DRC Protocol for the Response Time Reduction in CAN based Distributed Embedded System

Hiteshkumar Lad
M.Sc.IT Programme
Veer Narmad South
Gujarat University, Surat, India

Vibhutikumar Joshi
Department of Physics
Veer Narmad South
Gujarat University, Surat, India

ABSTRACT

Controller Area Network operational characteristic supports periodic, sporadic and event based task behavior of distributed embedded system for industrial applications. CAN connect distributed Electronics Control Units (ECU) serially in the network and share measured data of different parameters and control information from the different places. System reliability and redundancy can be affected by changing the number of connected ECUs, Bus length, data rate, etc. In this paper proposed algorithm tries to improve reliability of CAN by changing data rates, according to the length of transmission line for message transmission in the network. Also, Data Rate Change DRC protocol improves temporal behavior of the system with small CPU overheads.

Keywords

Controller Area Network (CAN), Distributed Embedded System (DES), Data rate Change (DRC), jitter, Message Length

1. INTRODUCTION

Controller Area Network (CAN) offers a communication stability and robustness consistency at low cost, which increase its demand in different industrial field applications [1, 4]. As a result, CAN modules are available as "on-chip" component in advance microcontrollers. CAN bus is used to connect Distributed Embedded System modules together and establish network [9]. CAN bus transmits messages in a short bunch of data (up to 8-byte), that is preferred in automation applications to transfer controlling information [3, 4, 14]. As special feature of CAN, a fault confinement mechanism improves the temporal behavior and reliability of the system.

CAN bus is considered under "Class-c network" based on the standards given by Society for Automotive Engineers (SAE) which can transfer information at 125kbps to 1 Mbps data transfer rate. Normally it is observed that Network establishes communication at a fixed data transfer rate to avoid complexity in industrial application. CAN uses multi-master broadcast concept for message transmission. using filter mechanism, the receiver filters out the relevant message only. For CAN, more than one nodes take part in arbitration process. To avoid collision in network CAN apply Carrier Sense Multi Access (CSMA/CA) policy [10, 11]. In this policy, nodes should be able to overwrite a recessive bit, during the arbitration process. This induced the problem of physical length and speed of a CAN bus [2]. The change in bit timing affects on signal propagation in prefixed network length [2]. As per SAE standards, CAN bus length should be 500 meters for communication speed of 125 kbps and 40 meters length for 1Mbps [2]. In this paper, we have improved response time efficiency of the network by reducing physical transmission time for different length of transmission line.

2. INTRODUCTION TO CAN STANDARD 2.0A

S O F	Arbitration field	Control field	Data field	CRC field	ACK field	E O F

Figure 1: Data frame format for CAN

CAN Protocol is uses Non-Return-Zero (NRZ) coding for information transmission [5, 6]. CAN bus communication is asynchronous type transmission, which does not require to transmit clock pulses along with information [5]. CAN-2.0-A Data Frame Specification format shown in Figure 1 where the single bit Start Of Frame (SOF) field provides synchronization between nodes, arbitration field containing 11-bit identifier for frame identification, 6-bit control field containing information about data length and frame is an expansion, 8-byte data field carrying information / Data, 15-bits long error detection field(CRC) use to check received data sequence is same as transmitted data. Acknowledgement field uses to check frame received at receiver and End of Frame (EOF) field [12]. IN CAN at transmitter node 5 identical bits of same polarity found then hardware stuffed one additional bit with opposite polarity and this process called bit stuffing. Total 111 bits are used for 8-byte data transmission without bit stuffing and if one considers worst case scenario with bit stuffing than 135 bits are required to represent message for 8-byte data field [7]. This Message length variation cause by bit stuffing mechanism effects on predictability of CAN network transmission scheduling of real time system results in inefficient operation of data transmission [08].

