FlexRay and MOST Automotive Protocols

[1]Rudrappa B. Gujanatti, [2]Bipin B.Kagganagi, [3]Bhagyashree Havaldar

[1]Assistant Professor, [2,3]Student
Electronics And Communication Department
KLE Dr.M S Sheshgiri College Of Engineering And Technology, Belagavi, INDIA

***Abstract*— The FlexRay automotive Protocol was developed by the FlexRay consortium which was started when BMW and DaimlerChrysler worked together to bring a new network scheme. In recent years modern automobiles integrates numerous number of Electronic components are increased rapidly. These automotive embedded systems have great demand for dependability, on designing the FlexRay protocol. FlexRay (FR) is mainly for scalable, flexible, high speed deterministic, error tolerant communication in order to meet growing safety related challenges in the automobile industry and where it will replace or support existing networking schemes such as Fieldbus technologies and CAN. Neither the High-Speed CAN (1 Mbps bandwidth) nor TTP protocol (10 Mbps bandwidth), can be compared to the 1 Gbps bandwidth that is typical for widely used computer networks. The throughput of communication network required for the solutions in the scope of vehicle control ans diagnostic are limited(<10 Mbps). They are unsuitable for smooth multimedia transmission from several sources.The article contains consistent summary of FlexRay and Media Oriented Systems Transport (MOST) communication protocol features.**

***Index Terms*— ECU, TDMA, communication cycle, communication controller (CC) ,Static segment, Dynamic segment**

I. INTRODUCTION

Safety-critical driver assistance functions with electronic interfaces to the chassis introduce extremely stringent requirements on the safety , reliability and real-time capability of the communication system. What is needed is a communication system with the property of compos ability, whose core property is to guarantee fault tolerant and deterministic data communication independent of bus load

Controller Area Network (CAN), the communication technology that has become established in the automotive industries, cannot fulfill this challenging set of requirements, since CAN is based on an event-driven communication approach, which means that every bus node of a communication system must be able to access the common communication medium at any time. The use of technologies for resolving collisions leads to a communication flow that cannot be determined until runtime. Event-driven communication systems enable quick reaction to asynchronous processes, but they are non-deterministic. Because there is no strict schedule in an event-driven communication system, adding and removing bus nodes affects the communication flow. Strictly speaking, such changes make it necessary to comprehensively revalidate the entire system. Event-driven communication systems do not exhibit the property of composability.

CAN automotive communication protocol cannot fulfill the high data requirements for fault tolerance due to its lack of redundant structures and it can deliver a maximum only 500kbit/s data rate in series production. Therefore some automotive OEMs were already experimenting with fault tolerant, time-triggered communication technologies, which allowed very high data rates, in the 1990s.

Nonetheless, the studies and experience gained at automotive OEMs did not yield any communication technologies that could meet all of the requirements for production implementation of future, safety-critical systems in motor vehicles. That is why DaimlerChrysler and BMW in 1999 agreed to work together to advance the specification and development of a future scope, time-triggering, uniform, and fault tolerant communication in automotive field. This cooperative effort resulted in requirements specification for FlexRay.
In the year 2001, the first multimedia installation based on MOST bus and protocol was introduced. During the same year MOST bus was applied in the next ten vehicle models. MOST cooperation consortium could report MOST introduction into 140 vehicle models including new modules i.e MERCEDES Class S and AUDI A3. MOST bus and protocol have been present in popular medium segment vehicles e.g. Opel Insignia and Volkswagen Golf as well as the models: Rolls Royce Ghost, Wraith and Phantom.
Linear bus topology is based on the functioning of majority of wire communication buses in motor vehicles. Since MOST bus is based on ring topology, it is a unique solution(Fig. 1.1).Only after transceivers replacement the communication via cable connections is possible. One of devices (master) contains the complete database for devices in the ring (Fig. 1b)

