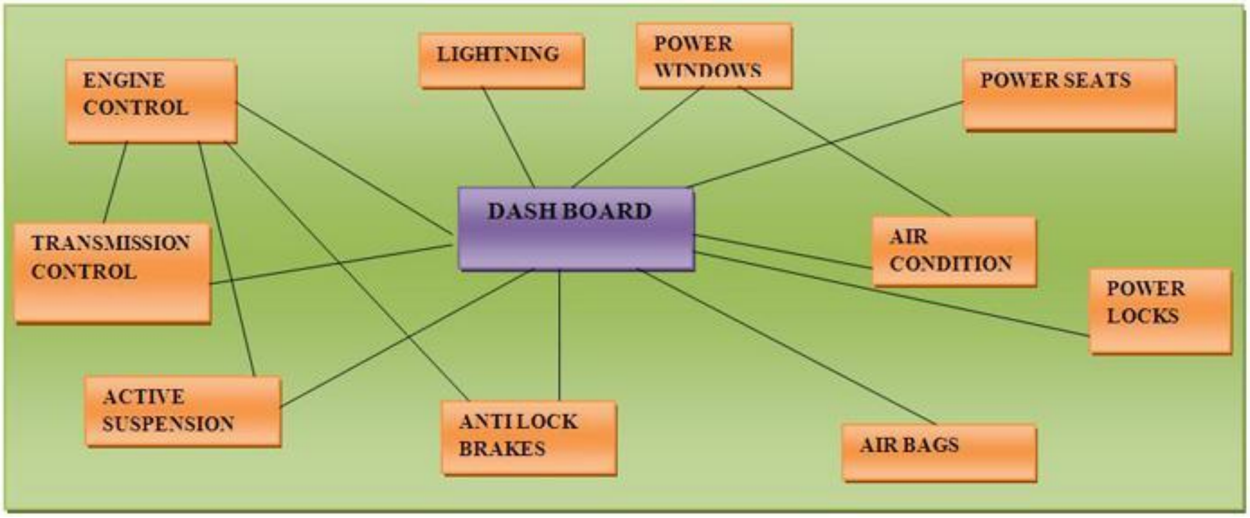
CAN

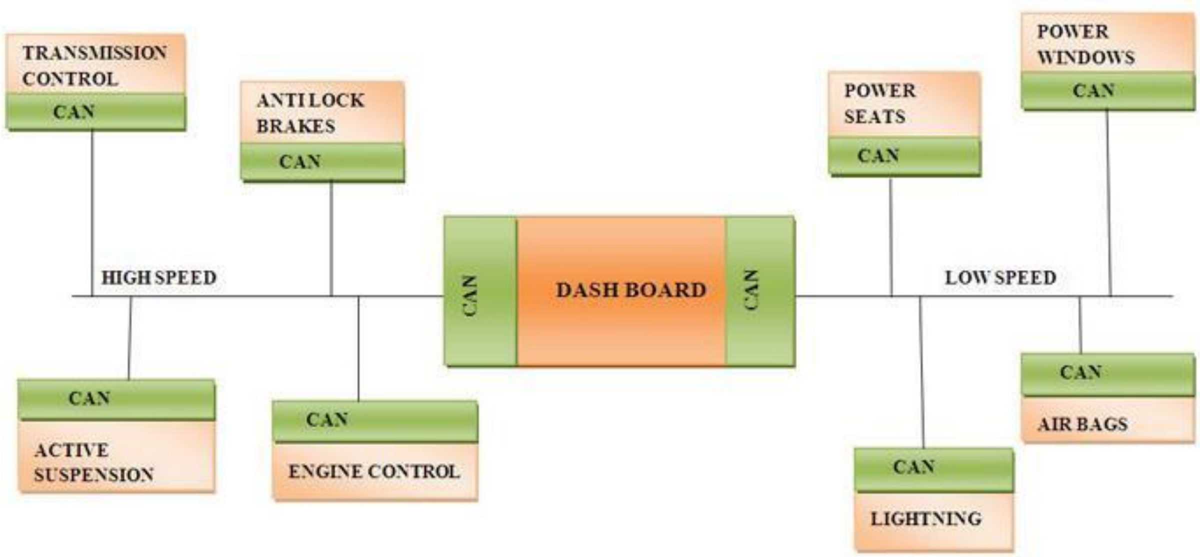
Controller Area Network or CAN protocol is a method of communication between various electronic devices like engine management systems, active suspension, ABS, gear control, lighting control, air conditioning, airbags, central locking etc embedded in an automobile. An idea initiated by Robert Bosch GmbH in 1983 to improve the quality of automobiles thereby making them more reliable, safe and fuel efficient. With the developments taking place in the electronics and semiconductor industry the mechanical systems in an automobile were being replaced by more robust electronics system which had an improved performance. New technologies, products and inventions with added or improved functions started to shape a complete new era for the automobile industry which promised more robust vehicles with use of electronics. The increasing number of electronic devices used communication signals with more complex interrelations between them. Thereby making the life difficult for automobile engineers when they designed systems wherein one electronic device needs to communicate with others to operate. Realizing the problem of communication between different electronic modules Robert Bosch came up with this new protocol called CAN which was first released in 1986. CAN provide a mechanism which is incorporated in the hardware and the software by which different electronic modules can communicate with each other using a common cable.

