

The Impact of Network Topologies on the Performance of the In-Vehicle Network

Rajeshwari Hegde, *Member, IACSIT*, Siddarth Kumar, and K. S. Gurumurthy

Abstract—Today's vehicles contain hundreds of circuits, sensors and around 80-120 Electronic Control Units (ECUs). The communication is needed among many circuits and functions of a vehicle. In earlier vehicle systems, this type of communication was handled via a dedicated wire through point-to-point connections. If all possible combinations of switches, sensors, ECUs and other electronic devices in fully featured vehicles are accumulated, the resulting number of connections and dedicated wiring is enormous. Hence networking of these components is necessary to reduce the complexity of electronics inside a vehicle. In-vehicle networking provides a more efficient method for today's complex in-vehicle communications. This paper focuses on the comparison of the performance of ring and star network topologies on the basis of bus load.

Index Terms—CANoe, CAPL, ECU, IVN.

I. INTRODUCTION

Today's high end cars contain more than ten distributed audio and video ECUs such as visual sensors driver assistance cameras, DVD player, and audio sources such as FM and HD-radio systems. These in-vehicle devices are currently interconnected by different automotive specific network technologies such as Media Oriented System Transport (MOST), Controller Area Network (CAN), and Local Interconnect Network (LIN) [1] that provide limited transmission capacities. Point-to-point links realized by analogue Color Video Blanking Signal (CVBS) cables and Low-Voltage Differential Signaling (LVDS) wires are additionally used to transmit real-time video streams from driver assistance camera systems. The application of different network technologies and point-to-point links leads to an inflexible network architecture and a complex cable harness in the car which is expensive and requires a high validation and management effort. Due to the growing demand for new applications in the driver assistance and multimedia fields, the in-vehicle network will become even more complex and costly in the near future. Thus, traditional automotive network technologies are no longer suitable [1].

Network systems in vehicles represent very high complex systems. Hence, topologies typically become very complex and the layout criticality is a major topic to be considered. Analysis of almost one hundred different topologies of vehicle manufactures worldwide led to the conclusion that less than 50 % of these layouts had been non-critical, if

tolerances of all the parameters involved had been considered in their realistic worst-case scenario. Current vehicle network systems consist of various communication protocol networks such as CAN high speed, CAN low speed, LIN, FlexRay, MOST and others. The major challenge that network developers, dealing with the physical layer implementation, are facing is related to the signal integrity of the communication system. Meaning even if the logical set up and evaluation of the network is fine, the physics can make enormous problems and destroy the complete communication. Since each protocol has its own specification with respect to the physical layer implementation, all of them have their individual issues that the developer must take care of when creating the network. [2]. The paper is organized as follows. Section II gives an insight into the In-Vehicle Network. Section III explains the vehicle communication system. Section IV explains the different types of network topologies. Section V explains the CANoe tool. Section VI explains the CAPL language. Section VII explains the implementation. The paper is concluded in section VIII.

II. IN-VEHICLE NETWORK

As automakers are incorporating more and more advanced features into vehicles, there is a growing need for enhanced processing power. S. Channon and P. Miller [3] estimate that the number of microprocessors per vehicle will increase exponentially and by the end of year 2010, the number of microprocessors in any high end vehicle will be 250 [4]. As today's vehicles contain hundreds of circuits, sensors and around 80-120 ECUs, communication is needed among many circuits and functions of a vehicle. In earlier vehicle systems this type of communication is handled via a dedicated wire through point-to-point connections. If all possible combinations of switches, sensors, ECUs and other electronic devices in fully featured vehicles are accumulated, the resulting number of connections and dedicated wiring is enormous. Networking provides a more efficient method for today's complex in-vehicle communications. In-vehicle networking, also known as multiplexing is a method for transferring data among distributed electronic modules via a serial data bus. Just as LANs connect computers, control networks connect a vehicle's electronic equipments. These networks facilitate the sharing of information and resources among the distributed applications [5]. Without serial networking, inter-module communications requires dedicated, point-to-point wiring resulting in bulky, expensive, complex and difficult to install wiring harnesses. Added wiring increased vehicle weight, weakened performance, and made adherence to reliability standards difficult. For an

Manuscript received September 5, 2012; revised October 18, 2012.

