International Journal of Innovative Research in Science and Engineering
Vol. No.2, Issue 05, May 2016
www.ijirse.com
ISSN: 2454-9665

A CONTROLLER AREA NETWORK ARCHITECTURE USING ETHERNET FOR AUTOMATION

Prof. Aashish A. Gadgil

Dept. of E&C Engineering,

KLS Gogte Institute of Technology, Belagavi, India

ABSTRACT

Controller Area Network (CAN) was created for automotive applications as a method for enabling robust serial communication. The goal was to make automobiles more reliable, safe and fuel-efficient while decreasing wiring harness weight and complexity. Since its inception, the CAN protocol has gained widespread popularity in industrial automation and automotive/truck applications. Other markets where networked solutions can bring attractive benefits like medical equipment, test equipment and mobile machines are also starting to utilize the benefits of CAN. In this paper a solution to the problem related to controller area network (CAN) and Ethernet is presented. Control and monitoring via a CAN and Ethernet interface thus combining the two technologies is presented here. A new method to controlling all the devices connected via a CAN through the use of Ethernet is proposed in this paper which can lead to building automation.

Keywords: Building automation, Ethernet, CAN, CAN master and slave, nodes

I. INTRODUCTION

Controller Area Network (CAN) was created by Robert BOSCH for automotive applications as a method for enabling robust serial communication. CAN is used in medical equipment, test equipment and mobile machines. The protocol was developed 1980 by BOSCH for automotive applications[1]. The CAN standard includes: 1. Physical layer, 2. Data-link layer, 3.Some message types, 4. Arbitration rules for bus access, 5. Methods for fault detection and fault confinement. CAN is used to control and automate various devices in buildings leading to building automation. By combining both CAN and Ethernet the scope of both can be increased. This combination of CAN and Ethernet will increase the scope of CAN in automation to multiple buildings instead of a single large building. In this paper, I propose different architectures and ways in which both CAN and Ethernet can be combined. In section 2, the two technologies Controller Area network and Ethernet are discussed in detail. In section 3, I have proposed a new method integrating Ethernet and CAN together. Here I discuss the architecture and propose the system. In section 4, I conclude with the benefits of combining both the technologies and the need to use them together. In section 5, future scope is presented to extend the usage to other areas to help in automation.

ISSN: 2454-9665

II. CONTROLLER AREA NETWORK

CAN has gained widespread use in Industrial Automation, Automotive industry etc. The template of layered approach was created by ISO called ISO Open Systems Interconnections (OSI) Network Layering Reference Model[1]. CAN protocol itself implements most of the lower two layers of this reference model. The Controller Area Network (CAN) is a serial communications protocol suited for networking sensors, actuators, and other nodes in real-time systems. In this section, I first give a general description of CAN including its message formats, principle of bus arbitration, and error-handling mechanisms.

2.1 Description

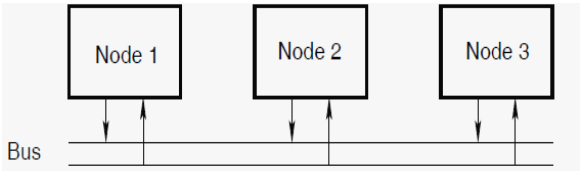


Fig 1: Three nodes connected through a CAN bus

A CAN bus with three nodes is depicted in Fig. 1. The CAN specification [4] defines the protocols for the physical and the data link layers, which enable the communication between the network nodes. The application process of a node, e.g., a temperature sensor, decides when it should request the transmission of a message frame. The frame consists of a data field and overhead, such as identifier and control fields. Since the application processes in general are asynchronous, the bus has a mechanism for resolving conflicts. For CAN, it is based on a non-destructive arbitration process. The CAN protocol therefore belongs to the class of protocols denoted as carrier sense multiple access/collision avoidance (CSMA/CA), which means that the protocol listens to the network in order to avoid collisions. CSMA/CD protocols like Ethernet have instead a mechanism to deal with collisions once they are detected. CAN also includes various methods for error detection and error handling. The communication rate of a network based on CAN depends on the physical distances between the nodes. If the distance is less than 40 m, the rate can be up to 1 Mbps.