Need for CAN

A vehicle contains a network of electronic devices to share information/data with each other. For example A spark ignition engine requires a spark to initiate the combustion chamber at the correct time so it communicates with engine control unit that adjusts the exact timing for ignition to provide better power and fuel efficiency.. Another example is of a transmission control unit that changes the ratio of gear automatically with the changing speed. It uses information from engine control unit and various sensors in the system. .Every electronic device has an ECU/MCU (electronic/microcontroller control unit) with its own set of rules to share/transfer information. For two or more devices to interact they should have the necessary hardware and software which allows them to communicate with each other. Before CAN was introduced in the automobiles, each electronic device was connected to other device using the wires (point to point wiring) which worked fine, when the functions in the system were limited. The figure below is the pictorial view of the point to point wiring connection.



One of the major problems for automotive engineers was linking the ECUs of the different devices so that real time information can be exchanged. CAN protocol was designed to address this problem. It laid down the rules through which the various electronic devices can exchange information with each other over a common serial bus. This in turn reduced the wiring connections to a great extend thereby reducing the bulkiness and complexity of the system. The image below shows how the different devices are connected using the CAN protocol.



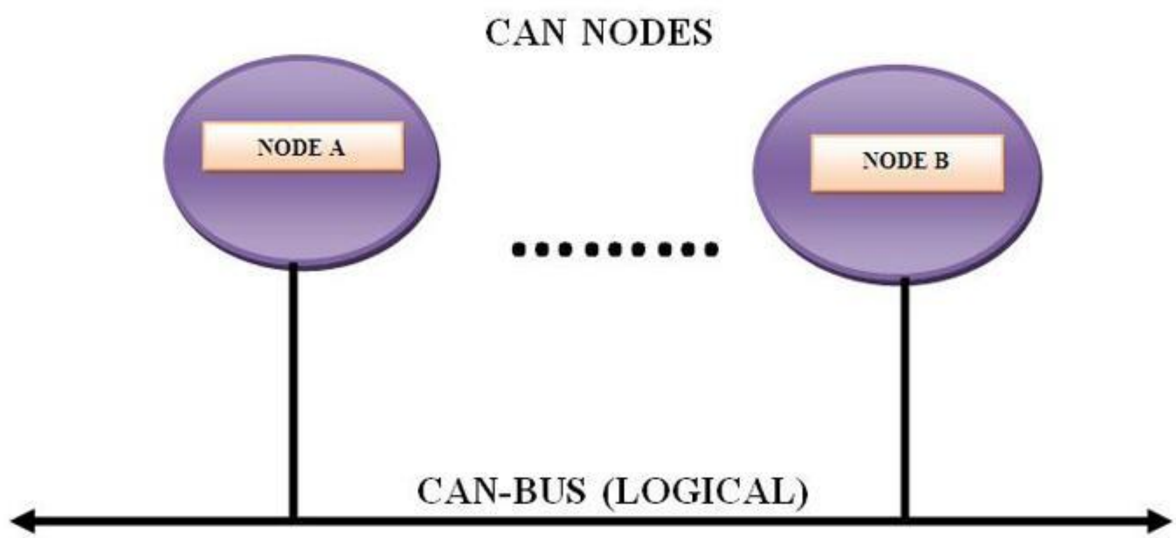
Also the standard technology of time, asynchronous transmitter/receiver did not support multi domain communications. Domain is a group of electronic devices that have almost similar requirements to work in the system. For example CD/DVD PLAYER, GPS system, monitors and displays etc. form a single domain. Similarly air conditioning and climate control, dashboards, wipers, lights doors etc. form another domain. Hence the electronic devices implanted in a vehicle can be classified under different domain. CAN facilitates multi-domain communication for the engineers.

The table below shows the different domains

Domain	Application Area	Examples
Power train	Power generation in engine and transmission through gear box	Engine control
Chassis	Active safety, driving mechanism and assistance	ABS, ASC
Body	Implements body comfort functions	Dash board, Climate control, Wipers etc.
Telemetric	Implements entertainment units	CD/DVD player, for GPS, Multimedia
Passive safety	Safety mechanism	Rollover sensors, Air bag system and Belt pretensions

