits and checksum enable the receiver of a frame to detect bits that have been inverted during transmission.

LIN defines five different frame types: unconditional, event-triggered, sporadic, diagnostic, and user defined. Frames of the latter type are assigned a specific identifier value and are intended to be used in an application-specific way that is not described in the specification. The first three types of frames are used to convey signals. Unconditional frames are the usual type of frames used in the master–slave dialog and are always sent in their frame slots. Sporadic frames are frames sent by the master, only if at least one signal composing the frame has been updated. A slave will only answer the master if the signals it produces have been updated, thus resulting in bandwidth savings if updates do not take place very often. If more than one slave answers, a collision will occur. The master resolves the collision by requesting all signals in the list one by one. It is also worth noting that LIN offers services to send nodes into a sleep mode and to wake them up, which is convenient, since optimizing energy consumption,

especially when the engine is not running, is a real matter of concern in the automotive context.

2) The TTP/A Network

TTP/A pursues the same aims and shares the main design principles as LIN, and it offers, at the communication controller level, some similar functionalities—in particular, in the areas of plug-and-play capabilities and online diagnostics services. TTP/A implements the classic master–slave dialogue, termed master–slave round, where the slave answers the master's request with a data frame having a fixed length data payload of 4B. The "multipartner" rounds enable several slaves to send up to an overall amount of 62 B of data after a single command frame. A "broadcast round" is a special master–slave round in which the slaves do not send data; it is, for instance, used to implement sleep/wake-up services. The data rate on a single wire transmission support is, as for LIN, equal to 20 kb/s, but other transmission supports enabling higher data rates are possible. To the best of our knowledge, TTP/A is not currently in use in production cars.

E. Multimedia Networks

Many protocols have been adapted or specifically conceived for transmitting the large amount of data needed by emerging multimedia applications in automotive systems. Two major contenders in this category are MOST and IDB-1394.

1) MOST Network

MOST [10] is a multimedia network development of which was initiated in 1998 by the MOST Cooperation (a consortium of carmakers and component suppliers). MOST provides point-to-point audio and video data transfer with a data rate of 24.8 Mb/s. This support end-user applications like radios, global positioning system (GPS) navigation, video displays, and entertainment systems. The MOST's physical layer is a plastic optical fiber (POF) transmission support which provides a much better resilience to EMI and higher transmission rates than classical copper wires. Current production cars from BMW and DaimlerChrysler employ a MOST network.

2) The IDB-1394 Network

IDB-1394 is an automotive version of IEEE 1394 for in-vehicle multimedia and telematics applications. The system architecture of IDB-1394 permits existing IEEE 1394 consumer electronics devices to interoperate with embedded automotive grade devices. IDB-1394 supports a data rate of 100 Mb/s over a twisted pair or POF, with a maximum number of embedded devices which are limited to 63 nodes. From the point of view of transmission rate and interoperability with existing IEEE 1394 consumer electronic devices, IDB-1394 is a serious competitor for MOST technology.

IV. CONCLUSION

The use of communication protocols in automotive industry has helped in design of automobiles with high comfort zone. In-vehicle networking not only supplies the communication between the ECUs, but also increase the performance and safety of vehicle with a great luxury. Standard protocols allow modules from many suppliers to easy link together forming a type of open architecture. As compared to all the protocols CAN is widely used in current vehicle networks. Although multiple vehicle network architecture and the choice of particular protocol are motivated by both business and technical reasons. With the development of automobile industry, the current vehicle network communication protocol should be modified, or new protocols may be developed to adopt the new requirements.

REFERENCES

[1] A. Albert, "Comparison of event-triggered and time-triggered concepts with regards to distributed control systems," presented at the Embedded World Conf. 2004, Nürnberg, Germany, 2004.

[2] C. Wilwert, N. Navet, Y.-Q. Song, and F. Simonot-Lion, "Design of automotive X-by-Wire systems," in The Industrial Communication Technology Handbook, R. Zurawski, Ed. Boca Raton, FL: CRC, 2004.

[3] M. Ayoubi, T. Demmeler, H. Leffler, and P. Köhn, "X-by-Wire functionality, performance and infrastructure," presented at the Convergence Conf. 2004, Detroit, MI.

[4] J. Rushby, "A Comparison of Bus Architecture for Safety-Critical Embedded Systems," NASA/CR, Tech. Rep. NASA/CR-2003- 212161, Mar. 2003.

[5] "J2056/2 survey of known protocols," in SAE Handbook. Warrendale, PA: Soc. Automotive Eng. (SAE), 1994, vol. 2.

[6] Intel Corp.. (2004) Introduction to in-vehicle networking. [Online]. Available: http://support.intel.com/design/auto/autolxbk.htm

[7] LIN Consortium. (2003, Sep.) LIN Specification Package, Revision 2.0. [Online]. Available: http://www.lin~subbus.org/

[8] A. Rajnák, The Industrial Communication Technology Handbook, R. Zurawski, Ed. Boca Raton, FL: CRC, 2005.

[9] Class B Data Communications Network Interface—SAE J1850 Standard—Rev. Nov. '96.

[10] MOST Cooperation. (2004, Aug.) MOST Specification Revision 2.3. [Online]. Available: http://www.mostnet.de

[11] TTTech Computertechnik GmbH. (2003, Nov.) Time-Triggered Protocol TTP/C, High-Level Specification Document, Protocol Version1.1. [Online]. Available: http://www.tttech.com

[12] N. Navet and Y.-Q. Song, "Validation of real-time in-vehicle applications," Comput. Ind., vol. 46, no. 2, pp. 107–122, Nov. 2001.

[13] CAN in Automation. (2005) Challenges in automotive applications. [Online]. Available: http://www.can-cia.org/applications/passengercars/challenge.html

[14] L.-B. Fredriksson, "CAN for critical embedded automotive networks," IEEE Micro, vol. 22, no. 4, pp. 28–35, July–Aug. 2002.

[15] M. Waern, "Evaluation of protocols for automotive systems," M.S. thesis, KTH Machine Design, Stockholm, Sweden, 2003.

[16] H. Kopetz, Real-Time Systems: Design Principles for Distributed Embedded Applications. Norwell, MA: Kluwer, 1997.

[17] H.Kopetz, R. Nossal, R. Hexel, A. Krüger, D. Millinger, R. Pallierer, C. Temple, and M. Krug, "Mode handling in the time-triggered architecture," presented at the IFAC-DCCS '97, Seoul, Korea, 1997.

[18] Robert Bosch GmbH. (2004) Time Triggered Communication on CAN. [Online]. Available: http://www.can.bosch.com/content/ TT_CAN.html

[19] TTA Group. (2004) TTP—Frequently asked questions. [Online]. Available: http://www.ttagroup.org/technology/faq.htm

[20] "J2056/1 class C application requirements classifications," in SAE Handbook. Warrendale, PA: Soc. Automotive Eng. (SAE), 1994.

[21] H. Kopetz et al., Specification of the TTP/A Protocol. Vienna, Austria: Univ. Technol. Vienna, 2002.