2.1.1 CAN Protocol Basics

CAN is a CSMA/CD protocol. Logic states define dominant and recessive. In CAN logic bit 0 is dominant bit and logic bit 1 is recessive bit. Arbitration used to decide. CAN is a message based protocol and not an address based protocol.

- Message formats

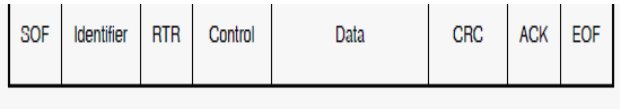


Figure 2: CAN message frame [2]

CAN distinguishes four message formats: data, remote, error, and overload frames. Here we limit the discussion to the data frame, shown in Fig. 2. A data frame begins with the start-of-frame (SOF) bit. It is followed by an eleven-bit identifier and the remote transmission request (RTR) bit. The identifier and the RTR bit form the arbitration field. The control field consists of six bits and indicates how many bytes of data follow in the data field. The data field can be zero to eight bytes. The data field is followed by the cyclic redundancy checksum

ISSN: 2454-9665

(CRC) field, which enables the receiver to check if the received bit sequence was corrupted. The two-bit acknowledgment (ACK) field is used by the transmitter to receive an acknowledgment of a valid frame from any receiver. The end of a message frame is signaled through a seven-bit end-of-frame (EOF). There is also an extended data frame with a twenty-nine-bit identifier (instead of eleven bits) [2].

- Arbitration

Arbitration is the mechanism that handles bus access conflicts. Whenever the CAN bus is free, any unit can start to transmit a message. Possible conflicts, due to more than one unit starting to transmit simultaneously, are resolved by bit-wise arbitration using the identifier of each unit. During the arbitration phase, each transmitting unit transmits its identifier and compares it with the level monitored on the bus [3]. If these levels are equal, the unit continues to transmit. If the unit detects a dominant level on the bus, while it was trying to transmit a recessive level, then it quits transmitting (and becomes a receiver).

The arbitration phase is performed over the whole arbitration field. When it is over, there is only one transmitter left on the bus. The arbitration is illustrated by the following example with three nodes (see Fig. 3). Let the recessive level correspond to "1" and the dominant level to "0", and suppose the three nodes have identifiers *Ii*, *i* = 1, 2, 3, equal to

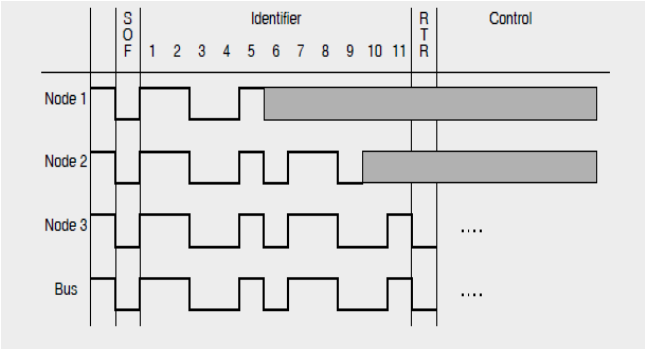


Figure 3: Arbitration

- Error handling

Error detection and error handling are important for the performance of CAN. Because of complementary error detection mechanisms, the probability of having an undetected error is very small. Error detection is done in five different ways in CAN [1]: bit monitoring and bit stuffing, as well as frame check, ACK check, and CRC. Bit monitoring simply means that each transmitter monitors the bus level, and signals a bit error if the level does not agree with the transmitted signal. (Bit monitoring is not done during the arbitration phase.) After having transmitted five identical bits, a node will always transmit the opposite bit. This extra bit is neglected by the receiver. The procedure is called bit stuffing, and it can be used to detect errors. The frame check consists of checking that the fixed bits of the frame have the values they are supposed to have, e.g., EOF consists of seven recessive bits. During the ACK in the message frame, all receivers are supposed to send a dominant level. If the transmitter, which transmits a recessive level, does not detect the dominant level, then an error is signaled by the ACK check mechanism. Finally, the CRC is that every receiver calculates a checksum based on the message and compares it with the CRC field of the message.