3. CAN TRADITIONAL SYSTEM MODEL AND ANALYSIS

Tindell et.al in 1994 presented a CAN network Data traffic scheduling analysis considering worst case latencies response time for fixed priority frame [13]. To analyze the system model Tindell et.al considered 11 bit message identifier with fixed priority for static data payload of the set of 1 to 8 byte. The system model explains worst case response time analysis with fixed priority scheduling for non pre-emptive messages. Eq.1 defines worst case response time $\mathrm{R_m}$ in terms of queuing jitter$\mathrm{J_m}$, waiting time $\mathrm{W_m}$ and Physical transmission time $\mathrm{C_m}$ for the mth message.

$$\mathrm{R_m = J_m + W_m + C_m} \qquad \text{(Eq. 1)}$$

Where: $\mathrm{J_m}$ is the queuing jitter which gives the latest queuing time of the message, relative to the beginning of the task.

W_m is the Waiting time (delay) for the transmission introduced by high priority messages or continuity of lower priority message transmission, and

C_m is the transmission time of mth message to transmit on physical bus.

Transmission time is decided based on total message length, including data content, additional stuffed bits, and data transmission speed and frame overheads of the message [13]. In case, due to occurrence of some event, System task activates and put message in queue for the transmission.

The factor affecting queuing delay is given by Eq.2 as below [13]:

$$w_m = B_m + \sum_{\forall j \in hp(m)} \left\lceil \frac{W_m + J_j + T_{bit}}{T_j} \right\rceil C_j \quad (Eq. 2)$$

Where: The hp(m) is the set of messages considered of higher priority than mth message.

T_j is the time require by message j to complete the transmission.

J_j and C_j are queuing jitter and physical transmission time for j th message respectively.

W_m is Waiting time (delay) for the transmission introduced by high priority messages or continuity of lower priority message transmission. s

T_{bit} is the unit bit time for configured speed.

Blocking time delay ($B_{m)}$ is introduced by the time required to complete the pre-initiated transmission of lower priority message k as compare to m message. The B_m time can be expressed by Eq.3[13].

$$B_m = \max_{\forall k \in lp(m)} (C_k) \qquad (Eq. 3)$$

Where: lp(m) is the set of lower priority messages. C_k consider as Physical transmission time for k message.

The Physical transmission time require for the successful transmission of message m is defined as C_m in Eq.4[13].

$$C_m = ((g + 8S_m + 13 + \left\lfloor \frac{g+8S_m-1}{4} \right\rfloor)T_b) \qquad (Eq.4)$$

Where the term S_m consider as number of payload data bytes of message m. g is used for message header value, including arbitration field, control field and CRC value. Term T_b indicates bit time of the bus, $((g + 8S_m - 1)/4)$ term used for stuffed bits, total 47 overhead beats considering message heading and control information. During the transmission, if 5 consecutive bits found with same polarity, then opposite polarity bit is stuffed after 5 consecutive bits in content at transmitter. This stuffed bit is also considers as a part of bit sequence i.e. each stuff bit begins a sequence of 5 bits that itself also subject to bit stuffing. In Eq.4, *g value* is 34 for standard format (11-bit identifiers) or 54 for extended format (29-bit identifiers).

4. INTRODUCTION OF DRC PROTOCOL

Normally, same network data rate is set for long range and short range data communication (i.e. Data transmission rate, configure at 125kbps for network wire of the length of 500 meters and 40 meters as well). CAN supports high data transmission rate of 1Mbps for the short range up to the 40 meters length of transmission wire while 125 Kbps Data transmission rate for both long range and short range communication. If 125 kbps data transmission rate set for long range and 1Mbps data transmission rate set for short range transmission then Network bandwidth can be better utilized.

To avoid complexity in the system model consider DRC protocol message transmission is implemented using Time Division Multiplexing Access (TDMA) strategy. Therefore the possibility priority collision occurrences as well as message blocking are avoided. Here in this paper, a CAN bus communication with different length of transmission line and corresponding algorithm has been developed for the implementation of DRC. If bus length of the transmission line is above 40 meters, the network uses 125 kbps data rate while for the length of transmission line below 40 meters, the network uses 1 Mbps data rate. In order to avoid error in the data transmission, synchronization between every node is mandatory. The DRC algorithm takes care of synchronization between node by sending the control code from transmission node on 125kbbps without any payload. This control code sets 1mbps speed for all the nodes of short distance and "listen only" mode to all long distance nodes. The data are transmitted on 1 Mbps speed for short distance nodes is carried out only once. On the completion of successful transmission, the "listen-only" mode is removed and all short and long distance nodes are set to 125kbps data rate again.