Rajeshwari Hegde is with the Dept of Telecommunication Engineering, BMS College of Engineering, Bangalore, India. (email: Rajeshwari.hegde@gmail.com)

average well-tuned vehicle, every extra 50 kilograms of wiring—or extra 100 watts of power—increases fuel consumption by 0.2 liters for each 100 kilometers traveled. Also, complex wiring harnesses took up large amounts of vehicle volume, limiting expanded functionality [5].

As a result, the wiring harness became the single most expensive and complicated component in vehicle electrical systems [6]. Today’s control and communications networks, based on serial protocols, counter the problems of large amounts of discrete wiring. For example, in a 1998 press release, Motorola reported that replacing wiring harnesses with LANs in the four doors of a BMW reduced the weight by 15 kilograms while enhancing functionality. Applying a serial data bus reduces the number of wires by combining the signals on a single wire through time division multiplexing. Information is sent to individual control modules that control each function, such as anti-lock braking system, turn signals and dashboard display. As the electronic and electrical content of today’s vehicles continue to increase the need for networking is even more evident. For example, some high-end luxury would need more than three miles and nearly 200 pounds of wiring if point-to-point connection is used. The resulting number of connections creates a reliability nightmare. Fig. 1 shows the point to point connection between ECUs.

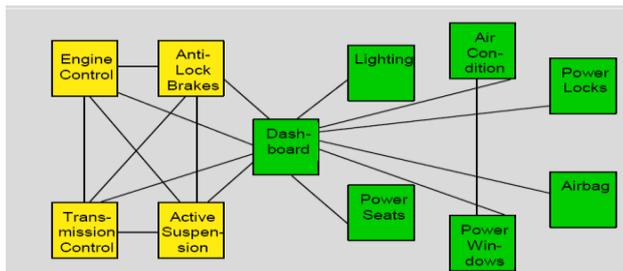


Fig. 1. Interconnection of ECUs using point-to-point connection

In-vehicle networking provides many system-level benefits: A decreased number of dedicated wires are required for each function and this reduces the size of the wiring harness. System cost, weight, reliability, serviceability and installation are also improved. Common sensor data such as vehicle speed, engine temperature etc is available on the network, so data can be shared, thus eliminating the need for redundant sensors.

Networking allows greater vehicle content flexibility as functions can be added through software changes. Existing systems require an additional module or additional I/O pin for each function added. Car manufacturers are discovering new features that are enabled by networking. For example the Lincoln Continental’s memory profile system stores each driver’s preference for ride firmness, seat positions, steering assist effort, mirror positions and even radio station presets. Fig. 2 shows the example of in-vehicle network in an automotive system using various buses [7].

Unlike a point-to-point connection a bus is a communication system that can logically connect several peripherals, i.e. bus controllers over the same set of wires. The consequential potential savings of cost and weight encourage the increasing application of bus systems as communication systems within the automotive area.

Moreover busses are easy to implement and to extend, and the failure of one node should not affect others. However, since in a bus system all nodes share the same communication line, they need schemes for collision handling or collision avoidance, or require a bus master which controls access to the shared bus resource. Furthermore, bus systems have a limited cable length and a limited number of nodes. The performance of a bus communication degrades the more nodes are connected, whereas a cable break can disable the entire vehicular bus network [8].

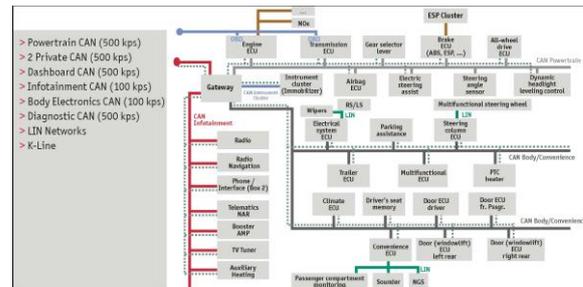


Fig. 2. Current network architecture at Volkswagen

From the point of view of the electric/electronics, the complete vehicle system can be divided into four domains or functional areas, Driver train, Chassis, Interior and Telematics. In the domains “driver train” and “chassis”, it is primarily real-time applications that are in the foreground. In the domain “interior”, the focus of networking is on multiplex aspects. In the domain “telematics”, it is primarily multimedia and infotainment applications that are networked.