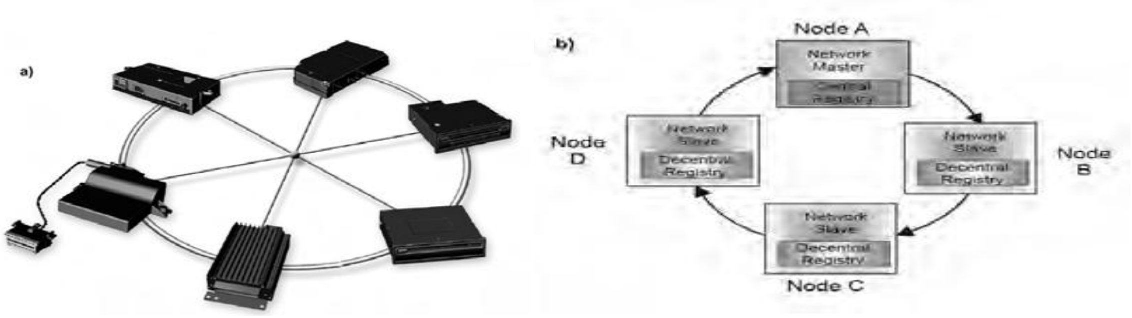


FIG. 1.1 THE COMMUNICATION BUS IN RING TOPOLOGY : A)VIEW OF REAL DEVICES, B) COMMUNICATION MECHANISM DIAGRAM

MOST bus operation is typical for ring topology. Information and commands source gathered is used by the data block received from preceding node. The block received is regenerated and forwarded. Switched off devices transmit optical signal without its inspection. When the block is received by its sender, the data transfer is finished. There are some special nodes contained in the ring which are responsible for the ring management on the basis of user activities which is used for the ring synchronization (Fig.1.1(b)). MOST bus and protocol are dedicated to multi116 media networks which are sometimes called Infotainment (Information+Entertainment) networks. For data stream in such networks high throughput levels are required. In spite of MOST150 standard functioning past many years, this fact has been not mentioned in many publications. Many times the graphical presentations inform about the throughput of about 25 Mbps (Fig.) which is underestimated by 3 times. The throughput of 150 Mbps will be probably exceeded soon. The manufacturers of POF(Plastic Optical Fibers) indicate the throughput's of 500 Mbps along the section of 0.02km or 170 Mbps along the section of 0.115 km. The transceiving equipment is prepared with throughput of 5 Gbps for operation. The throughput of 5 Gbps is sufficient to use MOST as an element in the network supporting images received from the games network or from security camera.
The communications based on Function Blocks is the usual feature of all MOST versions. As it is a "new" communication method, it is possible to find references for its individual elements in layer type OSI communication model. Application layer commands are the key Media Oriented System Transport (MOST) Function Blocks as the standardized commands used to control the behavior of devices installed in the ring and enabling the data transfer. On each device installed in the ring and conforming with MOST specification will support the Function Block commands.Built-in support for Function Block. The functional addresses are assigned to the devices after their overview into the ring. By means of told addresses the devices performing the role of controllers can read and declare the operation parameters which is used for the devices performing the role of saves.
The devices and accessible software can be divided into three groups: Human Machine Interface-the software making it possible for the user to communicate with Media Oriented System Transport (MOST) bus devices, also the program user's interface; Controllers – the devices functioning as the controllers for other devices,they manage the information about the Function Block commands for controlled systems used in the MOST; Slaves – the actuators which doesn't have any information about the system structure but respond to the Function Block commands associated with them. Because to their not active role in the system, the slave devices can be added to and removed from the system without changing the configuration of the whole bus. The communication method based on Function Blocks is common for all versions of MOST protocol: 25, 50, 150.

Automobiles have changed from having a simple radio with perhaps a cassette or CD player to having a variety of sophisticated information systems and entertainment that need to communicate and interact with each other and with a human user.
Why do these devices need networking? Shared displays for radio, navigation, rear-view camera. Traffic information transmitted to the radio (TMC/RDS) is used for efficient navigation Why do MOST devices need a new networking protocol? Safety-critical applications require reliable, real-time networking. Infotainment (Information/Entertainment) applications require fast, short-delay networking with continuous bandwidth availability. Automotive media-oriented networking

Media Oriented Systems Transport (MOST).It was in the year 1998 MOST cooperation (BMW, Daimler, Audi, Oasis, Harman-Becker) was found. More than 70 suppliers used by 16 car makers. Application in BMW series 7, Audi A6/A8/Q7.Transmission of streaming data (e.g. entertainment system) and internet information (packet-based data) .Based on D2B (developed by Daimler).Standardization of all 7 ISO/OSI protocol layers.

No. of layer	ISO/OSI ref model	MOST protocol specification			
7	*Application*	Function Block	Function Block	Function Block	Stream Service
		Network Service Layer 2 (Application Socket)			
6	*Presentation*				
5	*Session*	Network Service Layer 1 (Basic Level)			
4	*Transport*				
3	*Network*				
2	*Data Link*	Low-level System Services			
1	*Physical*	Physical Layer			

FIG.1.2 MOST IN ISO-OSI REFERENCE MODEL

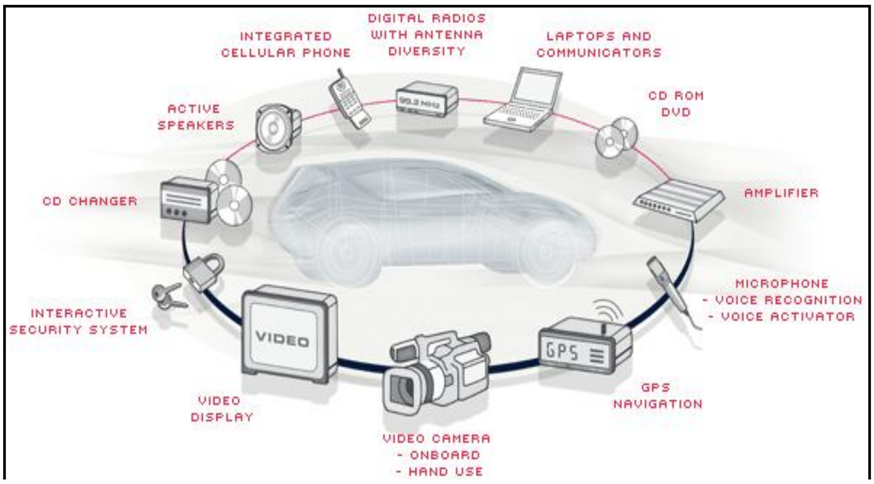


FIG.1.3 MOST DEVICES

II. FLEXRAY NODE ARCHITECTURE

Each node has a communication controller, a host cpu, a power supply unit and two bus drivers for each channel. Fig shows node architecture of FlexRay.

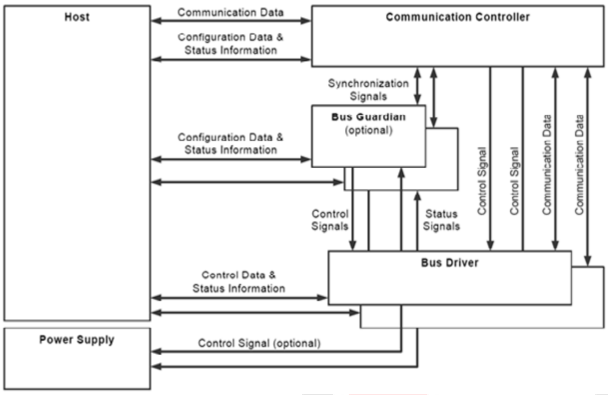


FIG. 2- ALL LOGICAL INTERFACES

The host handles the applications of the system where as the FlexRay protocol is handled by the communications controller(CC). The bus driver is used to write and read data to the physical medium over which the transmission of data take place. In sleep mode it has the ability to start the wakeup procedure when wakeup symbol is detected. The communications controller will mainly handle the framing of data and the checks for corrupted data that received data is uncorrupted before sending it to the host. The host and communications controller both exchanges the information such as control information from CC and payload data from the host cpu, while the communication controller relays status of information and received data . The host interface to the bus driver allows bus driver to change the operation as well as read status and error flags.