Every receiver node obviously tries to detect errors within each message. If an error is detected, it leads to an immediate and automatic retransmission of the incorrect message. In comparison to other network protocols,

ISSN: 2454-9665

this mechanism leads to high data integrity and a short error recovery time. CAN thus provides elaborate procedures for error handling, including retransmission and reinitialization. The procedures have to be studied carefully for each application to ensure that the automated error handling is in line with the system requirements.

2.2 Ethernet

Ethernet: uses CSMA/CD. The algorithm describing the working of Ethernet is as shown below [4]:

A: sense channel, if idle
then {
transmit and monitor the channel;
If detect another transmission
then {
abort and send jam signal;
update # collisions;
delay as required by exponential backoff algorithm;
goto A
}
else {done with the frame; set collisions to zero}
}
else {wait until ongoing transmission is over and goto A}
Jam Signal: make sure all other transmitters are aware of collision; 48 bits;

The Exponential Backoff algorithm has the constraints described below:

- Goal: adapt retransmission attempts to estimated current load
  - heavy load: random wait will be longer
- First collision: choose K from {0,1}; delay is K x 512 bit transmission times
- After second collision: choose K from {0,1,2,3}…
- After ten or more collisions, choose K from {0,1,2,3,4,…,1023}

i) Ethernet Frame Structure

Sending adapter encapsulates IP datagram (or other network layer protocol packet) in Ethernet frame

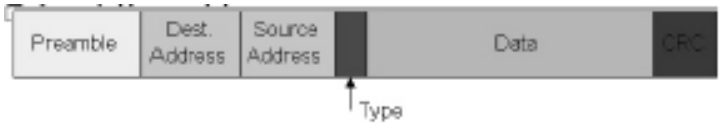


Figure 4: Frame structure [5]

The Ethernet frame is described below:

- Preamble: 7 bytes with pattern 10101010 followed by one byte with pattern 10101011 used to synchronize receiver, sender clock rates
- Addresses: 6 bytes, frame is received by all adapters on a LAN and dropped if address does not match
- Type: indicates the higher layer protocol, mostly IP but others may be supported such as Novell IPX and AppleTalk)

- CRC: checked at receiver, if error is detected, the frame is simply dropped

ii) Ethernet v2 Format

(7)	(1)	(6)	(6)	(2)	(46 – 1500)		(4)
Preamble	start of frame	destination address	source address	protocol	data	padding	checksum

Figure 5: Frame structure for ethernet v2 [5]

The frame structure consists of the components described below:

1. The preamble consists of the bit pattern 10101010, repeated 7 times. After translation into the encoding used on the cable (Manchester encoding) this generates a 10 MHz. square wave that's used to synchronize the station's bit clock. The framing byte is 10101011.

2. The destination and source addresses are 48 bits, and can specify an individual interface or a group of interfaces (multicast or broadcast). Addresses can also be global or local.

3. The protocol field specifies the protocol associated with the data.

4. The next fields are the data itself, plus padding (if necessary) to increase the frame size (excluding the preamble and start of frame bytes) to the minimum size (64 bytes) required for proper propagation of a collision to all stations on the cable.

5. The final field is a CRC checksum.

6. What sort of structure do we see in Ethernet addresses? In the most significant byte of the address, the least significant two bits are reserved to specify global/local and group/individual.

iii) Ethernet Technologies

10Base 2

The features of 10Base2 are listed below [6]:

- 10: 10Mbps; 2: under 200 meters max cable length thin coaxial cable in a bus topology

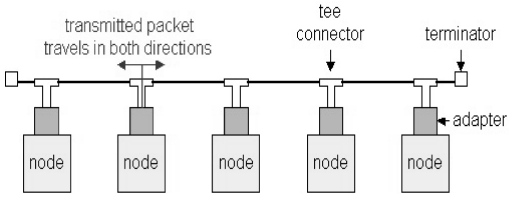
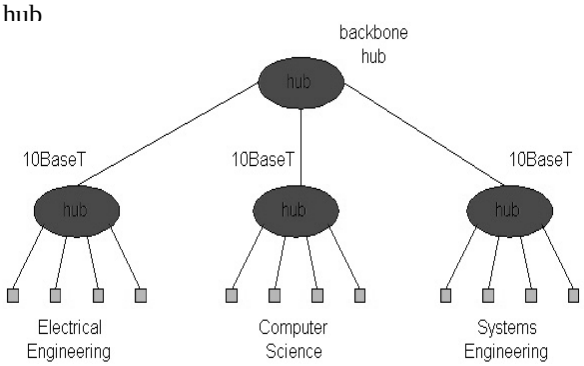


