Design of Distributed Scheduler on CAN for Real – Time Networking

Hong-Hee Lee, Ui-Hurn Jeong

School of Electrical Engineering,
University of Ulsan, Korea
Tel: +82-52-259-2187
Fax: +82-52-259-1686
Email Address: hhlee@mail.ulsan.ac.kr

Abstract

One of the basic goals, when considering networks for communication in industrial control applications, is the reduction of the complexity of the related wiring harnesses. In addition, the networking offers the advantages for industrial control applications, such as ease of cabling, ease of changes in the cabling, ease of adding controller modules, etc.

CAN(Controller Area Network) is generally applied in car networking in order to reduce the complexity of the related wiring harnesses. These traditional CAN application techniques are modified to achieve the real time communication for the industrial control applications.

In this paper, we design the special purpose scheduler on CAN for Real-Time network system. This scheduler can be used to schedule messages in DCS(Distributed Control System) with CAN for Real-Time network system. And also, the real-time network system is developed and the proposed scheduler is verified experimentally.

Keywords : Distributed Scheduler, Deadline, CAN, Identifier, Real-Time Network

1. Introduction

One of the basic goals, when considering networks for communication in industrial control applications, is the reduction of complexity of related wiring harnesses. In addition, the networking offers the advantages for industrial control applications, such as ease of cabling, ease of changes in the cabling, ease of adding controller modules, etc.

The increasing demand for communication has lead to the specification and existence of various communication protocols. Since the mid-1980s various fieldbus protocols and sensor/actuator protocols have been under design and/or available, such as Profibus, Interbus-S, P-Net, LON, FIP, etc. Since the end of the 1980s the so-called "autobus" protocols have been in their final development or early production phase. CAN protocol, which stands for Controller Area Network, ~~being one of the most~~ advanced autobus protocol in those days, was launched in 1989 as a standard product by Intel. CAN hardware implementations today cover the 2 layers, the physical layer and the data link layer, while the application layer is covered with various software solutions which include DeviceNet, CAN Kingdom, CANOpen. These software solutions for the application layer use the centralized scheduler to satisfy the real time networking. In this case, it is hard to modify the network structure while the data transmission scheduling is convenient.

To increase the system flexibility, we adopt the distributed scheduler. In priority based CSMA/CD protocol system, some message may not occupy network permanently because of its lower priority. It is serious problem in control system. To overcome these shortcomings, we introduce the deadline based priority allocation algorithm to guarantee messages transmission in distributed CAN network system. It is verified that the proposed algorithm improves the system flexibility and the performance of CAN network.

2. CAN Architecture

CAN has basically CSMA/CD + AMP(Carrier Sense Multiple Access/Collision Detection + Arbitation on Message Priority) protocol. Its protocol is similar as IEEE 802.3 CSMA/CD protocol and provides 1, 2 layer in ISO/OSI 7 layers. Because CAN protocol uses NBA(Non-deductive Bit-wise Arbitration) method, the lower priority message can not occupy the transmission line when the higher priority message is transmitted.

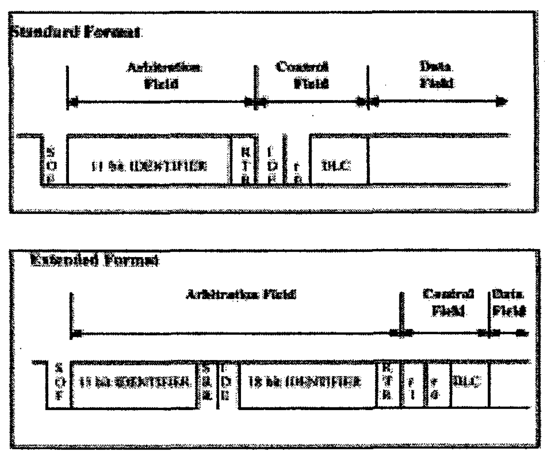

Fig. 1 ~~Data frame CAN 2.0A(11~~ Bit Identifier) and CAN 2.0B(29 Bit Identifier)

CAN data frames are described as Fig.1. The CAN specifications 2.0A and 2.0B are

available. The difference between CAN 2.0A and 2.0B is basically located in the format of the message header, especially the identifier. The CAN 2.0A defines CAN systems with a standard 11 bits identifier(Standard CAN). CAN 2.0B specifies so called extended frames with a 29 bits identifier(Extended CAN). The data format is shown in Fig. 2. The maximum data size in CAN protocol is 8 bytes and the maximum transmission speed is 1 Mbps.

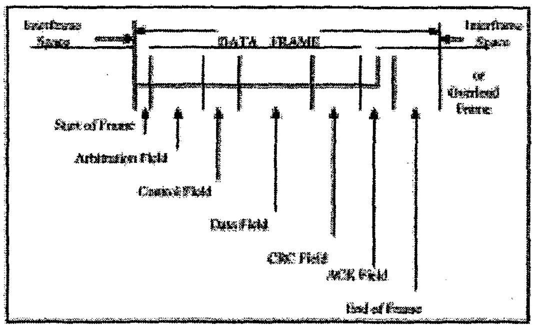

Fig. 2 CAN Data format

3. Identifier Assignment Algorithm

CAN protocol two basic functions which are priority determination and address recognition. The information included in identifier(ID) is used for address recognition by the LLC(Logical Link Control) in CAN data link layer and the MAC(Medium Access Control) which is also in CAN data link layer uses ID as priority determination. In CAN protocol, the priority decision function is dominant factor to network performance.