III. FLEXRAY TOPOLOGY AND LAYOUT

One of the things that distinguishes FlexRay, LIN and CAN from more traditional networks such as network layout or Ethernet is its topology. FlexRay supports multi-drop passive connections and also active star connections for complex networks. Depending on vehicle's layout and level of using FlexRay , designer can choose right topology based upon his requirements to reduce cost and compact design.

Multi – drop Bus

FIG 3.1

FlexRay protocol is commonly used in a simple **multi-drop bus** topology that features a single network cable run it connects more number of ECUs together. This is the same topology used by CAN (Controller Area Network) and LIN and is familiar to OEMs, making this as popular topology in first-generation FlexRay vehicles. Each ECU can "branch" up to very short distance from the core "trunk" of the bus. The ends of the network have termination resistors which is installed that eliminate problems with signal reflections. Because FlexRay operates at high frequencies and it has high data rate of 10 Mbit/s where as CAN has 1 Mbit, FlexRay designers much take care to correctly terminate and lay out networks to avoid signal integrity problems.

Star Network

FIG 1.2

The FlexRay standard supports "Star" configurations which consist of individual links that connected to central active node. This node acts similar to a hub found in PC ethernet networks. The active star formation makes it possible to run FlexRay networks over longer distances or to segment the network in such a way that, this star network makes it more reliable should a portion of the network fail. If one of the branches of the star is shorted or cut , the other legs will continue to functioning.

Hybrid Network

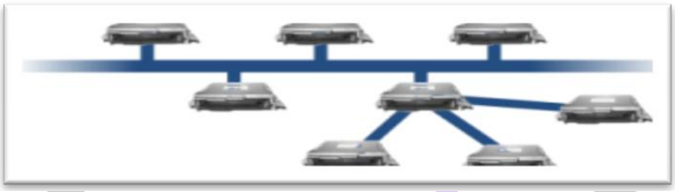


FIG. 3.3

The star and bus topologies can be combined to form a **hybrid** topology. In future FlexRay networks will likely consist of hybrid networks which takes the benifit of the ease-of-use and cost advantages of the bus topology while applying reliability and the performance of star networks where needed in a vehicle.

IV. FLEXRAY PRINCIPLE

The FlexRay protocol is a time-triggered protocol that provides options for predictable time frame (down to the microsecond) as well as CAN-like dynamic event-driven data to handle a different types of frames. FlexRay accomplishes this hybrid of core dynamic frames and static frames with a pre-set **communication cycle** that provides a pre-defined space for dynamic and static data. This space is configured by the network designer. While CAN nodes only needed to know the correct baud rate to communicate, nodes on a FlexRay network should know how all the pieces of the network are configured in order to communicate.

There are a variety of schemes used to prevent contention on a bus. CAN, for example, used bus arbitration scheme where nodes will yield to other nodes if they see a message with higher priority that sent through bus. While easy to expand and flexible , this technique does not allow for high data rates and it would not give guarantee on timely delivery of data. FlexRay protocol manages multiple nodes with a **Time Division Multiple Access** (TDMA) scheme. Each node of the FlexRay is synchronized to the same clock, and each node waits for its turn to write on the bus. Because the timing is consistent in a TDMA (Time Division Multiple Access) scheme, FlexRay has ability to give guarantee on consistency of data deliver to nodes or **determinism** of the network.
Embedded networks differ from PCB designed networks in that they have a closed formation and we can't change once they are assembled in the production product, which eliminates the need for additional mechanisms to automatically configure and discover devices at run-time, much like a PC does when joining a wireless or new wired network. By designing network configurations ahead of time, network designers will increase reliability of the network and also can save significant cost. For a TDMA network such as FlexRay to work correctly, all nodes must be synchronized correctly. The FlexRay standard is adaptable to many different types of networks and allows network designers to make interchange between network update deterministic data volume, speeds and dynamic data volume among other parameters. FlexRay committee standardized a format for the storage and transfer of these parameters in the engineering process. The Field Bus Exchange Format which is known as **FIBEX** file is an ASAM-defined standard that allows validaters ,network designers, testers and prototypers to easily share network parameters and quickly configure ECUs, hardware-in-the-loop simulation systems, test tools and so on for easy access to the bus.

V. COMMUNICATION CYCLE

The **FlexRay Protocol communication cycle** is the fundamental element of the media-access scheme within FlexRay. The duration of a cycle is fixed when the network is designed by the networker, but is approximately 1-5 ms. There are four main parts to a communication cycle:

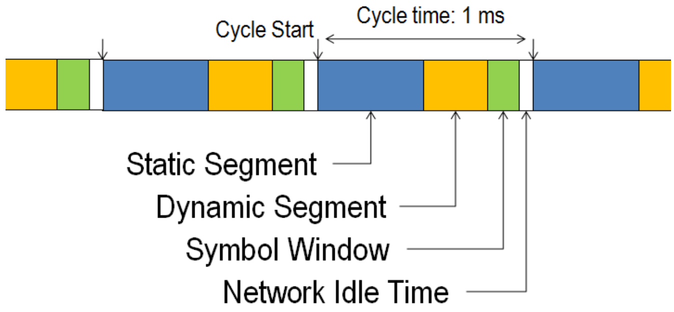


FIG. 2.1

1. **Staticsegment**
   Reserved slots for deterministic data that arrives at a fixed period.
2. **DynamicSegment**
   The dynamic segment behaves similar to CAN and is used for a wider variety of event-based data that does not require determinism.
3. **SymbolWindow**
   Typically used for network maintenance and signaling for starting the network.
4. **Network Idle Time**
   A known "quiet" time used to maintain synchronization between node clocks.