In DRC Due to switching between 1Mbps and 125 Kbps data transmission rate physical Transmission time varies which can be given by Eq. 6 and Eq.7. Response time R_{m_DRC} for worst case condition of DRC protocol can be described by Eq.5.

$$R_{m\_DRC} = J_{m\_DRC} + C_{m\_DRC} \qquad (Eq. 5)$$

Where, $J_{m_}$DRC is queuing jitter time considered for mth message transmission using DRC protocol described in Eq.11.

C_{m_DRC} is Physical transmission time considered for mth message described in Eq.8.

Physical transmission time C_{m_long} for long range communication is expressed by Eq.6 which is similar to the C_m given by Eq.4.

$$C_{m\_long} = ((g + 8S_m + 13 + \left\lfloor \frac{g + 8S_m - 1}{4} \right\rfloor)T_b \quad (Eq. 6)$$

Transmission time C_{m_short} for short range communication can be expressed by Eq.7.

$$C_{m_{short}} = (g + 13 + \left\lfloor \frac{g-1}{4} \right\rfloor)T_{b1} + 2T_{spc\,hange} + (g + 8S_m + 13 + \left\lfloor \frac{g + 8S_m - 1}{4} \right\rfloor)T_{b2} \quad (Eq. 7)$$

$$C_{m\_DRC} = \begin{cases} C_{m\_short}, & \text{distance} < 40 \text{ meters} \\ C_{m\_long}, & \text{distance} \geq 40 \text{ meters} \end{cases} \quad (Eq. 8)$$

In the Eq-7, T_{b1} is bit time for 125kbps speed, T_{b2} is bit time for 1mbps speed and $T_{spchange}$ is a process time required for switching of transmission speed which is negligible. Jittering queue time J_{m_DRC} is expressed by Eq.-11

Jitter queuing for short range message described in Eq.9 that is the augmentation of queuing time require for code message without and with payload respectively.

$$J_{m\_short} = J_{m\_code} + J_{m\_message} \qquad \text{(Eq. 9)}$$

Jitter queuing time for long range message described in Eq.10 is similar to normal network queuing time.

$$J_{m\_long} = J_m \qquad \text{(Eq. 10)}$$

Jitter queuing time for DRC algorithm messages described in Eq.11.

$$J_{m\_DRC} = \begin{cases} J_{m_{short}} & distance < 40\ meters \\ j_{m_{long}} & distance \geq 40 \text{ meters} \end{cases} \quad \text{(Eq. 11)}$$

5. PROPOSED ALGORITHM

Proposed DRC protocol Flow charts for transmitter and receiver nodes are shown in Figure 2 and Figure 3.

5.1 Transmitter Algorithm

Step-1: Collects information for the data transmission.

Step-2: Identify a data transmission range. If it is a short range (under 40 meters) transmission, then step-3 is bypassed and follows the steps-4 directly. If it is a long range (above 40 meters) transmission, then step-3 follows.

Step-3: Transmit data with full load at 125kbps speed and move to first step again.

Step-4: Generate group code in the frame identifier field and transmit frame with payload.

Step-5: Performs transmission speed switching operation and changes the speed to 1Mbps.

Step-6 Transmits the frame with data in payload.

Step-7: After completion of transmission, switching operation performs and sets 125 kbps speed again.

