



International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 1, Issue 8, October 2013

STUDY OF RESPONSE TIME ANALYSIS FOR CAN BUS

Parikshit mishra¹, Rohit vaishya²

M.Tech, Dept. of EC, IES College of technology, Bhopal, India¹

Assistant Professor, Dept. of EC, IES College of technology, Bhopal, India²

ABSTRACT: The Controller Area Network (CAN) (1990) is one of the most usable bus networks in automobile industries as well as in many other industries for CAN standardization (1993). The basic concept of this review paper is to analyze response time for data transferring messages in CAN bus network for WCRT analysis till today scenario (2013). This paper includes the worst-case response time analysis which is determinable and probable response time analysis which is stochastic and this method will guarantee provided that the response time of messages would not exceed their deadlines. This is because of the data traffic shaping strategy is very beneficial in terms of reducing response times especially at high data load levels. This is well designed bus network for transferring short real time messages.

Keywords: Real time system, CAN protocol, Scheduling, Worst case time task scheduling, Response time

I. INTRODUCTION

The CAN bus was developed by BOSCH (1983) as a multi-master, serial communication bus message broadcast system that specifies a maximum signaling rate of 1 megabit per second (bps)(5), CAN bus is a message-based protocol, designed specifically for automotive applications but now also used in other areas such as aerospace, maritime, industrial automation and medical equipment. CAN is a serial communication bus Unlike a traditional network such as USB or Ethernet, CAN does not send large number of blocks of data point-to-point from node A to node B under the supervision of a central bus master. In a CAN network, many short messages like temperature or RPM are broadcast to the whole network, which provides for data consistency at every node of the system. This bus is widely used in vehicle for data transmission for robust communications.

II. REAL TIME SYSTEM

Creating an embedded system with large number of processing elements talking over a network is definitely more complicated than using single large microprocessors to perform the same tasks for communication. Many distributed systems are required because the devices that the processing elements communicate with are physically separated in each case of communication. While the deadlines for processing the messages are short, it may be cost-effective to put the processing elements where the messages are located rather than build a higher speed communication network to carry the data to distant, fast processing elements.

A. REQUIREMENT OF NETWORKS IN EMBEDDED SYSTEM

Creating an embedded system with large number of processing elements talking over a network is definitely more complicated than using single large microprocessors to perform the same tasks for communication. Many distributed systems are required because the devices that the processing elements communicate with are physically separated in each case of communication. While the deadlines for processing the messages are short, it may be cost-effective to put the processing elements where the messages are located rather than build a higher speed communication network to carry the data to distant, fast processing elements.

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 1, Issue 8, October 2013

B. CAN PROTOCOL

The CAN protocol, like many networking protocols, can be decomposed into many abstraction layers. Different communication protocols are designed based on their application in different fields. Based on uses they are divided into "higher end" and "lower end" protocols. Higher end protocols are used for overall factory system information, known as factory bus. Lower end protocols are used for inter-processor communication, known as field bus [2]. Now day's automobile industries develop different electronic systems to maintain the growing demand for more comfort, safety and efficient fuel reduction technology. So the whole system is divided into several subsystems. Each subsystem is connected with each other with a dedicated wire line which increases the complexity of the wiring. To overcome this type of wiring complexity new protocol like Controller Area Network (CAN) protocol developed.