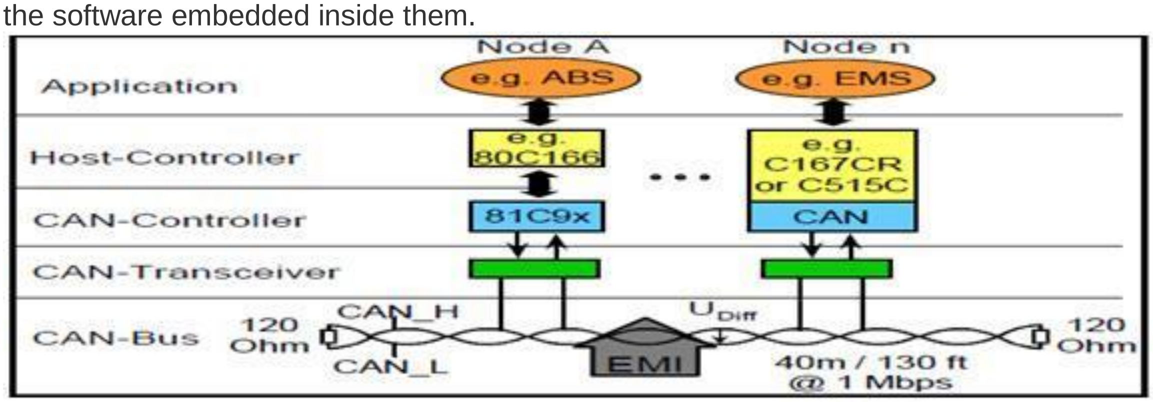
What is CAN protocol and how is it implemented?

CAN protocol can be defined as the set of rules for transmitting and receiving messages in a network of electronic devices. It means that it defines how data is transferred from one device to another in a network. It was designed specifically looking into the needs of the automobile industry. However CANs robust architecture and advantages has forced many industries like Railway, Aircrafts, medical etc to adopt CAN protocol in their systems.



Every electronic device (**also known as the node**) which needs to communicate using the CAN protocol is connected with each other via a common serial bus to transmit and receive messages.

For data exchange to happen among the nodes they must have the necessary hardware and the software embedded inside them.



As shown in the above figure a typical CAN network consists of various nodes. Every node has a Host controller (ECU/MCU) which is responsible for the functioning of the respective node. In addition to the host controller every node has a CAN controller and CAN transceiver. CAN controller convert the messages of the nodes in accordance with the CAN protocols to be transmitted via CAN transceiver over the serial bus and vice versa. CAN controller is a chip which can either be added separately or embedded inside the host controller of the node.

CAN does not follow the master-slave architecture which means every nodes has the access to read and write data on the CAN bus. When the node is ready to send data, it checks availability of the bus and writes a CAN frame onto the network. A frame is defined structure, carrying meaningful sequence of bit or bytes of data within the network. CAN transmitted frame does have address neither of transmitting node or the receiving node. CAN is a message based protocol. A message can be defined as a packet of data which carries information. A CAN message is made up of 10 bytes of data. The data is organized in a specific structure called frame and the information carried in every byte is defined in the CAN protocol. Protocols are generally of two types: address based and message based. In an address based protocol the data packets contain the address of the destination device for which the message is intended. In a message based protocol every message is identified by a predefined unique ID rather than the destination addresses. All nodes on CAN receive the CAN frame and depending on ID on the node CAN decides whether to accept it or not. If multiple nodes send the message at the same time than the node with highest priority (lowest arbitration ID) gets the bus access. Lower priority nodes wait till the bus is available.

BENEFITS

- **Low cost:** As CAN serial bus uses two wires, it offers good price/performance ratio. Also, driven by high volume production of low cost protocol devices, they are relatively cheap.
- **Reliable:** Because of excellent error detection and error handling mechanisms used by CAN, it offers high reliability transmission. It is also highly immune to Electromagnetic Interference
- **Flexibility:** CAN Nodes can be easily connected / disconnected. Also, the number of nodes is not limited by the protocol
- **Good Speed:** CAN supports data rate of 1 MBit/s @ 40m bus length.
- **Multi-master communication:** Any node can access the bus
- **Fault Confinement:** Faulty nodes do not disturb the communication.
- **Broadcast capability:** Messages can be sent to one /many/all nodes.
- **Standardized:** ISO has standardized CAN via ISO-DIS 11898 high speed applications) and ISO-DIS 11519-2 (low speed applications). CAN protocol is also standardized by industry organizations such as SAE-Society of Automotive Engineers.

Error Checking and Fault Confinement

This is one of the attributes of CAN that makes it robust. CAN protocol has five methods of error checking, out of which three are at message level while other two are at bit level. Every frame is simultaneously accepted or rejected by every node in the network. If a node detects an error it transmits an error flag to every node and destroys the transmitted frame and the transmitting node resends the frame

Message level

CRC check

In this stage a 15-bit cyclic redundancy check value is calculated by transmitting node and is transmitted in the CRC field. This value is received by all nodes. Then all the nodes calculate CRC value and matches the results with the transmitted value. If values differ than an Error Frame is generated. Since one of the nodes did not receive the message properly it is resent.

ACK slots

When transmitting node sends a message, a recessive bit is sent in acknowledgement slot. After message is received acknowledge slot is replaced by dominant bit which would acknowledge that at least one node correctly received the message. If this bit is recessive, then none of the node has received the message properly.