Figure 6: 10Base2

Figure 6: 10Base2

- repeaters used to connect multiple segments
- repeater repeats bits it hears on one interface to its other interfaces: physical layer device only!
- 10BaseT and 100BaseT
- 10/100 Mbps rate; latter called "fast ethernet"
- T stands for Twisted Pair
- Hub to which nodes are connected by twisted pair, thus "star topology"

❒ CSMA/CD implemented at hub



Hubs essentially are repeaters operating at bit level and have limitations:

The limitations of Hub are listed below:

- single collision domain results in no increase in max throughput
- multi-tier throughput same as single segment throughput
- Thus, limits on number of nodes in same collision domain and on total allowed geographical coverage
- cannot connect different Ethernet types (e.g., 10BaseT and 100baseT)

Bridges

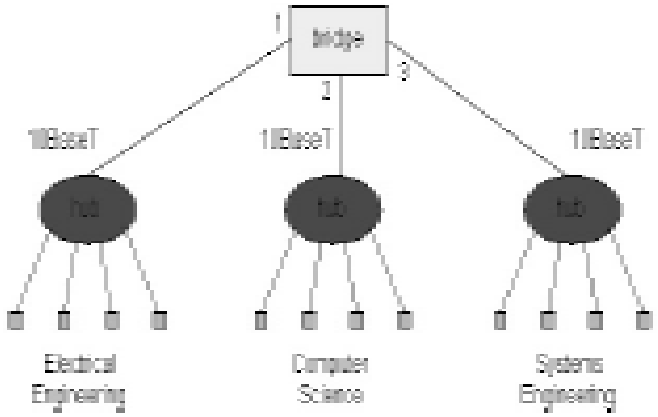


Figure 8: Usage of Bridges

The features of devices using bridges are listed below:

- Link Layer devices: operate on Ethernet frames, examining frame header and selectively forwarding frame based on its destination
- Bridge isolates collision domains since it buffers frames
- When frame is to be forwarded on segment, bridge uses CSMA/CD to access segment and transmit

Interconnection Without Backbone

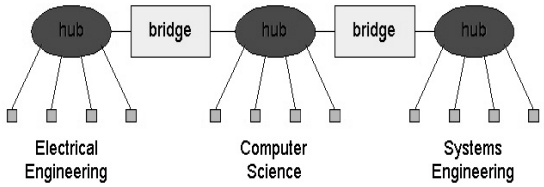


Figure 9: Interconnection of bridge without using backbone

ISSN: 2454-9665

This type of connection is not recommended for two reasons:

- single point of failure at Computer Science hub
- all traffic between EE and SE must path over CS segment

III. SYSTEM PROPOSAL

The system proposed will be based on the approach shown in the figure below:

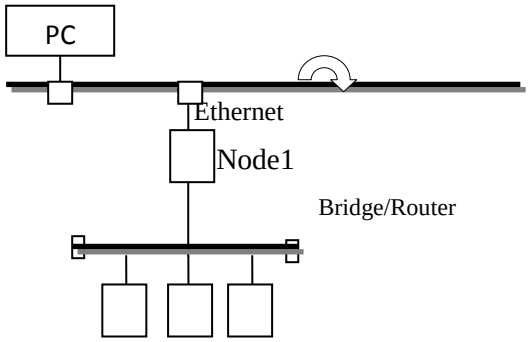


Figure 10: Proposed System

Data is sent through the Ethernet as multicast message. Each multicast message is sent through a router/bridge. Several functions are grouped properly for multicast messages. Example:{NM, EM, ERR, Time sync},{Temp, EM, ERR, Time Sync}, {PM, EM, ERR, Time Sync, Data}

The messages are prioritized. In case of CSMA i.e. Collision the messages are decided based on the priorities. In CAN node design the CAN controller has separate transmit and receive paths. So as the node is writing onto the bus it is also listening back at the same time. “Logic 1” is recessive bit and “Logic 0” is dominant bit.