III. VEHICLE COMMUNICATION SYSTEM

A wide variety of vehicle communication systems exist in today’s automotive system. [9]. Vehicle functions are divided into systems and sub-systems to provide for passenger entertainment, comfort, and safety, as well as to improve vehicle performance and enhance powertrain control. These systems must communicate with one another over a complex heterogeneous in-vehicle network (IVN). Each network typically contains multiple communication protocols including the industry standard Controller Area Network (CAN), LIN, FlexRay and MOST [10]. The popular Network Communication Protocols used are as follows.

A. Local Interconnect Network (LIN)

A low-speed master-slave time triggered protocol meant to connect on-off type loads to higher speed networks. Typical loads include door locks, sun roofs, rain sensors, and powered mirrors. A LIN network is used as a low cost alternative if the full functionality of the CAN protocol is not required.

Fig. 1 illustrates a typical example of LIN reducing the wire count into a door from dozens of wires to a minimum of three (LIN, power, ground) [11].

B. Controller Area Network (CAN)

An event driven communication protocol used in

applications such as engine management and body electronics. The maximum specified data rate is 1 Mbps, though the practical maximum is 500 Kbps. High-speed CAN is suitable for critical loads such as anti-lock braking systems and cruise control. Low-speed CAN is fault-tolerant and used for loads such as power seats and motorized windows.

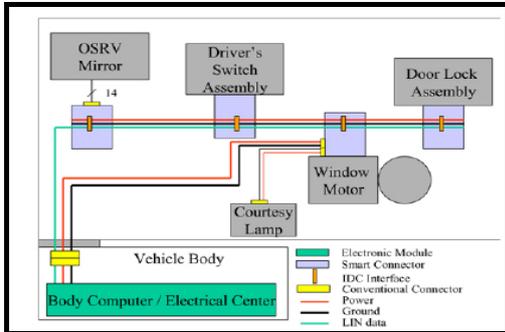


Fig. 3. Multiplexed Car Door Example

C. FlexRay

A fault-tolerant high-speed communication protocol targeted toward safety-related applications. The protocol can be operated in single or dual channel mode, where each channel has a maximum data rate of 10 Mbps. Using a dual-channel configuration, a FlexRay network can operate at speeds 20x faster than the maximum CAN bus data rate specification. Along with enabling safety-related applications, a FlexRay network is well suited as a communication backbone connecting heterogeneous networks together.

D. Media-Oriented Systems Transport (MOST)

The applications of MOST, a fiber-optic network protocol with capacity for high-volume streaming, include automotive multimedia and personal computer networking. More than 50 firms—including Audi, BMW, Daimler-Chrysler, Becker Automotive, and Oasis Silicon Systems—developed the protocol under the MOST Cooperative.

E. Byteflight

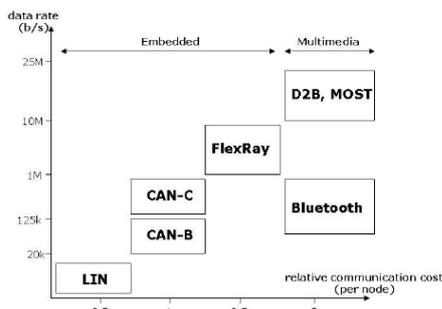


Fig. 4. Comparison of several in-vehicle network protocols with respect to data rate and communication cost [12].