Form Error

End of frame, Inter-frame space, Acknowledge Delimiter are fields that are always recessive, if any node detects dominant bit in one of these fields than CAN protocol calls it a violation and a Form Frame is generated and original message is resent after certain period.

Bit level

Stuff error

Bit stuffing - It is a very common technique used in telecommunication and data transmission to insert non -informative bits to have same bit rates or to fill the frames .These extra bits are removed by data link layer to retrieve the original message. This same technique is used in bit error.CAN bus is never idle because it uses NRZ method. After five consecutive bits of the same value, a bit with a complement or opposite value is stuffed into the bit stream. If six bits of the same value are detected between SOF and CRC delimiter, error frame is generated. Upon detection of errors, the transmission is aborted and frame is repeated. If errors continue, then the station or node may switch itself off to prevent the bus from being tied up.

Bit error

A node that is sending the bit always monitors the bus. If the bit sent by transmitter differs from the bit value on the bus then error frame is generated. But there is an exception in case of arbitration field or Acknowledge slot where a recessive bit is sent and a dominant bit is received. Then no Bit Error is occurs when dominant bit is monitored.

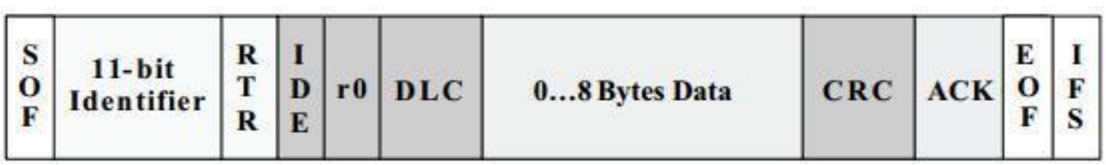
Conclusion

CAN initially developed for in-vehicle networking of automobiles has expanded its applications in various other industries. The application started for luxurious cars is now being used in heavy duty vehicles like trucks, buses, trains and rail vehicles. The unique feature of CAN that allows various electronic units to communicate with each other made it important in healthcare domain. For example intensive care units and operating rooms where time and communication is of utmost importance. Entertainment industry also used CAN protocols to improve features in studios to control lights and door system and to control stage of theatres, event halls etc. Gambling machines and toys are other examples in entertainment field. In the field of science the high energy experiments and astronomical telescope use CAN in embedded network.

Message framing

Messages in CAN are sent in a format called frames. A frame is defined structure, carrying meaningful sequence of bit or bytes of data within the network. Framing of message is done by MAC sub layer of Data Link Layer .There are two type of frames standard or extended .These frames can be differentiated on the basis of identifier fields. A CAN frame with 11 bit identifier fields called **Standard CAN** and with 29 bit identifier field is called extended frame.

Standard frame



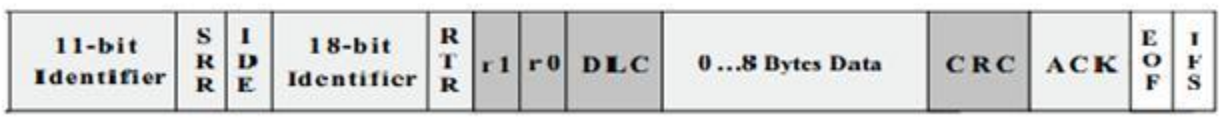
Various fields in standard CAN are as follows-

- SOF - Start of Frame bit. It indicates start of message and used to synchronize the nodes on a bus. A dominant bit in the field marks the start of frame.
- IDENTIFIER - It serves dual purpose one, to determine which node has access to the bus and second to identify the type of message.
- RTR - Remote Transmission Request. It identifies whether it’s a data frame or a remote frame .RTR is dominant when it is a data frame and recessive when it is a remote frame.
- IDE – Identifier Extension. It is used to specify the frame format. Dominant bit is for standard frame and recessive for extended frame.
- R0 - Reversed bit. Not used currently and kept for future use.
- DLC – Data Length Code. It is 4 bit data length code that contains the number of bytes being transmitted.
- DATA– Used to store up to 64 data bits of application data to be transmitted.

- CRC– Cyclic Redundancy Check. The 16-bit (15 bits plus delimiter) cyclic redundancy check (CRC) contains the checksum of the preceding application data for error detection.
- ACK – Acknowledge (ACK) field. It compromises of the ACK slot and the ACK delimiter. When the data is received correctly the recessive bit in ACK slot is overwritten as dominant bit by the receiver.
- EOF– End of Frame (EOF). The 7-bit field marks the end of a CAN frame (message) and disables

  Bit - stuffing, indicating a stuffing error when dominant.
- IFS - Inter Frame Space that specifies minimum number of bits separating consecutive messages. It provides the intermission between two frames andconsists of three recessive bits known as intermission bits. This time allows nodes for internal processing before the start of next frame.