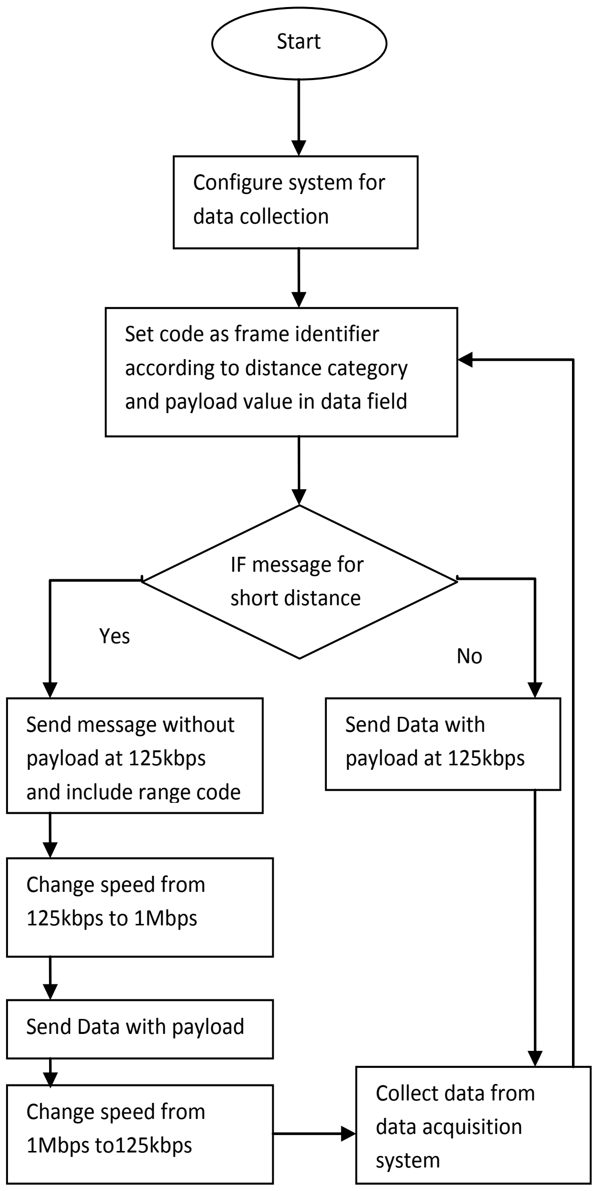
Start
Configure system for data collection
Set code as frame identifier according to distance category and payload value in data field
IF message for short distance
Yes
No
Send message without payload at 125kbps and include range code
Send Data with payload at 125kbps
Change speed from 125kbps to 1Mbps
Send Data with payload
Change speed from 1Mbps to125kbps
Collect data from data acquisition system

Figure 2: DRC algorithm flowchart for Transmitter node

5.2 Receiver Algorithm

Step-1: Extract the identifier field and payload value from received message of control field.

Step-2: If payload value is '0' then flow checks for the matching of the identifier field for the group code. If code matches then the flow bypasses the step-3.

Step-3: If identifier field does not match with speed change group code, then speed remains unchanged.

Step-4: Changes speed to 1mbps and starts receiving message.

Step-5: After completion of successful reception of message the speed again changed to 125kbps.

Step-6: Wait for next message reception.

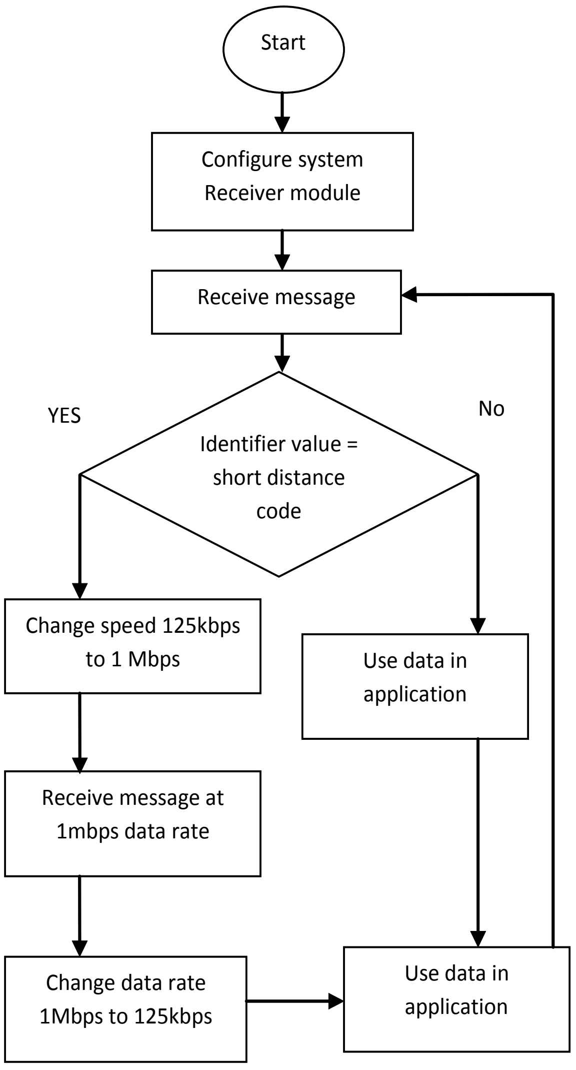
Start
Configure system Receiver module
Receive message
Identifier value = short distance code
YES
No
Change speed 125kbps to 1 Mbps
Use data in application
Receive message at 1mbps data rate
Change data rate 1Mbps to 125kbps
Use data in application