A flexible Time Division Multiple Access (TDMA) protocol for safety-related applications, Byteflight can be used with devices such as air bags and seat-belt tensioners. Because of its flexibility, Byteflight can also be used for body and convenience functions, such as central locking, seat motion control, and power windows. BMW, EL MOS, Infineon, Motorola, and Tyco EC collaborated in its

development. Although not specifically designed for X-by-wire applications, Byteflight is a very high performance network with many of the features necessary for X-by-wire.

Fig. 4 shows the major protocols used in the vehicles.

IV. NETWORK TOPOLOGIES

The request for more new functions has significantly increased the number of ECUs. Up to 22 ECUs is not uncommon in today's high-speed CAN networks. When designing the networking topologies in terms of signal integrity however, several challenging issues should be considered beyond the number of ECUs [13]. A network topology is understood to be the structure consisting of network nodes and connections. This merely shows which nodes are interconnected, but does not depict underlying details such as the length of the connection. Every The different network topologies are explained in the following section.

A. Star Topology

In this topology there is a central hub at which all devices are connected. Each device has its own line. If the central hub fails, the entire communication breaks down. Fig. 5 shows the star topology.

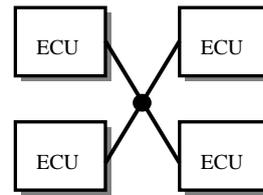


Fig. 5. Interconnection of ECUs using star topology

B. Bus Topology

Devices are connected by short branch lines to a main line. Every communication flows over this main line. If this main line is interrupted, two segments are formed which normally continue functioning. Line topology is also called "Linear topology" or "Bus topology". The CAN bus also uses this topology.

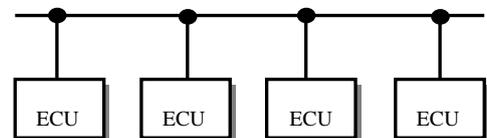


Fig. 6. Interconnection of ECUs using bus topology

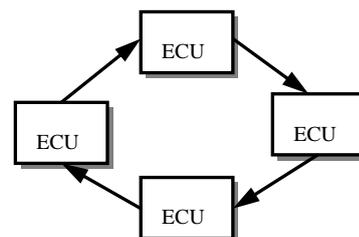


Fig. 7. Interconnection of ECUs using ring topology

C. Ring Topology

The point-to-point connections between devices are

representative of ring topology. All connections are arranged in a closed chain. The communication can be done in only one direction. If a section of line fails, the entire system no longer functions [14].

V. CANOE-THE PREMIER TOOL FOR AUTOMOTIVE NETWORK DEVELOPMENT

CANoe is an all-round tool suite for the development, testing and analysis of networks and ECUs. It allows the user to analyze, simulate, and test an entire CAN network in a user friendly, highly configurable environment. It provides the user with various diagnostic and analysis features. Due to its open architecture, CANoe is able to solve complex tasks and be tailored for special application. Models both graphic and text based as well as evaluation windows are provided for simulating and analyzing the entire distributed networks. Instiutive user control panels can be created for monitoring and controlling tasks, e.g. in the production or assembly context It has a built-in COM interface. COM is a standard defined by Microsoft for the communication between different software components. Different programming and scripting languages can be used to access the COM server functionality of CANoe. All used scripts must be based on Microsoft's window script software components. Scripting languages such as VB script or Jscript which are the parts of Microsoft windows 98 and above packages, can be used as scripting languages. Programming languages such as Microsoft Visual Basic, Microsoft Visual C++ and Borland Delphi, CAPL can be used to create user specific applications. Included with the product is the CANdb++ editor and the Panel editor. CANoe has several interfaces to simulate and test networks of ECUs. [15].

VI. COMMUNICATION ACCESS PROGRAMMING LANGUAGE-CAPL

Based on the C programming language, CAPL, or CAN Access Programming Language, is the programming language used exclusively within the PC-based tool environments of CANoe. The original design intent behind CAPL (which is pronounced "kapple") was to meet the CAN-based distributed embedded system developer's requirements. The creation of CAPL and its programming environment became the implementation to meet these requirements. Using CANoe in combination with CAPL makes it possible to create custom tool applications with user defined behavior. Potential applications are limited only by imagination, available communication hardware limitations (if applicable), and the speed of the PC [16].