EXTENDED CAN



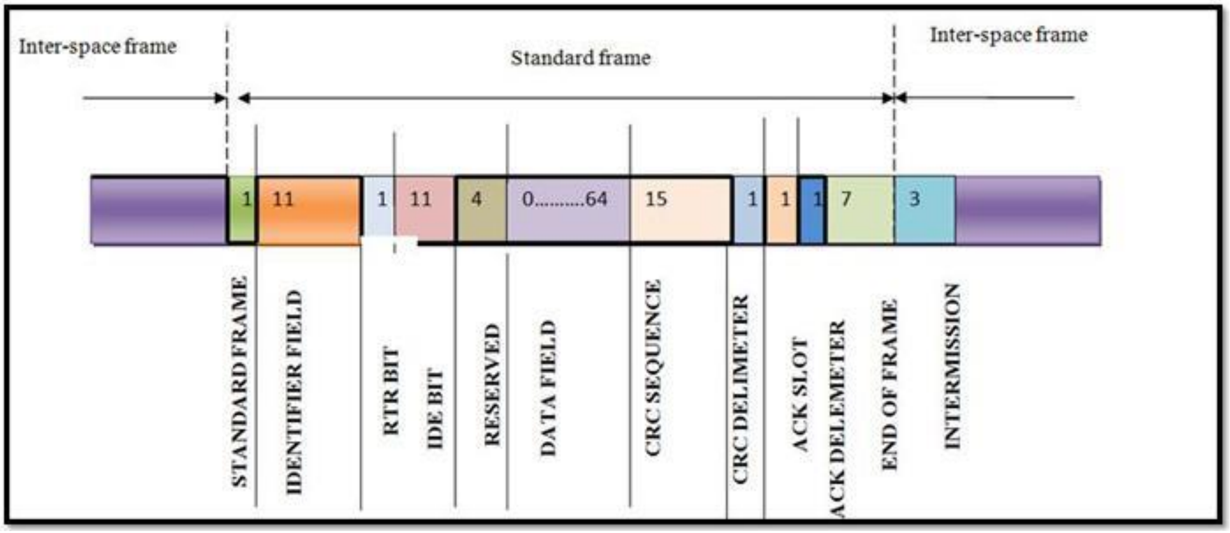
It is same as 11-bit identifier with some added fields

- SRR- Substitute Reverse Request. The SRR bit is always transmitted as a recessive bit to ensure that, in the case of arbitration between a Standard Data Frame and an Extended Data Frame, the Standard Data Frame will always have priority if both messages have the same base (11 bit) identifier.
- R1- It is another bit not used currently and kept for future use.

Message frame

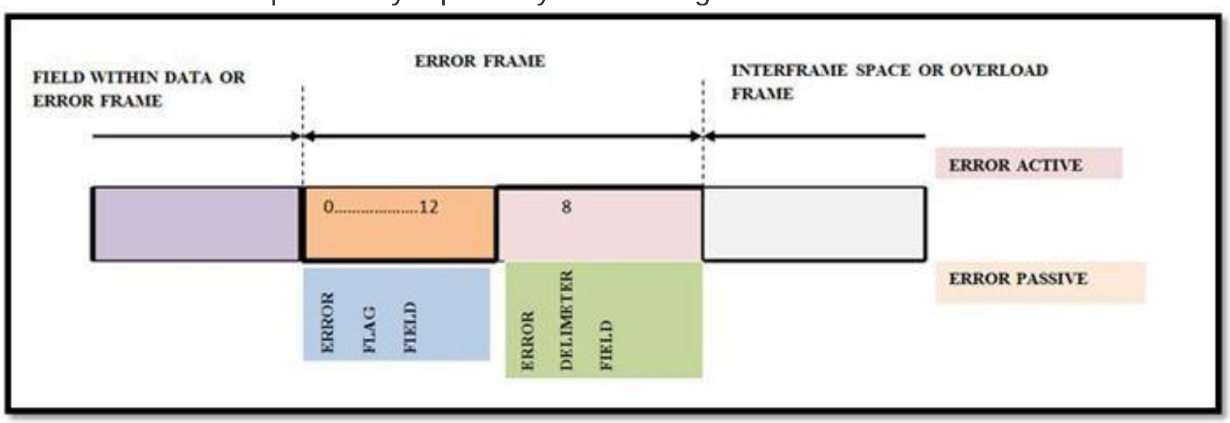
There are four different frames which can be used on the bus.

Data frames- These are most commonly used frame and used when a node transmits information to any or all other nodes in the system. Data Frames consist of fields that provide additional information about the message as defined by the CAN specification. Embedded in the Data Frames are Arbitration Fields, Control Fields, Data Fields, CRC Fields, a 2-bit Acknowledge Field and an End of Frame.