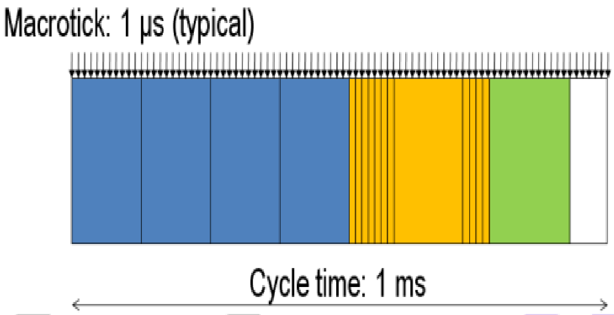


FIG. 5.2- MACROTIC

The smallest practical unit of time on a FlexRay network is a **macrotick**. FlexRay controllers actively synchronize themselves and they adjust their local clocks so that the macrotick occurs at the same point in time on every node across the network. During configuration of particular network, macroticks are often 1 microsecond long.

1 *Static Segment*

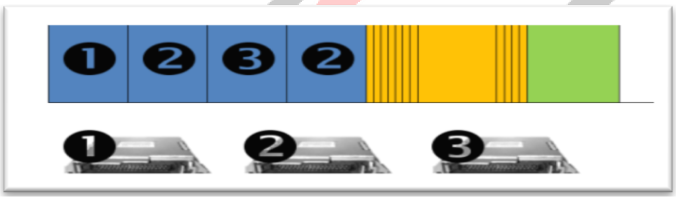


FIG. 5.3- ILLUSTRATION OF A STATIC SEGMENT WITH 3 ECUS TRANSMITTING DATA TO 4 RESERVED SLOTS.

The blue segment represents the static segment and is the space in the cycle is dedicated to scheduling a number of time-triggered frames. The segment is divided into slots, each slot contains a reserved frame of data. When each slot occurs in time, the reserved ECU will try to transmit its data into that slot. Once that time passes, the ECU should wait until the next cycle to transmit its data in that slot. Because the exact point in time is known in the cycle, the data is deterministic and programmers will come to know that exactly how old the data is. This is very useful when calculating control loops that depend on consistently spaced data. **Fig. 7** illustrates a simple network with four static slots being used by three Electronic Component Units. Actual FlexRay networks may contain up to several dozen static slots.

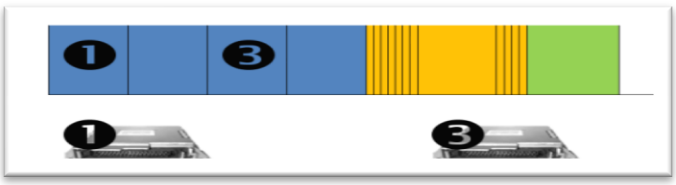


FIG. 5.4- STATIC SLOT WITH ECU 2 IS MISSING.

If an ECU goes offline or decides not to transmit data, its slot remains open.It is not used by any other ECU, as shown in Fig

2 DynamicSegment

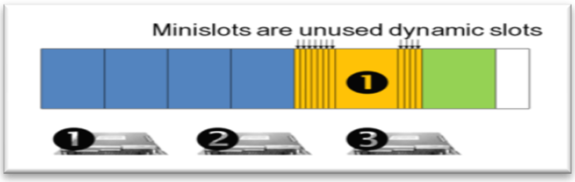


FIG. 5.5- ILLUSTRATION OF FLEXRAY DYNAMIC SLOTS WITH ONE ECU BROADCASTING DATA.

Most embedded networks have a few number of high-speed messages and very large number of lower-speed, less-critical networks. To accommodate a wide variety of data without slowing down the FlexRay cycle with an more number of static slots, the dynamic segment allows occasionally transmitted data. The segment has fixed length, so there is a limit of the fixed amount of data that can be placed in the dynamic segment per cycle. To give priority to the data, **minislots** which are pre-assigned to each frame of data that can be transmitted in the dynamic segment. A minislot is typically a **macrotick** (a microsecond) long. The data which has higher priority receives a minislot closer to the beginning of the dynamic frame.
Once a minislot occurs, an ECU will get chance to broadcast its frame. If it doesn't broadcast, it will loose its spot in the dynamic frame and the next minislot occurs. This process will continue until an ECU elects to broadcast data. As the data is broadcast, future minislots should wait until the ECU completes its data broadcast. If the dynamic frame window ends, then the lower-priority minislots should wait until the next cycle for another chance to broadcast.

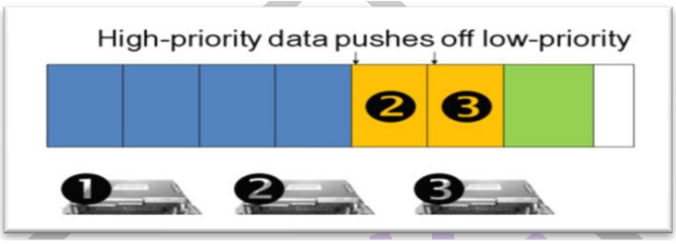


FIG. 5.6- DYNAMIC SLOTS ILLUSTRATION SHOWING ECUS 2 AND ECU3 BROADCASTING IN THEIR MINISLOTS AND LEAVING NO TIME FOR THE LOWER-PRIORITY MINISLOTS

Fig. shows first ECU is broadcasting in its minislot since the first 7 minislots chose not to broadcast. Fig. 6 shows second and third ECUs using the first two minislots, leaving no time for first ECU to broadcast. First ECU should wait for the next cycle to broadcast.
The end result of the dynamic segment is a scheme similar to the bus arbitration scheme used by CAN.

3 Symbol Window

The Symbol window is mainly used for identification and maintenance of special cycles such as cold-start cycles. Most of the high-level applications do not interact with the symbol window

4 Network Idle Time

The network idle time is of a pre-defined, known length by ECUs. The ECUs make use of this idle time to make an adjustment for any drift that may have occurred during the previous cycle.