VII. IMPLEMENTATION

In order to see the effect of topologies on the performance of the network, a software simulation was done and the results were compared. For instance, a simulation of two such topologies: ring and star was done and the results and observations are given below. The performance attribute compared here is the bus load. The simulation was done

using CANoe software.

The simulated network consisted of four nodes(equivalent to four ECU's) . Each node transmits two messages. In case of ring topology, each node receives only messages from its adjacent node. Whereas in case of star topology, each node receives from every other node. With this setup, the bus load was observed in the data measurement. It was observed that, after the same duration of time , the bus load with the ring topology was higher by about 3.93% than that in the star topology. A similar simulation can be done for different topologies and the effects can be compared. The simulation setup for ring and star topologies are shown in Fig. 8 and 9.

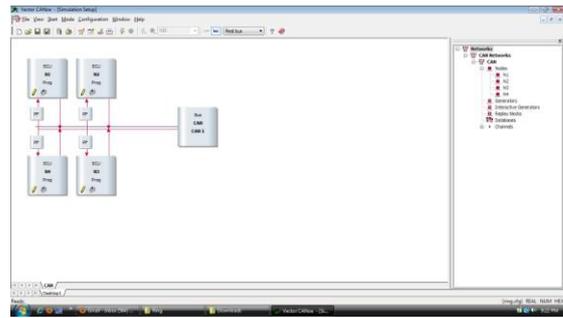


Fig. 8. Implementation of ring topology

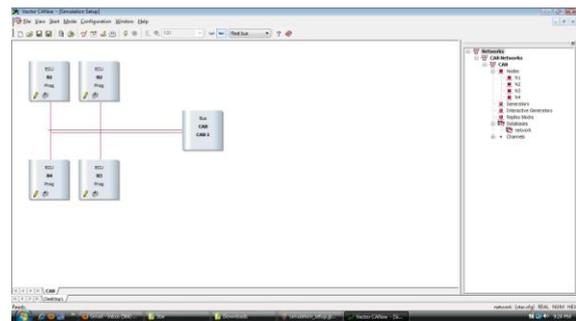


Fig. 9. Implementation of star topology

CANoe allows the simulation of a vehicle network by creating a software equivalent of ECU's which are termed nodes. Multiple nodes can be connected to a central bus. For the node to behave like an ECU , it should participate actively in the network by means of transmitting messages & signals, and in turn respond appropriately to those that it receives. This 'intelligence' is given to the node by CAPL programs. CAPL(Communication Access Programming Language) allows for event based as well as timed execution that makes it suitable for this purpose.

For the simulation here, the demo version of the software has been used. The only limitation was with respect to the number of nodes that could be simulated, which is four. In the full version, a simulation of more than four nodes is possible either by connecting directly to the bus or via a gateway which is a link to another similar network.

VIII. CONCLUSION

In this paper, the network topologies used for in-vehicle communication has been explained. The star and ring topologies were simulated using CANoe software and it was found that the star topology performs better than the ring

topology. The only limitation in the simulation was the number of nodes that could be connected to the network was limited to four. To see the effect of the network topologies on the performance of the in-vehicle network, the real ECUs were connected to the simulated network to measure the load to get a better estimate of the performances of the different topologies. To implement this, ECU were connected to the simulated bus via a CanCase box or CanCard and the ECUs were powered up to start transmitting and receiving the messages to measure the load. The other simulated nodes should be modified to simulate the actual ECUs that are available in the network. This is done by writing the appropriate CAPL code so that the required messages are transmitted and the received messages are suitably acted upon.

REFERENCES

[1] M. Rahmani, K. Tappayuthpijarn, B. Krebs, E. Steinbach, and R. Bogenberger, "Traffic Shaping for Resource-Efficient In-Vehicle Communication," *IEEE Transactions on Industrial Informatics*, vol. 5, no. 4, November 2009.

[2] W. Lawrenz and D. Bollati, "Validation of in-vehicle-protocol network topologies," presented at Second International Conference on Systems, 2007.

