Efficient CAN-based network for marine engine state monitoring system

Junseok Lee, Yoseop Hwang, Jaehan Jo and Jangmyung Lee

Dept. of Electronics Engineering, Pusan National University, Korea
(Tel : +82-51-510-1696; Fax : +82-51-510-1693)
(sukidda@pusan.ac.kr; mmx001@pusan.ac.kr; jjhan@pusan.ac.kr; jmlee@pusan.ac.kr)

Abstract: This paper presents the marine engine state monitoring system implemented with Controller Area Network. As the marine engine state monitoring system requires various kind of engine information, it consists of a lot of sensor nodes. So, with the features of Controller Area Network which supports huge numbers of message ID, and its message arbitration ability, this paper presents scheduling method for the performance of monitoring system. And its effectiveness and validity have been shown through experiments.

Keywords: CAN(Controller Area Network), Engine State Monitoring System, DPQ(Distributed Precedence Queue)

I. INTRODUCTIION

Marine Engine State Monitoring System purposes precise and rapid transmission of engine condition data from various measurement instruments to the engine control room. A failure of engine in sailing ship affects critically to the operating the ship, and may causes enormous financial loss. So, real-time monitoring system for preventing the unpredictable failure and maintaining support of marine engine is needed.

A major advantage in using the CAN technology with respect to the other kinds of field-buses available is the wide choice of very cheap electronic components. The Engine State Monitoring System requires lots of sensor nodes and needs cheaper field-bus solution. Additionally, CAN can simplify the sensor node wiring and prioritize the message transmissions which satisfy an urgent message transmission needed at industrial environment.[1],[2]

But if network is overloaded, data transmission is decreased rapidly due to increasing data transmission collision. If this state is continuous, a network may be groggy and non-transmission condition for a long time. This paper presents a mechanism that can be a fair transmission and can reduce delay time with assignment precedence queue to delayed sensor node.

II. Engine State Monitoring System

State Monitoring can be started with measuring of each part of engine (Fig. 2). Engine State Monitoring System can be divided into three parts. Status measuring sensors which measures temperature, pressure, perception of flowing and various status indication factors are equipped with the CAN Module. And CAN Module consist of microprocessor and CAN transceiver and transmits data from sensors to the SPU, respectively. And SPU(Signal Processing Unit) collects entire network data and transmits engine condition signal to the ECR(Engine Control Room) via RS-485.

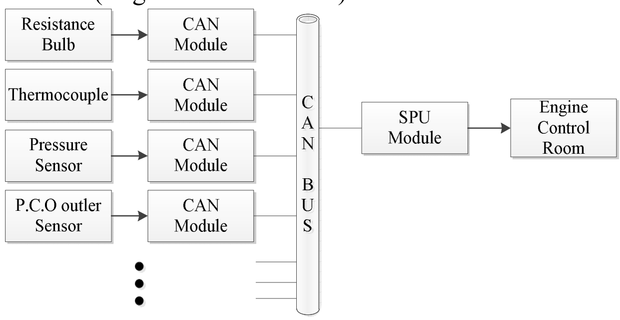

Fig. 1. Engine State Monitoring System Architecture

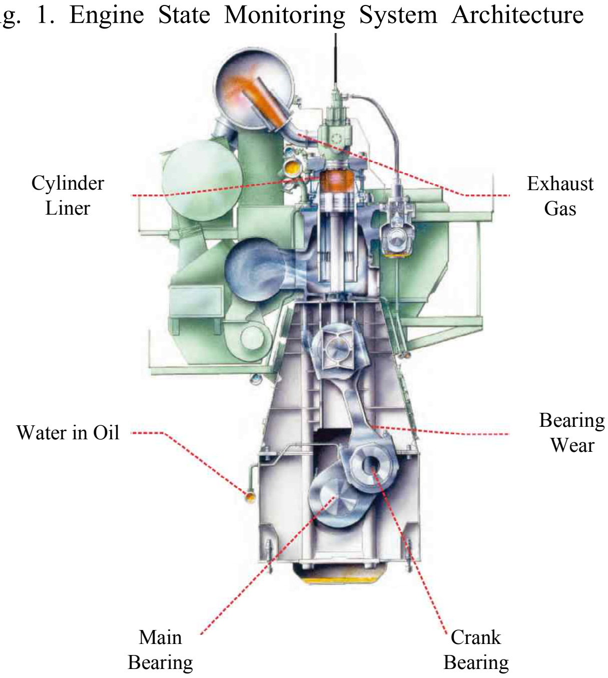

Fig. 2. Marine Engine

III. Controller Area Network Protocol

CAN is based on a CSMA/CS channel access technique, modified to enforce a deterministic resolution of collisions on a network which uses a priority scheme based on the identifiers of exchanged objects. The CAN protocol adopts a layered architecture which is based on the OSI reference model, even though it is not fully OSI compliant. As with other networks conceived for the factory automation environment, it relies on a reduced protocol stack, consisting only of the Physical layer, Data link layer, and Application layer. This paper is focused on the data link layer. In fact most of the features of CAN which concern topics such as the sharing of the bandwidth among the different stations and the access delays they experience depend on the mechanism adopted at this level.[3]

SOF	Arbitration Field	Control Field	Data Field	CRC Field	ACK	EOF
1 bit	11 bit or 29 bit	4 bit	0~8 bytes	15 bit	2 bit	7 bit

Fig. 3. CAN Data Frame format

IV. Distributed Precedence Queue (DPQ)

The CAN implicitly assigns to each object exchanged in the network a priority that corresponds to the identifier of the object itself. Even though this mechanism enforces a deterministic arbitration that is able to resolve any conflict that occurs when several nodes start transmitting at the same time, it is clearly unfair. If many nodes are connected in the network, nodes that are of low priority rank can continuously lose a transmission opportunity. That is, if high priority objects transmit continuously, finally a low priority object can miss an important message which is relatively unimportant compared to that of a high priority object.