VI. FRAME FORMAT

The frame of FlexRay divided into 3 sections.

1. Header segment
2. Payload segment
3. Trailer segment

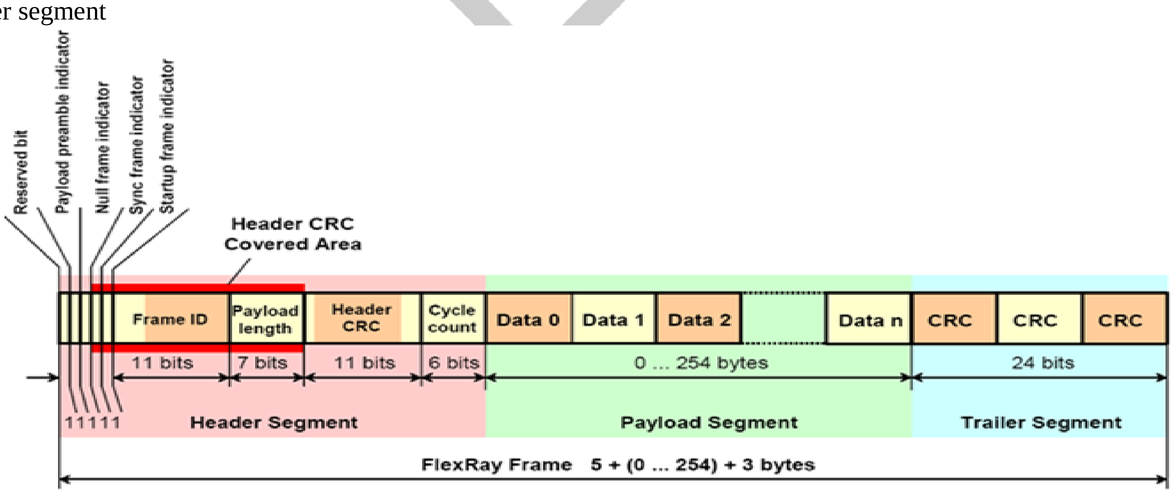


FIG. 6-FRAME FORMAT

Header Segment The FlexRay header segment is 5 bytes long. It consists of 9 parts. Each part has different function.

i. Reserved bit (1bit): It is reserved for future protocol use
ii. Payload preamble indicator (1 bit): It indicates the extension of vector information in the frames payload segment. If the frame is transmitted in the static segment, this position will indicate the presence of a network management vector at the beginning of the payload. If the frame is transmitted in the dynamic segment, this position indicates the presence of a frame ID at the beginning of the payload.
iii. Null frame indicator (1bit): It is used to identify the frame is empty. “0” means the payload segment contained no valid data, “1” means the payload segment contains valid data.
iv. Sync frame indicator (1bit): It is to check the frame is sync frame. If “0” means no synchronization for node; and “1” means all the receiving nodes used for synchronization.
v. Start up indicator (1bit) : It indicates frame is a start up frame. “0” means this frame this frame is not a start-up frame, “1” means this frame is a start-up frame.
vi. Frame ID (11 bits): A frame ID is unique on each channel in communication controller. The frame ID ranges from 1 to 2047. “0” is used for invalid messages.
vii. Payload length (7 bits): This part is used to indicate the size of the payload segment. The value of the payload length position is set to the number of payload bytes divided by 2. Its range is from 0 to 254 bytes.
viii. Header CRC (11 bits) : This part contains a cyclic redundancy check code (CRC), the start-up frame indicates the , the frame ID, and the payload length. The header CRC of transmitted frames is computed on time and provided to the Communication Controller by means of configuration. It is not computed by transmitting CC. The CC calculates the received frame’s header CRC in order to check that the CRC is correct.
ix. Cycle count (6 bits) : This part indicates the value of cycle counter from the transmitting nodes at the time of frame transmissions.

B. Payload Segment

The payload contains the actual data that is transferred by the frame. The length of the FlexRay payload or data frame is up to 127 words (254 bytes), which is over 30 times greater compared to CAN.

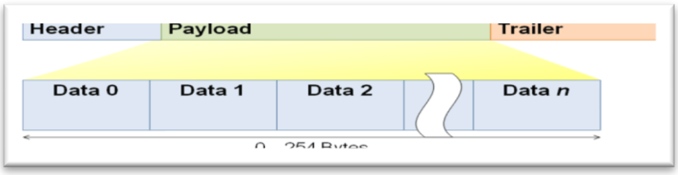


FIG. 6.1

C. Trailer Segment

The FlexRay trailer segment consists of 24-bit cyclic redundancy check code (CRC) for the frame. It is computed with the data in the header segment and the payload segment of the frame.

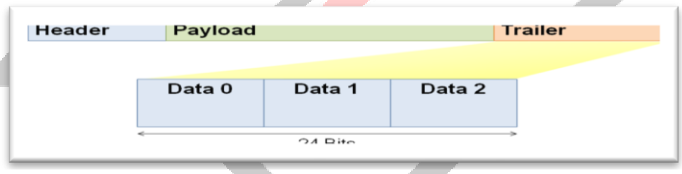


FIG. 6.2- TRAILER OF FLEXRAY

VII. BLOCK DIAGRAM OF MOST PROTOCOL

Function blocks

Functions are sorted in standardized interfaces (= function blocks).Interoperability and interchangeability: e.g. device that requires a GUI interface can be used by every device that requires a GUI. Similar to Profiles of Bluetooth and CAN open. Function blocks provide functions, i.e. Properties: specification of a specific device characteristic (e.g. current frequency of the radio tuner). Different types of methods: trigger an action(e.g. radio station scan).Controller application reads and later adjusts the properties of the slaves registers for particular events(property changes) of the slaves calls messages of the slaves.