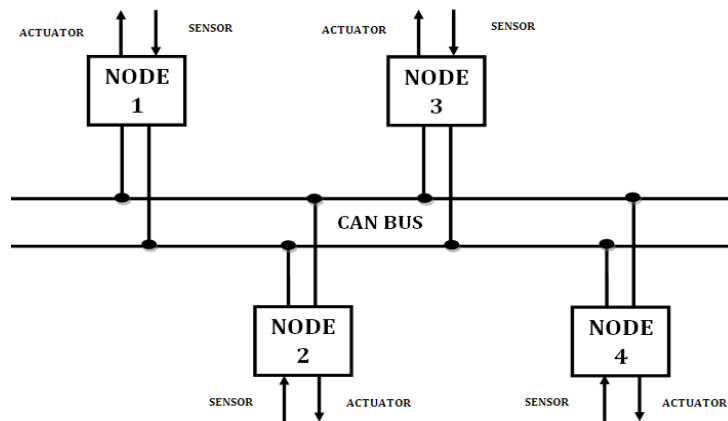


Figure 1: Distributed CAN network

The Physical layer of CAN manages the connection between the nodes in a network and the actual transmission of electrical impulses across a connecting media. It translates message received from the data link layers into an electronic signal. On the same way in the receiving end, physical layer converts the electronic signals back into a message format. Thus it provides standards for bit timing, bit representation and bit synchronization [4].

Data link layer builds messages or packets or data frame to hold data and control information. Control information helps to determine access to bus, identify frames and detect errors. Part of the control information is used to avoid conflicts when two messages try to access the CAN bus at a time, a function is known as Medium Access Control (MAC). The main function of the MAC in CAN protocol is to determine which message having a higher priority, and will get the bus access first.

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 1, Issue 8, October 2013

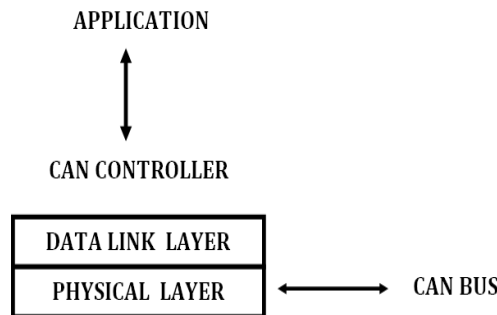


Figure 2: CAN Physical layer

III. WORST CASE RESPONSE TIME TASK SCHEDULING

In automotive applications for short messages communication, the messages sent on CAN network are used to communicate state information, referred to as signals, between different ECUs (electronic circuit unit). Examples of signals that include wheel speeds, oil and water temperature, engine rpm, gear selection, accelerator position, dashboard switch positions, climate control, window control settings, anti-breaking system control(ABS), fault codes, and diagnostic information. In a high-end vehicle there can be more than 2500 distinct signals for each communication node, for each effectively replacing what would have been a separate wire in a traditional point-to-point wiring loom for each node.

More of these signals have real-time constraints associated with them for communication. For example, an ECU reads the position of a switch connected to the brake pedal, engine ignition. This ECU (electronic circuit unit) must send a signal as in form of bit information, carrying information that the brakes have been applied, engine started over the CAN network so that the ECU responsible for the rear light clusters can recognize the change in the value of the signal in the form of bit information and switch on the brake lights. This all must happen within a few tens of milliseconds of the brake pedal being pressed. Engine, transmission and stability control systems typically place even tighter time constraints on signals, which may need to be sent as frequently as once every 5 milliseconds to meet their time constraints [1]. Hence it is essential that CAN messages meet their deadlines.

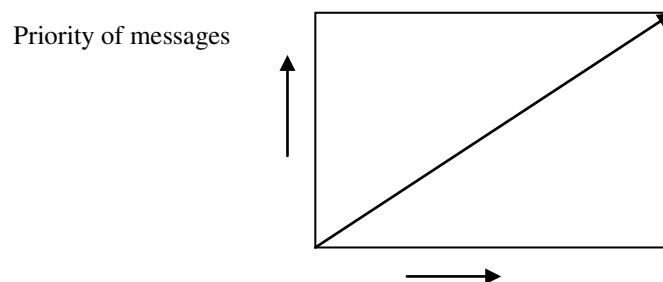


Figure 3: Priority Assignment to messages with Rate in RMA

A. BIT STUFFING IN CAN MESSAGE

CAN protocol standard allow only 5 consecutive bits of same polarity between Start of Frame and the CRC (Cyclic redundancy check) field of a message frame. Bit stuffing is the process of adding one opposite polarity bit whenever a consecutive 5 bits of the same polarity are transmitting. Figure 4 shows a simple example of bit

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 1, Issue 8, October 2013

stuffing.