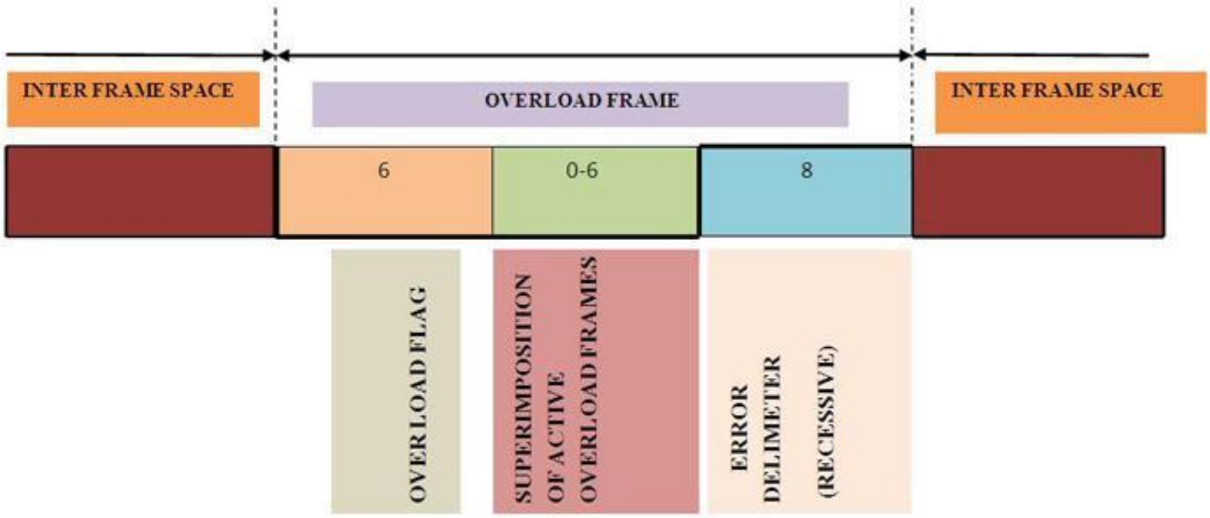


· **Remote frames** - The purpose of the remote frame is to seek permission for the transmission of data from another node. This is similar to data frame without data field and RTR bit is recessive. For example, the microprocessor controlling the central locking on your car may need to know the state of the transmission gear selector from the power train controller.

· **Error frames –** If transmitting or receiving node detects an error, it will immediately abort transmission and send error frame consisting of an error flag made up of six dominant bits and error flag delimiter made up of eight recessive bits. The CAN controller ensures that a node cannot tie up a bus by repeatedly transmitting error frame.



· **Overload frame**-It is similar to error frame but used for providing extra delay between the messages. An Overload frame is generated by a node when it becomes too busy and is not ready to receive.



Arbitration

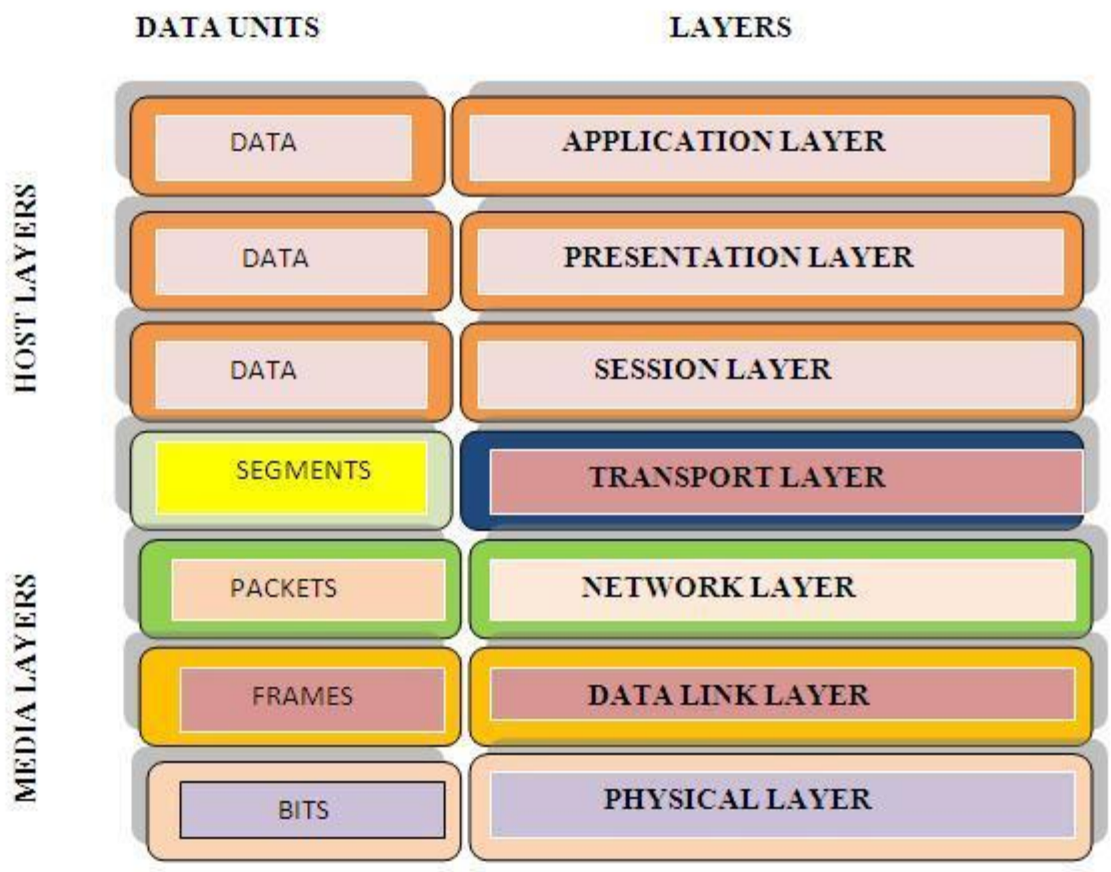
It is a mechanism which resolves the conflict when two or more nodes try to send the message at the same time. In this technique whenever the bus is free any unit can transmit a message. If two or more units starts transmitting at the same time access to the bus is conflicted, but this problem can be solved by arbitration using identifier. During arbitration every transmitter compares the value of transmitted bit with bit value on the bus. If the bit value is same, the node continues to send the bits. But at any time if transmitted bit value is different from bus value the dominant bit overwrites the recessive bits. The arbitration field of the CAN message consists of an 11- or 29-bit identifier and a remote transmission (RTR) bit. The identifier having lowest numerical value has the highest priority. RTR simply distinguishes between remote frame for which RTR is recessive and data frame for which RTR is dominant. If both data frame and remote frame with the same identifier is initiated at the same time data frame will prevail over remote frame. With the concept of arbitration neither information nor time is lost.