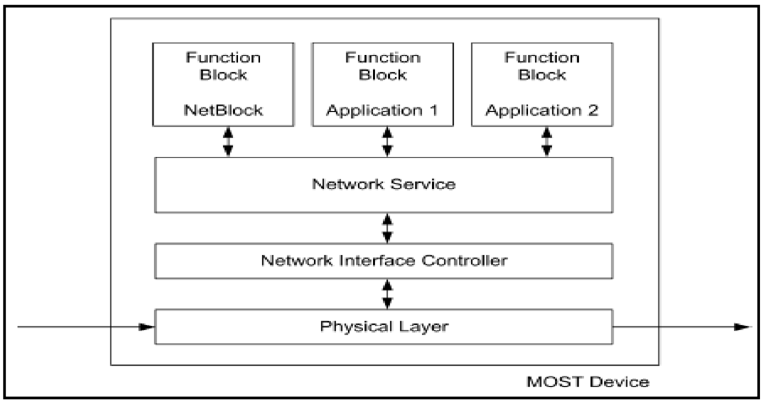


FIG.7.1 MOST FUNCTIONAL BLOCK

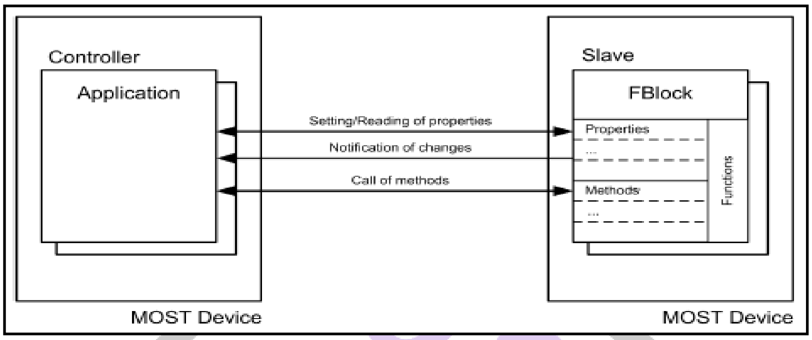
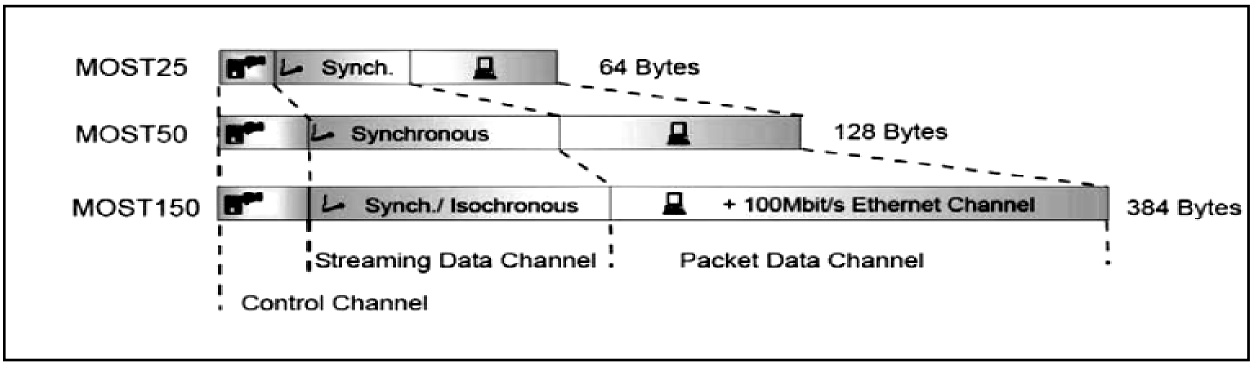


FIG.7.2 MOST DEVICES FBLOCK

A MOST device consists of three parts
• PI (Physical interface)
• NS (Network Services)
A NIC (Network Interface Controller) handles these services. Modern Network Interface Controller's has a processor already built into it and are called Intelligent NICs.
• Function Blocks (FBlocks):These see to the services that the device can fulfill. A MOST device is not connected to a bus in the common sense. It has an out-port and an in-port and passes the information from the in-port to the out-port Function Blocks can have functions with two different targets. FBlocks can be of three types
• Slaves
• Controllers
Controls one or more FBlocks. Function Blocks that can be controlled by other Function Blocks (The system network) .HMIs – Human Machine Interface. Used for the interaction between the user and the device.
The Function Blocks use different types of functions
• Methods, Properties and Events.
Functions that can be started and will lead to the final result after a some period of time. Functions that changes the status of a device. Similar to properties but they do not need an external request. A number of standard Function Blocks that can be the in the same way that has been specified.
Some examples
• Net Block – system FBlock must reside in every device
• Network Master
• Connection Master
• Vehicle – interface for data related to vehicles
• Diagnosis – for diagnostic functions
• Enhanced Testability – necessary for compliance tests
• Audio Amplifier.
• TMC Tuner – Traffic information (Traffic Message Channel) signals



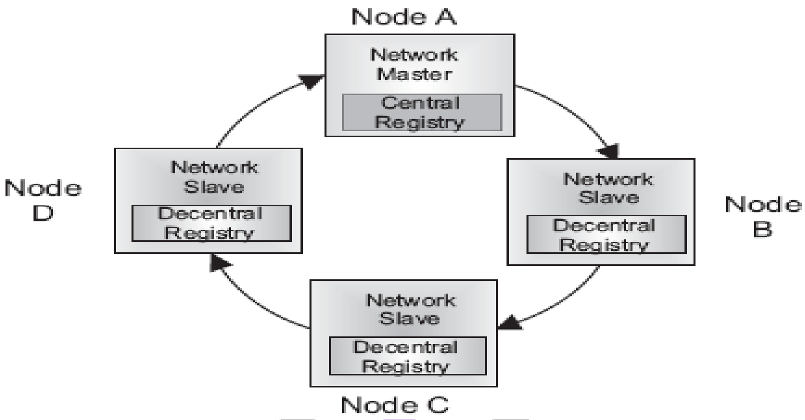


FIG.7.3 MOST SYNCHRONIZATION

VIII. FRAME FORMAT

There are three communication channels open to applications

• Control Channel: For event-oriented transmission with low bandwidth (10 kBits/s) and small package length.
• Asynchronous Channel: Packet oriented transmission with large block size and high bandwidth.
• Synchronous Channel: Continuous data streams that require high bandwidth.