CAN ID is generally determined in off line condition. This fixed ID allocation method has no problem in case of small size nodes or messages, but in large size system, the lower priority nodes may not be served and this condition derives that the messages can not be transmitted within the specified time(deadline). In this paper, dynamic priority allocation algorithm is proposed to remove this kind serious problem.

3.1 CAN Identifier Field Reconfiguration

As mentioned before, the CAN specifications 2.0A and 2.0B are available, and we used CAN 2.0B, extended frame, in order to make the stable scheduling in large size network system. Fig. 3 describes the extended ID field.

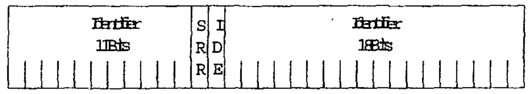

Fig. 3 ID Field of CAN Extended Mode

The maximum transmission bits in CAN extended mode is given as follows:

start bit <1> + identifier bits <11> + SRR bit <1> + IDE bit <1> + identifier bits <18> + Control bits <6> + (maximum) Data bits <64> + CRC bits <15> + (maximum) Stuff bits <23> + CRC delimiter <1> + ACK slot <1> + ACK delimiter <1> + EOF bits <7> + IFS bits <3> = 154bits (Data Length : 1Byte => 88bits)

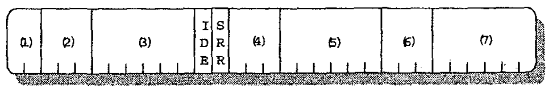

Fig. 4 CAN ID Field Re-Configuration

CAN ID field is reconfigurated to make the stable real time network system which guarantees the message deadline. We assume for CAN ID reconfiguration:

(1)each node has deadline information for generated message

(2)each node can reallocate ID with on-line

(3)each node has the same number preemptive priority queue and the network is non-preemptive.

The reconfigurated ID field is shown in Fig. 4 and the seven separated field means:

(1) 2 bits: degree of quick service

(2) 3 bits: normalized time scale index

(3) 6 bits: normalized deadline

(4) 3 bits: normalized time scale index of 2nd priority message

(5) 6 bits: normalized deadline of 2nd priority message

(6) 3 bits: number of buffed message

(7) 6 bits: ID or address of node(message)

3.2 CAN ID Assignment Algorithm

CAN ID as described in Fig. 4 is assigned using the dynamic priority assignment method, and the assignment procedure is as follows:

(1) Each node assigns the initial value of field (2) and (3) using the known deadline information at starting point.

(2 When a message is waiting because of its lower priority than the current transmitting message, increase the priority step by step not to excess the deadline.

(3) If it is hard to transmit a message within the specified time, increase the priority abruptly with the field (1).

(4) Update the deadline information of the next message with the field (4) and (5) in active node.

(5) Update the number of waiting messages in active node.

The message deadline can be obtained as Eq.(1).

$$D_{mi} = T_{pi} \cdot T_{ri} + r_i \qquad \textbf{(1)}$$

where, i = test message, D_{mi} = deadline of Message i (normalized), T_{pi} = time index of Message i, T_{ri} = time value of Message i, r_i = redundant value.

The schedulability test is done to get the allowable delay time in the proposed ID assignment method, and scheduling constraint(S.C) is given in Eq.(2). In order to guarantee message deadline, S.C must be greater than unity.

$$S.C = \frac{D_{mi} - T_{mi}}{\sum_{\forall j \in hp(i)} T_{mj}} > 1 \qquad (2)$$

where, T_{mi} = transmission time of message i, $\sum T_{mj}$ (priority i < j) = sum of higher priority messages transmission time

Fig. 5 shows the flow chart of ID assignment algorithm.

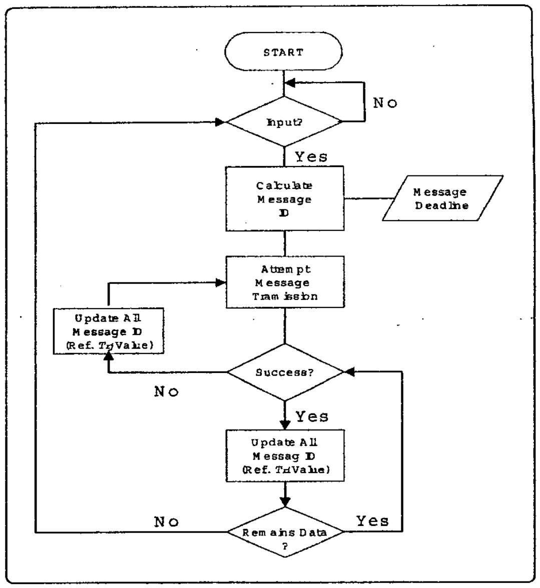

Fig.5 Flow chart of ID Assignment Algorithm

4. Experimental Results

The proposed ID assignment algorithm is experimented with simplified CAN network shown in Fig. 6. CAN network is stand-alone type, and one is used as traffic generator for simple hardware equipment. PC is used to monitor the network conditions.