CAN as a CSMA protocol

CSMA is a carrier sense, multiple-access protocol in which node verifies the absence of traffic before transmitting on a shared medium such as electrical bus. In CSMA each node on a bus waits for a specific time before sending the message. Once this wait period is over every node has equal opportunity to send the message. Based on pre-programmed priority of each message in identifier field i.e. highest priority identifier wins the bus access. It is implemented on the physical layer of OSI model. Let us understand CSMA with an example. In a discussion every person gets an equal opportunity to voice their thoughts however when a person is talking others keep quiet and listens and waits for their chance to speak (carrier sense). But if two or more people start speaking at the same time then they detect the fact and quit speaking (collision detection).

CAN Architecture

CAN uses the existing OSI reference model to transfer data among nodes connected in a network. The OSI reference model defines a set of seven layers through which the data passes during communication between devices connected in a network. The 7-layered structure of the OSI model is a very robust approach widely adopted in many communication protocols. The figure below gives the clear picture of OSI model.



Each layer has its specific function that supports the layer above and below as described under-

- **Application layer**

It serves as a window for users and application processes to access network services. The common functions of the layers are resource sharing, remote file access, network management, electronic messages and so on.

- **Presentation layer**

The most important function of this layer is defining data formats such as ASCII text, EBCDIC text BINARY, BCD and JPEG. It acts as a translator for data into a format used by the application layer at the receiving end of the station.

- **Session layer**

It allows to establishing, communicating and terminating sessions between processes running on two different devices performing security, name recognition and logging.

- **Transport layer**

The transport layer ensures that messages are delivered error-free, in sequence, and without loss or duplication. It relieves the higher layer from any concern with the transfer of data between them and their peers.

- **Network layer**

It provides end to end logical addressing system so that a packet of data can be routed across several layers and establishes, connects and terminates network connections.

- **Data link layer**

It packages raw data into frames transferred from physical layer. This layer is responsible for transferring frames from one device to another without errors. After sending the frame it waits for the acknowledgement from receiving device. Data link layer has two sub layers:

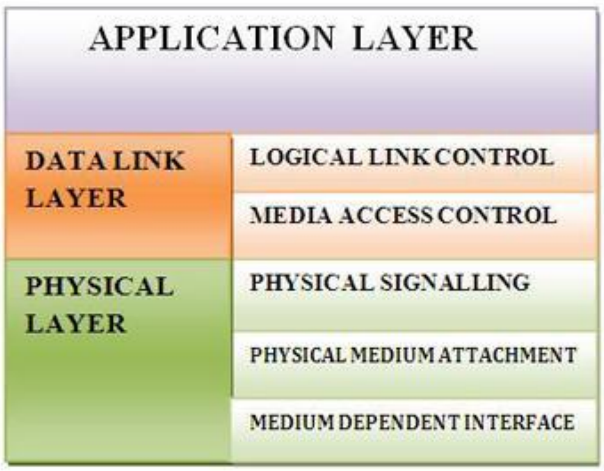
1. MAC (Medium Access Control) layer: It performs frame coding, error detection, signaling, serialization and de-serialization.