A MOST network can manage upto64 Media Oriented System Transport (MOST) devices in a ring configuration. Virtual star network or other topologies can also be used to set up for MOST networks. Safety critical applications use redundant double ring configurations . Switches or hubs are also possible, but they are not unshakable in the automotive sector. In a MOST network, one device is assigned with the timing master. It continuously supplies the as well as asynchronous(package) data transfer over and ring with MOST frames. A preamble is sent at the starting of the frame transfer. The other devices, known as timing followers, preamble is used for synchronization. Synchronous transfer is used for encoding which allows constant post-sync for the timing followers.

MOST25

MOST25 provides a bandwidth of approximately 23 mega baud for streaming (synchronous) as well as package (asynchronous) data transfer over a physical layer. It is separated into 60 physical channels. The user has the flexibility to select and configure the channels which is of four bytes each for each group. MOST25 provides many services and methods for the de-allocation and allocation of physical channels. MOST25 supports up to15 MPEG-1 channels for audio/video transfer or 15 uncompressed stereo audio channels with CD-quality sound, each of which uses four physical channels (four bytes). MOST also provides a path for transferring control information. The system allows a bandwidth of 705.6 kbit/s with frequency of 44.1 kHz, enabling 2670 control messages per second to be transferred. MOST devices are configured by Control messages and even for synchronous and asynchronous data transfer. The system frequency closely keeps on the CD standard. Control channel is used to transfer Reference data. Some limitations restrict MOST25's efficient data transfer rate is about 10 kB/s. For the protocol overhead, the application can use only 11 of 32 bytes at segmented transfer. Only one third of the control channel bandwidth can be used at any time by MOST node.

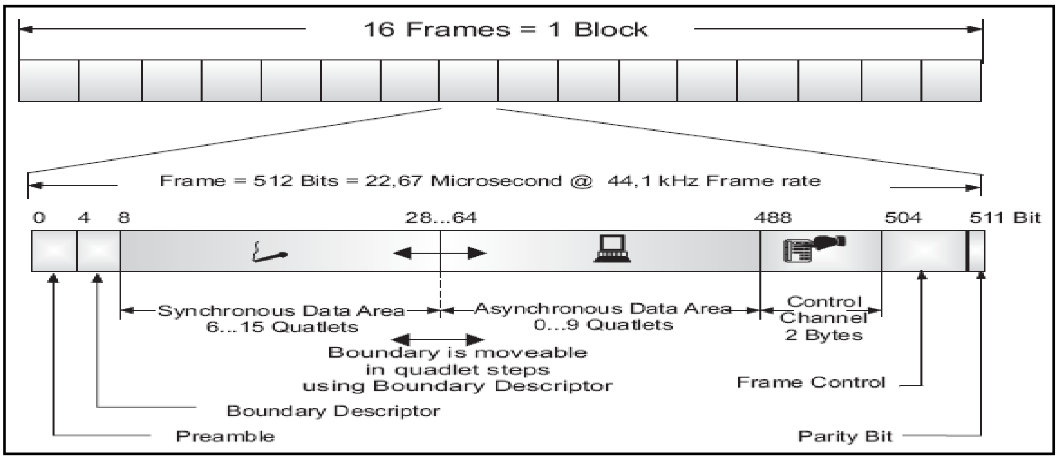


FIG 8.1 MOST25 FRAME FORMAT

MOST50

MOST50 doubles the bandwidth of a MOST25 version. It also increases the frame length to 1M bit. There are three established channels of MOST25i.epacket data channel ,control message channel, streaming data channel is the same, but the length of the control channel and that of the sectioning between the synchronous and asynchronous channels are flexible. While MOST50 is specified to help both electrical and optical physical layers, the existing MOST50 Intelligent Network Interface Controllers (INICs) use [Unshielded Twisted Pair (UTP)](#)to support electrical data transfer.

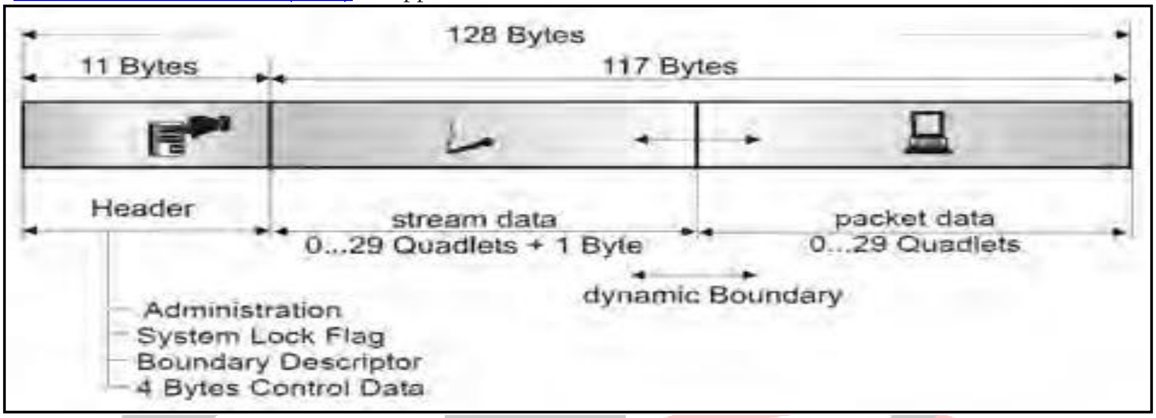


FIG.8.2 MOST50 FRAME FORMAT

MOST150

MOST150 was introduced in October 2007 and it provides a physical layer to device [Ethernet](#) in automobiles. It increases the frame length up to 3072 bits. It also adds an Ethernet channel with changeable bandwidth in addition to the three established channels of the other grades of MOST. MOST150 also permits asynchronous move on the synchronous channel. While the transfer of stream data requires a frequency other than the one listed by the MOST frame rate, it is also possible with MOST150.