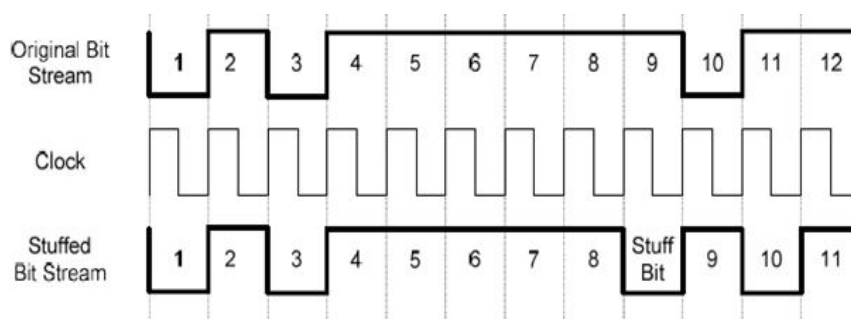


Figure 4: Bit Stuffing

The node which sends bits needs to add the “stuff bit” (opposite polarity bit) before transmission and the receiver eliminate the “stuff bit” to get the original message. The benefit of bit stuffing is that, it provides additional signal edges for data transfer synchronization after 5 bits.

B. BIT ENCODING IN CAN MESSAGES

There are many ways to encode bits in digital systems. We describe two here, Non-Return to Zero and Manchester. Both methods have advantages and disadvantages.



Figure 5: CAN Bit encoding (a) Manchester encoding (b) NRZ encoding

The first one method is the *Manchester encoding scheme* represents the bits as transitions from ‘0’ to ‘1’ or ‘1’ to ‘0’. Even if a frame has a string of ‘1’s or ‘0’s the transition is always necessary. The Manchester encoding scheme is very useful in asynchronous communication systems since there is always a signal transition for bit synchronization is occur. The primary disadvantage of Manchester encoding is that it requires more bandwidth because each bit must have two time slots to be encoded while communication is on processing.

The other one method is *Non-return to zero (NRZ) encoding* does not require signal transitions to represent each bit. The signal remains ‘0’ or ‘1’ for the entire time slot when communication is on process. If a frame has a string of ‘1’s or ‘0’s, the signal will remain constant for as many bit times as necessary. The disadvantage of NRZ is that there is no easy way to tell where each bit starts or ends when there are more than two ‘1’s or ‘0’s in a row. The only way to know where a bit starts or ends is for the receiver to have a clocking source that is identical to the transmitter so that it can decipher the bit stream. This is called synchronous communication. However, it is very difficult to keep two clocking source exactly synchronized for very long at the bit rates used by CAN.

C. ERROR FRAMES IN CAN NETWORKS

An error frame initiates the termination of a damaged data or remote frame. This is actually done by violating the Copyright to IJIRCE

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 1, Issue 8, October 2013

CAN standard. According to CAN standard for maintaining the synchronization there should not be more than 5 consecutive same polarity bit. One extra bit i.e. Minimum 6 consecutive bits of the same polarity bit initiates the error frame. When the error frame is detected the transmitter stop transmitting the current frame and start retransmitting the frame again.

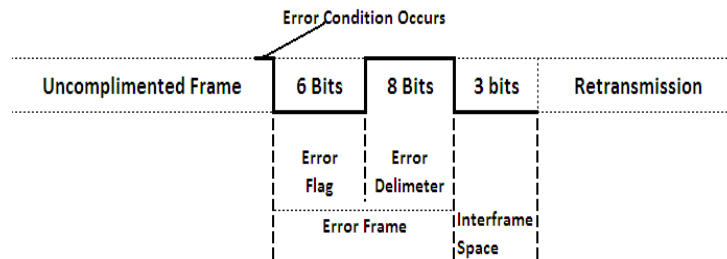


Figure 6: Error Frame Architecture

As shown in the Figure 6 the error frame consists of error flag of 6 sizes bit and error delimiter of size 8 bits. The error flag is to violate CAN standard by transmitting same polarity bit. The bits can be dominant or recessive based on this it is called active error flag (dominant bit) or passive error flag (recessive bit). The error delimiter is represented by a sequence of 8 recessive bits. An error detecting node, after completing the error flag, will send the first recessive bit to the bus. The node will continue sending a recessive level until the bus actually turns recessive.