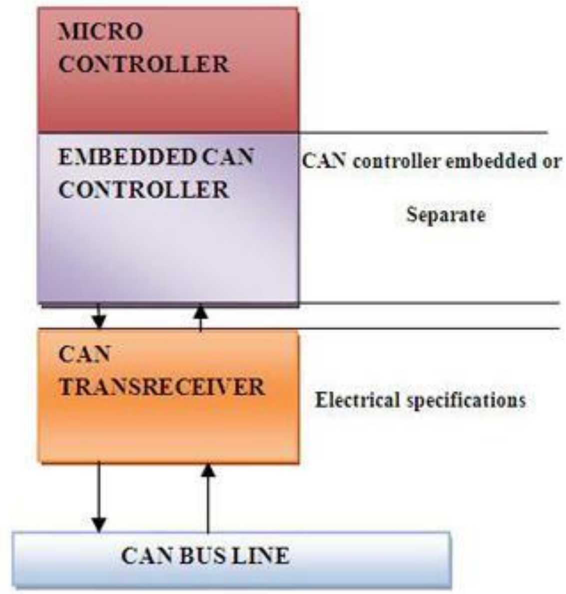
LLC (Logical Link Control) layer: The LLC sub layer provides multiplexing mechanisms that make it possible for several network protocols (IP, Decnet and Appletalk) to coexist within a multipoint network and to be transported over the same network medium. It performs the function of multiplexing protocols transmitted by MAC layer while transmitting and decoding when receiving and providing node-to-node flow and error control.

· Physical layer

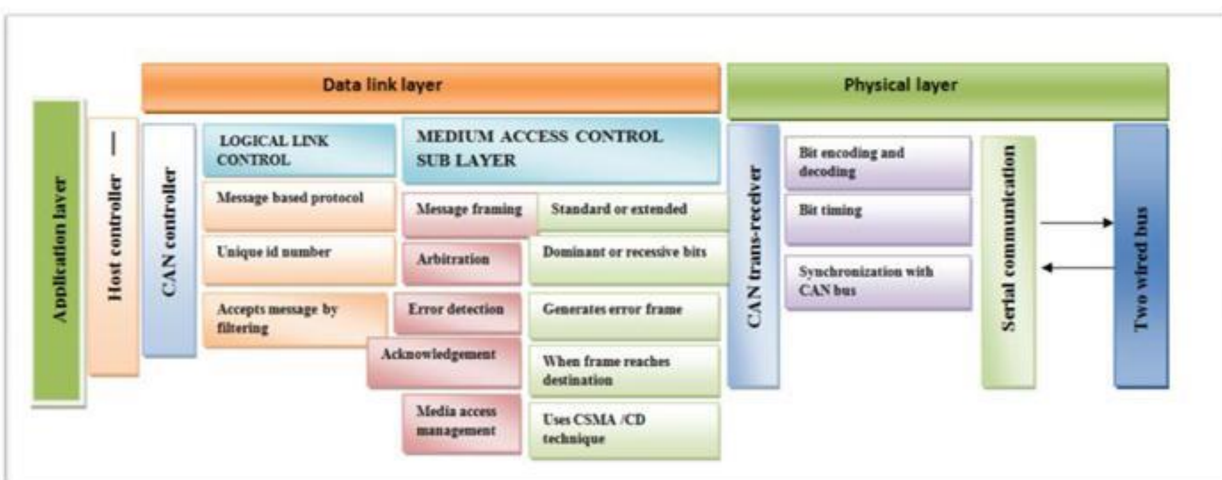
The physical layer transmits bit from one device to another and regulates the transmission of bit streams. It defines the specific voltage and the type of cable to be used for transmission protocols. It provides the hardware means of sending and receiving data on a carrier defining cables, cards and physical aspects.

CAN protocol uses lower two layers of OSI i.e. physical layer and data link layer. The remaining five layers that are communication layers are left out by BOSCH CAN specification for system designers to optimize and adapt according to their needs.





Function of each part is shown with the figure below.



The concept of the CAN protocol can be understood using the figure above. Every node has a Host controller also known as micro-controller which is a small and low-cost computer .Host controller implements application layer of OSI model. Micro-controller gathers information from other electronic control units like braking, steering, power windows etc. to communicate with other nodes and transfers it to CAN controller. CAN controller incorporate logical link control and MAC medium access control of data link layer. LLC allows filtering of messages by using unique ID on each message then MAC sub layer frames the message. Once, framing is done it is followed by arbitration, error detection and acknowledgement that all comes under MAC sub layer of data link. The frame is transferred to CAN trans-receiver, for encoding and decoding. Finally CAN trans-receiver synchronizes with the CAN bus to send the message to anther node.

Terms associated with CAN protocol.

BUS VALUES

Binary values in CAN protocol are termed as dominant and recessive bits.

- CAN define the logic "0" as dominant bit.
- CAN define the logic "1" as recessive bit.

In the CAN system dominant bit always overwrites the recessive bit.

Message based communication

A Message is packet of data that carries the information to be exchanged between the nodes. Each message in CAN has a unique identification number. The identification number is specified according to the content of the message and stored in message identifier. This identification number is also unique within the network so when the transmitting node places the data on the network for access to all nodes it checks unique ID number to allow the message to pass through the filter and rest are ignored. This is done to save the time spent on sorting. With message based protocol other nodes can be added without re-programming since the units connected to the bus have no identifying information like node addressing. So there is no change needed in the software and hardware of any of the units connected on the bus.