Figure 3: DRC algorithm flowchart for Receiver node

6. EXPERIMENTAL WORK

DRC protocol has been tested on a test bench contained three Freescale Demo board DEMO09S08DZ60 with CAN trans-receiver IC TJA1040 and microcontroller- MCS9S08DZ60. Algorithm is developed and implemented using C-programming language in Freescale Code-worrier tool. Experiments performed on different sized payload and measured its effect on the response time. Out of three one board is used as master node and rest of two boards are used as slaves. The Experimental work successfully established message transmission in CAN with TDMA technique. Normally when message is transmitted using TDMA technique, the queuing delay is consider to be Zero. It has been observed experimentally that the transmitter node and two receiver nodes communicates during predefined allotted timeslot only.

Table 1. Response time for Normal and DRC protocol based CAN network communication

Payload (Byte)	Normal Network Communication Response time (microsecond)		DRC protocol based Network Communication Response time (microsecond)	
	Long range	Short range	Long range	Short range
8-Byte	1555	1555	1555	892
7-Byte	1073	1073	1073	868.63
6-Byte	987.8	987	987	851.88
5-Byte	920.12	920	920	834.3

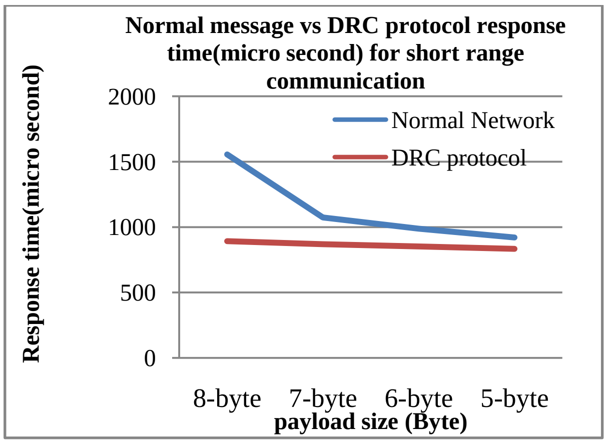

Figure 4: Normal Network message Vs DRC protocol message Response time for Short range Messages

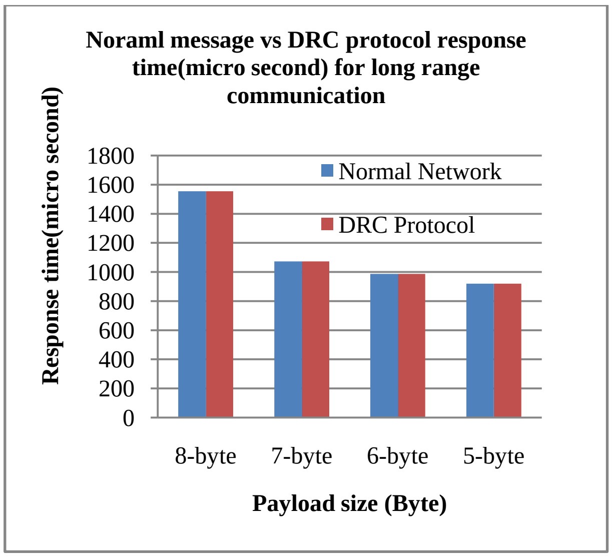

Figure 5: Normal Network message Vs DRC protocol message Response time for Long range Messages

7. CONCLUSION

The results for different payload are presented in the bar graph shown in Figure 5. It can be observed in Figure-5 that for the long distance transmission with and without DRC Protocol, the response time difference is not significant. Figure 4

Indicates reduced response time for short distance transmission. This figure also indicates significant difference in the response time between normal communication and DRC protocol. Further the figure-4 shows the range of time reduction from 600 microseconds to 40 microseconds for payload range of 8byte to 5byte respectively. The system requires a finite switching time which is an additional burden on CPU. However, as compare to the benefit in the overall response time, this additional burden is viable.