The functionalities that are required in each CAN node connected to the Ethernet for successful automation are:

1. Lightings Control: This can be performed in two steps: i) Motion/Occupancy Sensor: Using this it will be easy identify whether a particular room is occupied so that ambience can be improved.

ii) Timer controlling ambience: After getting output from the sensor a timer is started. After some time the ambience of the room is improved and the timer is used to keep track for how much time the ambience should change.

2. Fire prevention: This functionality is implemented by setting the fire alarms to go off as soon as there is fire detected using a heat sensor.

3. Security: Video and audio monitoring system are used due to operator problems. In such a case the user can be notified through a system to the system or mobile, pager etc. Security cameras can also be used.

4. Smoke detection: Smoke alarm is sounded in case of smoke. The valves (plates) are turned off to stop air entering in case of smoke alarm and exhaust is started.

5. Temperature control and Air conditioning: A temperature sensor is used to detect temperature at any point in time and control the turning on and off of the ventilation and air conditioning system.

6. Pressure management: Mechanical parts are needed to check or move valves when pressure sensing is to be done using a pressure sensor.

ISSN: 2454-9665

So all the functionalities can be grouped into four major categories:

1. Controlling
2. Monitoring
3. Optimizing
4. Reporting

3.1 Architecture Diagram

Design 1

Each system present in this design consists of one Master node and a set of Slave nodes connected to different devices in the network. A master will control the devices within each system only using the slave nodes as all of them are connected through a single CAN. Each system has its own CAN through which all the devices in that system are controlled. All the different CANs are connected to each other via Ethernet to provide expandability. Over the Ethernet the controller operates which sends multicast messages. The controller can be a simple personal computer also as shown in fig 11. For example, each system can be one department of a huge organization connected over a private Ethernet to provide a centralized control.

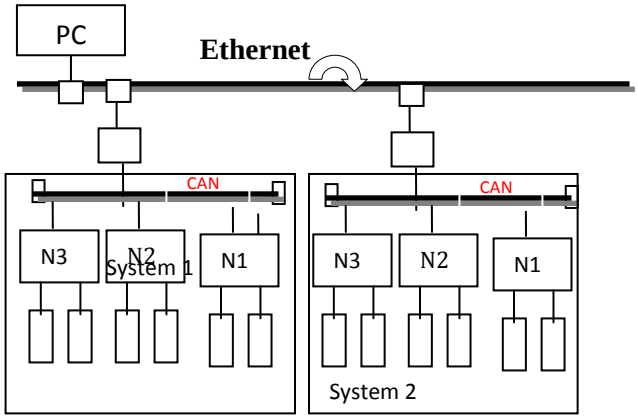


Figure 11: Architectural design 1

Design 2

In the design proposed in fig. 12 the slave nodes and master nodes are separated as shown in the figure below. Unlike in the previous design all the master nodes are connected via a single CAN. Each master node controls a set of slave nodes connected to different devices. All the slaves of a system are chosen such that if they are controlled then an entire building or department could be controlled by a single master node. All the slaves of a system are connected via a separate CAN. Master node of a particular system is selected via Ethernet and performs the control operations through the bridge connected to a particular system CAN i.e. CAN of particular system as shown in the figure below. Whenever a controlling operation needs to be performed the system is identified and the operation is performed via Cloud by the master thus reducing the chances of failure of the design.

International Journal of Innovative Research in Science and Engineering
Vol. No.2, Issue 05, May 2016
www.ijirse.com
ISSN: 2454-9665