[3] S. Channon and P. Miller, "The requirements of future in-vehicle networks and an example implementation," *SAE paper 2004-01-0206*.

[4] P. R. Ramteke and S. M. Mahmud, "An Adaptive Data-Reduction Protocol for the Future In-Vehicle Networks," *SAE World Congress*, 2005.

[5] G. Leen and D. Heffernan, "Expanding Automotive Electronic Systems," in *Proceedings of Computer*, vol. 35, no. 1, pp. 88-93, 2002.

[6] B. Somers, "Investigation of a FlexRay-CAN Gateway in the Implementation of Vehicle Speed Control," M.S thesis, Dept. of Engineering Technology, Waterford Institute of Technology, June 2009

[7] *Current Challenges in Automotive Networking*, Vector, Technical Article.

[8] M. Wolf, A. Weimerskirch, and C. Paar. Secure In-Vehicle Communication. [Online]. Available: <http://www.buchhandel.de/WebApi1/GetMmo.asp?MmoId=5448260&mmoType=PDF>

[9] H. Heit, "The Role of CAN in ECU Communication," *CAN Newsletter Automotive Networks*, 2006.

[10] Mechatronics. [Online]. Available: http://www.synopsys.com/tools/sld/mechatronics/Saber/CapsuleModule/saber_invehicle_network_ds.pdf

[11] Network Topology. [Online]. Available: <http://delphi.com/pdf/techpapers/2010-01-0687.pdf>

[12] U. Keskin, *In-Vehicle Communication Networks: A Literature Survey*, Technische Universiteit Eindhoven (TU/e), The Netherlands, 2009.

[13] The Virtual Vehicle:Part1-In-Vehicle networking simulation and analysis. [Online]. Available: <http://www.automotivedesignline.com>

[14] Network. [Online]. Available: <http://www.lernsystem.original-marken-partner.de/Network-topology.436.0.html?&L=1>

[15] R. Hegde and K S Gurumurthy, "Programming Environment for CANoe and its Implementation for Automotive Development Application," *IEEE International Advanced Computing Conference*, pp. 2886-2889, March 2009.

[16] *Programming with CAPL*, Vector CANtech, Inc, USA, 2004



Rajeshwari Hegde completed her Bachelor of Engineering in Electronics and Communication Engineering from National Institute of Engineering, Mysore and Master of Engineering in Electronics from BMS College of Engineering, Bangalore. She has completed her research in the field of automotive embedded system at Bangalore University, India, under the guidance of Dr K S Gurumurthy. Her major field of study includes embedded system, signal processing and wireless communication. She is currently working as Associate Professor in the Department of Telecommunication Engineering, BMS College of Engineering, Bangalore, India. She has published 47 research papers in international conferences, national conferences and reputed journals.



Siddharth Kumar completed his Bachelor of Engineering in Telecommunication Engineering from BMS College of Engineering, Bangalore in 2008. He worked as a software Engineering in Robert Bosch Business and Engineering Solutions, Bangalore, India for two years. Currently he is pursuing his Masters in Hamburg University of Technology, Hamburg, Germany.



K. S. Gurumurthy completed his Bachelor of Engineering in Electronics and Communication, from Mysore University and Master of Engineering in Solid State Electronics from U.O.R., Roorkee (now IIT, Roorkee), and PhD from Indian Institute of Science, Bangalore, India. His major field of study includes VLSI and microelectronics.

He has two years of industrial experience and more than 30 years of research and academic experience. He is currently working as Professor and Head, Department of Electronics and Communication Engineering, University Visvesvaraiiah College of Engineering, Bangalore, India. He has given more than 50 guest lectures on different topics of VLSI DESIGN at various colleges, WIPRO, CRL and at IETE. He has published more than 85 research papers in various international journals and conferences. His research interests are VLSI Design, Embedded System and Communication. Dr Gurumurthy is the professional member of IEEE and ISTE. He is the recipient of KHOSLA AWARD for the BEST Technical Paper presented at UOR, Roorkee, India.