The results shown in Table1 are encouraging for the DRC protocol transmission, especially during the short rang message transmission.

8. REFERENCES

[1] G. M. Martinov, A. B. Lyubimov, A. I. Bondarenko, A. E. Sorokoumov, and I. A. Kovalev,"An Approach to Building a Multiprotocol CNC System", ISSN 0005-1179, Automation and Remote Control, 2015, Vol. 76, No. 1, pp. 172–178.

[2] Robert I. Davis, Alan Burns, Reinder J. Bril, Johan J. Lukkien," Controller Area Network (CAN) schedulability analysis: Refuted, revisited and revised", Real-Time System,, April 2007, Volume 35, Issue 3, pp 239-272 (2007) 35:239–272,

[3] Hanxing Chen, Jun Tian, " Research on the Controller Area Network", 2009 International Conference on Networking and Digital Society,, 2009 IEEE, DOI 10.1109/ICNDS.2009.142,978-0-7695-3635-4/09,pp:251-254.

[4] Weiming Tong, Chengde Tong and Yong Liu," A Data Engine for Controller Area Network", 2007 International Conference on Computational Intelligence and Security, 2007, IEEE,DOI 10.1109/CIS.2007.137, 0-7695-3072-9/07, pp: 1015-1019

[5] Mouaaz Nahas , Michael J. Pont, Michael Short," Reducing message-length variations in resource-constrained embedded systems implemented using the Controller Area Network (CAN) protocol", Journal of Systems Architecture 55 (2009), Elsevier, pp. 344–354, doi:10.1016/j.sysarc.2009.03.004, 1383-7621.

[6] Mouaaz Nahas, Michael Short and Michael J. Pont, "The impact of bit stuffing on the real-time performance of a distributed control system", ICC 2005, CAN in Automation.

[7] [M. Nahas, M. Short and M. J. Pont," Using XOR operations to reduce variations in the transmission time of CAN messages: A pilot study", Proceedings of the Second UK Embedded Forum, University of Newcastle upon Tyne, pp. 4-17, ISBN 0-7017-0191-9, November-2005

[8] F. C. Braescu, C. F. Caruntu, L. Ferariu, C. Lazar," OSEK based embedded networked controller handling communication delays", Second Eastern European Regional Conference on the Engineering of Computer Based Systems, pp. 71-77,IEEE, ISBN: 978-0-7695-4418-2/11, 2011.

[9] Vittoria Aiello, Parnian Najafi Borazjani, Ermanno Battista, Massimiliano Albanese, " Next-Generation Technologies for Preventing Accidental Death of Children Trapped in Parked Vehicles",2014 IEEE 15th International Conference on Information Reuse and Integration (IEEE IRI 2014), August 13-15, 2014, 978-1-4799-5880-1/14,IEEE,pp 508-513

[10] Lu´ıs Rodrigues, M´ario Guimar˜aes, Jos´e Rufino, "Fault-tolerant clock synchronization in CAN," 19th IEEE Real-Time Systems Symposium (RTSS'98), ISBN: 0-8186-9212-X, 1998.

[11] K. M. Zuberi, K. G. Shin, " Design and Implementation of Efficient Message Scheduling for Controller Area Network", VOL. 49, NO. 2, IEEE transactions on computers, ISSN: 0018-9340, FEBRUARY 2000.

[12] Thomas Nolte, Hans Hansson, and Christer Norstr¨om," Probabilistic Worst-Case Response-Time Analysis for the Controller Area Network", Proceedings of the 9th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS'03), 2003 IEEE, 1080-1812/03.

[13] K.W. Tindell and A. Burns, "Guaranteeing message latencies on Controller Area Network (CAN)", In Proceedings of 1st International CAN Conference, pp. 1-11, September 1994.

[14] Yang Shunkun, Tang Dongxiao, Shi Xiaohua, " Testing System for CAN Bus-oriented Embedded Software",2014 IEEE/ACIS 13th International Conference on Computer and Information Science (ICIS), 10.1109/ICIS.2014.6912162, 978-1-4799-4860-4/14, IEEE, June 4-6, 2014,379 - 384