MOST150's higher functions and enhanced BW (bandwidth) will allow a multiplex network infrastructure capable of sending all forms of infotainment data, including video, throughout an automobile.

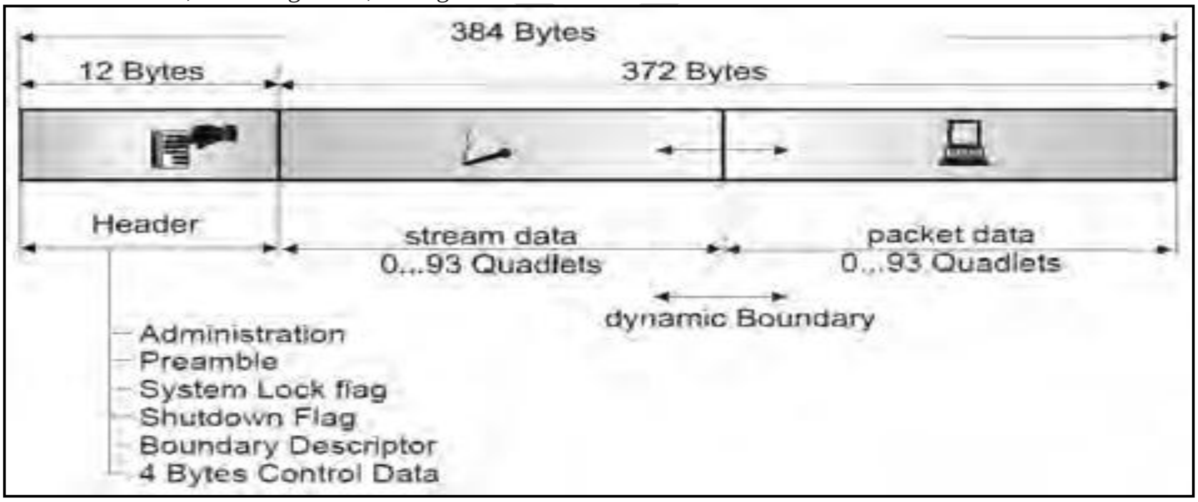


FIG.8.3 MOST150 FRAME FORMAT

IX. ADVANTAGES AND DISADVANTAGES

Advantages of FlexRay

1. Probably has primitives necessary for a critical x-by- wire application.
2. Static segment provides timing guarantees and some fault tolerance.
3. High Flexibility.
4. FlexRay provides the infrastructure to design reliable(safety-critical communication systems) .
5. High speed communication.

Disadvantages of FlexRay

1. After 10 years it is getting matured.
2. Does not provide as complete set of primitives as TTP.
3. More expensive than CAN.

Advantages of MOST protocol over FlexRay

1. Higher Bandwidth.
2. Lighter and more flexible compared to shielded electric data lines.
3. Meets strict EMC requirements.

X. APPLICATIONS

Applications of FlexRay

1. Chassis application.
2. Backbone.
3. High speed CAN replacement.
4. X- By-Wire system.

Applications of MOST Protocol

1. Used for both inside and outside the Car for application purposes.
2. Used in every Car brand.

XI. CONCLUSION

Vehicular Communication Systems is one of the emerging types of networks in which vehicles and roadside units for the communicating nodes. Due to major increase of electronics requirement in modern vehicles, the CAN protocol is initialized in order to reach its operational limits, but it has limited data rate. In order to overcome this drawback, the concept of FlexRay protocol become the standard communication for more advanced networking application like drive by wire, steer by wire etc. Likewise, MOST protocol in MOST150 version is characterized by high flexibility in the scope of band allocation for synchronous and asynchronous data. MOST150 version is characterized by facilitated transmission of data based on IP protocol and by introduced Quality of Service support. MOST bus uses specialized diagnostic tools for elimination of errors in fibre optic ring.

REFERENCES

[1]. FlexRay Communications System – Protocol Specification, v2.1 Revision A, FlexRay Consortium, Dec. 2005.

[2]. *Yi-Nan Xu, I. G. Jang, Y. E. Kim, J. G. Chung, Sung- Chul Lee* "Implementation of FlexRay Protocol with An Automotive Application" 2008 International SoC Design Conference, pp. II-25 to 26

[3]. Robert shaw, Brendan Jackman "An Introduction to Flexray As an Industrial Network" 2008 IEEE

[4]. R.Radhiga and J.Pradeep " Design Of FLEXRAY Communication Controller Protocol For An Automotive Application" IEEE Sponsored 9th International Conference on Intelligent Systems and Control (ISCO)2015

[5].file:///D:/FlexRay%20Automotive%20Communication%20Bus%20Overview%20-%20National%20Instruments.html

[6]. Grzemba, Andreas (2007). MOST: Das Multimedia-BussystemFür Den EinsatzImAutomobil (in German). Poing: Franzis. ISBN 978-3-7723-4149-6.

[7]. Grzemba, Andreas (2011). MOST: The Automotive Multimedia Network; from Most25 to Most150. Poing: Franzis.ISBN 978-3-645-65061-8.

[8].Zimmermann, Werner, Schmidgall, Ralf (2008). Bussysteme in der Fahrzeugtechnik: Protokolle und Standards; Mit 96 Tabellen (in German) (3rd ed.). Wiesbaden: ViewegTeubner. ISBN 978-3-8348-0447-1.