Table 1 shows message type in each node. In case of message pattern shown in table 1, 3 messages can not transmitted within deadline as a result of schedulability test(Eq.(3)).

Another message pattern is given in table 2-1(periodic message) and table 2-2(sporadic message). These messages are input to the CAN node simultaneously, message 2 excesses the deadline in case of fixed priority assignment method. But, this problem can be removed using our ID assignment algorithm.

$$\sum_{k=1}^{n} \left\lceil \frac{P_{mk}}{P_{\max}} \right\rceil \times T_{\max} \times r_k < P_{\max} \qquad (3)$$

where, P_{mk}:Time period of message k, P_{max}: Time period of Input Message, T_{max}: Max. Message Transmission Time, r_k: Number of message k

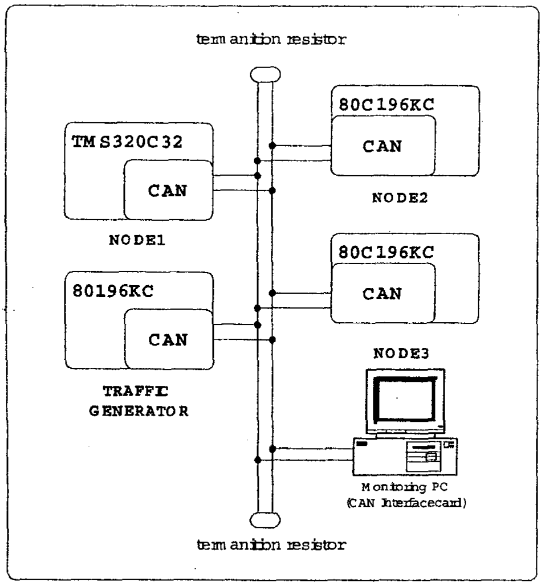

Fig.6 CAN Network

Table 1 Input Messages(Periodic)

Periodic Messages	Node1		Node2		Node3	
	TP/D	DL	TP/D	DL	TP/D	DL
Message1	3/1.2	8	2.5/1	8	3/1.2	8
Message2	2/0.8	8	5/2	8	3.75/ 1.5	8
Message3	2.5/1	8	3.75/ 1.5	8	5/2	8
Message4	5/2	8	5/2	8	2.5/1	8
Message5	1/0.4	8	2/0.8	8	1.25/ 0.5	8

D : Message Deadline (ms)

DL: Data Length(Bytes)

TP : Time period of Input Message (ms)

Table 2-1 Input Messages (Periodic)

Periodic messages	Node1		Node2		Node3	
	TP /D	DL	TP /D	DL	TP /D	DL
Message 1	3/1.2	8	2.5/1	8	3/1.2	8
Message 2	2/0.8	8	5/2	8	3.75/ 1.5	8
Message 3	2.5/1	8	3.75/ 1.5	8	5/2	8

Table 2-2 Input Messages (Sporadic)

Periodic messages	Node1		Node2		Node3	
	TP /D	DL	TP /D	DL	TP /D	DL
Message 1	-/0.5	8	-/0.5	8	-/0.5	8
Message 2	/1	8	-	8	-	8

5. Conclusions

When we adopt the centralized scheduler in distributed control system, it is hard to modify the network structure while the data transmission scheduling is convenient. The distributed scheduler is applied to CAN network to increase system flexibility, and the design algorithm of the distributed scheduler for CAN network is developed to achieve the real time data transmission. It is verified that the proposed algorithm improves the system flexibility and the performance of CAN network.

6. References

[1]Wolfhard Lawrenz, "CAN System Engineering from Theory to Practical Application", *Springer*, 1997.

[2]G. Cena and A. Valenzano, "An Improved CAN Fieldbus for Industrial Application", *IEEE Trans. On Industrial Electronics*, Vol.44, No.4, Aug. 1997.

[3]K M. Zuberi and Kang G. Shin "Scheduling Messages on Controller Area Network for Real-Time CIM Application", IEEE Trans. On Robotics and Automation, Vol.13, No.2, pp. 310 – 314, April 1997

[4]CiA, CAN Spoecification 2.0 PartA, B

[5]CiA, CAN Application Layer, *www.can-cia.de*, 1997

[6]CiA, CAN Datalink Layer, *www.can-cia.de*, 1997

[7]CiA, CAN Physical Layer, *www.can-cia.de*, 1997

[8]Philips, "SJA1000 Stand-alone CAN Controller", Application Note AN97076 & Specification *Philips Semiconductor*, 1997 Nov.

[9]W. H. Kim and S. H. Hong, "A Study on the Implementation of CAN in Distributed Control System of Power Plant", Tran. KIEE, Vol. 48A, No. 6, pp.760-772, Jun. 1999

7. Acknowledgement

A special word of thanks to the Institute of Information Technology Assessment(IITA) for financial support.