IV. CONCLUSION

Inside this review paper, the worst case response time analysis of messages in CAN and the probable response time analysis of CAN messages are reviewed. The WCRT analysis indicates the response time analysis presented in early 1990s by Tin dell et al. [2–6] and the WCRT analysis by Davis et al. [1] in 2007. There are many methods present to analyze response time analysis in WCRT, in worst-case analysis it is assumed that every error flag transmitted has a retransmission associated, whereas this is not true (7), since the same error can cause many error flags and only one retransmission. Due to assumption causes the level of pessimism for two method one is worst case bit stuffing method (3) another is probable method (7). One other method is using XOR operation on the messages before transmission (encoding) and redoing the XOR after reception (decoding), thus ignoring having continuous bits of 0s or ones and 1s or 0s, thereby ignoring bit stuffing. In forthcoming the CAN bus will be use in space research for on board computer. In this CAN bus of communication network and the response time analysis for task in worst case analysis response time which is meeting at their deadlines.

ACKNOWLEDGMENT

The correspondence author is thankful to all the faculty members of department of Electronics & Communication, IES College of Technology Bhopal, M.P. and India for continuous support and encouragement. This is an opportunity to prove our technical skill for the benefit of social civilization.

REFERENCES

1. R. I. Davis, A. Burns, R. J. Bril, and J. J. Lukkien, "Controller area network (CAN) schedulability analysis: refuted, revisited and revised," *Real-Time Systems*, vol. 35, no. 3, pp. 239–272, 2007. View at Publisher · View at Google Scholar · View at Scopus
2. K. W. Tin dell, H. Hansson, and A. J. Wellings, "Analyzing real-time communications: controller area network (CAN)," in *Proceedings of the 15th IEEE Real-Time Systems Symposium (RTSS '94)*, pp. 259–263, IEEE Computer Society Press, 1994.
3. T. Nolte, H. Hansson, and C. Norstrom, "Minimizing CAN response-time analysis jitter by message manipulation," in *Proceedings of the 8th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS '02)*, pp. 197–206, September 2002.
4. Daniel Mannisto, Mark Dawson, *An Overview of Controller Area Network (CAN) Technology*, mBus, 2003.
5. *Introduction to the Controller Area Network (CAN) SLOA101A*—August 2002—Revised July 2008
6. K. Tin dell, A. Burns, and A. J. Wellings, "Calculating controller area network (can) message response times," *Control Engineering Practice*, vol. 3, no. 8, pp. 1163–1169, 1995. View at Scopus.
7. P. Axer, M. Sebastian, and R. Ernst, "Probabilistic response time bound for CAN messages with arbitrary deadlines," in *Proceedings of the Design, Automation & Test in Europe Conference & Exhibition (DATE '12)*, pp. 1114–1117, Dresden, Germany, March 2012.



ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 1, Issue 8, October 2013

8. N. Navet, Y. Q. Song, and F. Simonot, "Worst-case deadline failure probability in real-time applications distributed over controller area network," *Journal of Systems Architecture*, vol. 46, no. 7, pp. 607–617, 2000. View at Scopus.
9. I. Broster, A. Burns, and G. Rodriguez-Navas, "Probabilistic analysis of can with faults," in *Proceedings of the 23rd IEEE Real-Time Systems Symposium (RTSS '02)*, pp. 269–278, 2002.
10. Rajib mall "real time operating system" module 2.(Text book)
11. lawrence jenkins,z.c. Alex,gerardine I.mary "Response Time Analysis of Messages in Controller Area Network: A Review" December 2012 hindawi journals article id-148015.2012