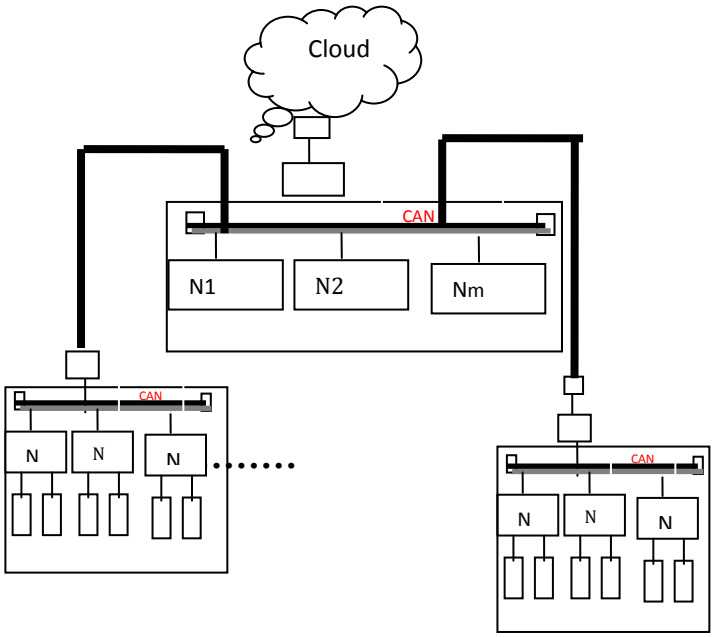


Figure 12: Architectural design 2

3.2 System Diagram

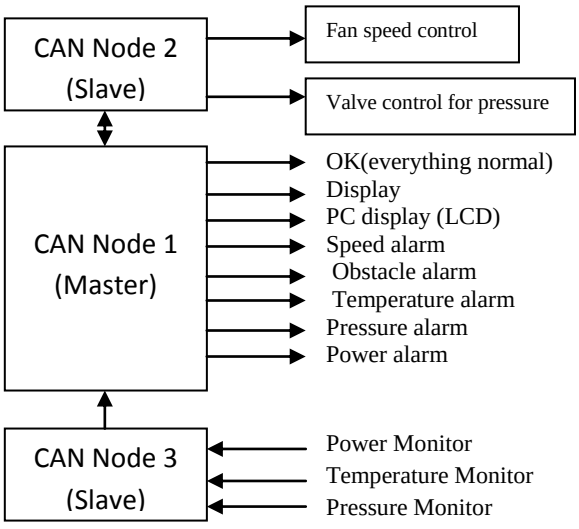


Figure 13: System proposed

Each system proposed in fig. 13 consists of a master node and two slave nodes. Based on identifier and priorities the functions can be performed. The information is sent from the slave nodes to the master node via identifier and controlling operations are done by the master node through the CAN. The PC is connected to the master node through the serial communication port. The master node controls all the slave nodes connected by checking the signals coming from a slave and taking necessary action. The action taken may be related to

ISSN: 2454-9665

temperature, power, pressure etc. It notifies the user of the problem occurred and takes necessary action like maintaining a predetermined temperature in HVAC system.

IV. CONCLUSION

The system architecture proposed in this paper provides evidence of how well CAN be implemented in an organization/building. CAN is being used increasingly because of its inherent properties which go a long way to help in building automation. The capability of CAN will be increased exponentially when it is combined with the existing Ethernet thus providing a larger controlling functionality. The two combined will pave way for a new networking technology with a higher controlling and networking capacity because the limitation of connectivity of CAN is increased by combining it with Ethernet. The use of CAN is not only in Building automation but in other areas also like automobiles which is also a huge market.

V. FUTURE SCOPE

The proposed method could be further expanded in scale to integrate the automobiles with the buildings so that both can be controlled simultaneously.

REFERENCES

[1] Introduction to the Controller Area Network (CAN) Steve Corrigan, Texas Instruments.
[2] Controller Area Networks And The Protocol Can For Machine Control Systems, Lars-Berno Fredriksson
[3] Controller Area Network (CAN) Application in Security System Mazran Esro, Amat Amir Basari , Siva Kumar S, A. Sadhiqin M I , Zulkifli Syariff.
[4] Comparison between Networked Control System behaviour based on CAN and Switched Ethernet networks, B. Brahimi, E. Rondeau, C. Aubrun.
[5] Remote Control via Can/Ethernet Gateway Using GPRS, James Agajo, Idigo V.E. and Isaac Avazi, IJCTE.
[6] TDTS09 Computer networks and internet protocols, Juha Takkinen.
[7] Communication Systems For Building Automation And Control, Wolfgang Kastner, Georg Neugschwandtner, Stefan Soucek, And H. Michael Newman.
[8] B.J. Casey, “Implementing Ethernet in the industrial environment”, in IEEE Industry Applications Society Annual Meeting, Seattle, WA, vol. 2, pp. 1469{1477, Oct. 1990.