Accordingly, a mechanism that uses a relative priority according to the consideration of low priority nodes is necessary although the CAN implicitly assigns a priority. Fair behavior, which for example enforces a round-robin policy among different stations, has to be guaranteed to all the objects exchanged at a given priority level.

In this paper, it is shown that this kind of behavior can be obtained by slightly modifying the frame acceptance filtering function of the LLC sub-layer. In particular, only the significance of the identifier field in the transmitted frame has to be modified in some way. The resulting arbitration mechanism is able to enforce a round-robin policy among the stations that want to transmit a message on the bus, and provides two levels of priority for the frame transmission services. Little or nothing has to be changed at the MAC level; and in this way it is possible to reuse the same electronics components developed for the implementation of the standard CAN protocol.

1. DPQ basic principle

The basic idea of this CAN fairness control mechanism that is to insert into a global queue all of the nodes that want to transmit over the shared medium. For Node C, of which transmission is continuously delayed, a queue is created to transmit Node C and the other nodes that transmit with C. So, several queues can be partially made in this research, two queue were used.

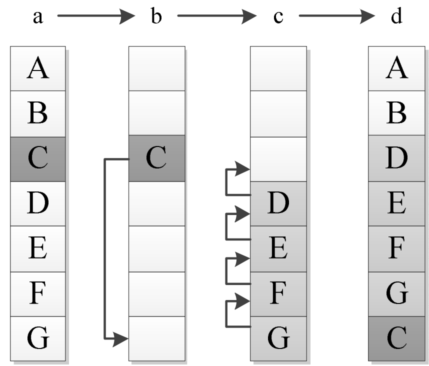

Fig. 4. Generation of a precedence queue in DPQ

This distributed precedence queue protocol provides the opportunity to create precedence queues for all nodes in a network. And, in the case that several precedence queues exist, each precedence queue assigned a priority so that they can be implemented independently.

The DPQ mode ID, which is stored in the 11 bit standard ID field, indicates the precedence queue order of each node. Whenever a node carries out a transmission, it moves to the end of the queue, thus lowering its precedence to the minimum. All of the nodes following the transmitting node advance by one position in the queue, occupying the space that has just been created. Using this round-robin policy, collisions among messages are avoided.

The queue is not stored in some specific location. Instead, it is distributed among all the nodes in the network. Each node is responsible for storing and updating. That is, if the maximum permission delay time is reached, it creates a precedence queue, and then it has to dynamically change priorities to transmit preferentially with other nodes. And a precedence queue has to be dissolved when is completed an urgent task.[4]

DLC	Queue Entry (1 byte)	Precedence Priority (7 byte)	CRC
	8 byte (data field)		

Fig. 5. Structure of a data field for DPQ

We suppose a network that is composed of Nodes A to G as shown in Fig 4. If Node C builds up a queue, the ID that is entered into the data frame queue can transmit and designate to 7 by lower 7 byte. At this time, it will be designated precedence priority to higher byte. Then, each node filters to enter itself into the queue, and it assigns its queue. After Node C transmits a message, it will go to the last position in the queue. And the other nodes will move up one position by order. And the remaining nodes that to be transmitted are designated using the upper 1 byte as shown in Fig 5; their queues will be dissolved or maintained using the upper 1 byte, as shown the Fig 5 after all transmissions are completed.

2. DPQ Implementation Issues

The DPQ mechanism can be implemented without any modifications to the basic format of CAN frames. It uses an identifier field to designate the priority queue. Because the length of the conventional identifier field defined in the CAN standard is too small, the CAN extended format can be adopted.

SOF	11 bits Base ID	SSR	IDE	18 bits Ext. ID	RTR	r0	r1	DLC	...

Fig. 6. Format of the header of extended CAN Frame

The DPQ uses the first 11 bits of the identifier field for its control information, whereas the remaining lower order 18 bits (ID ext.) are used to dynamically store the effective identifier of the an exchanged object (EID).

SOF	t0	t1	P	8 bit PL	SSR	IDE	18 bits Ext. ID	RTR	r0	r1	DLC	...

Fig. 7. Format of the header DPQ Frame

The first two bits (t0, t1) must be set at the logical value of zero as shown in Fig 7. Then, the protocol is divided by a standard CAN communication and DPQ mechanism. So, DPQ always has a higher priority than a CAN mechanism, and they can exist in this same space.

The priority bit P specifies whether the frame has to be transmitted as a high priority frame (P=0) or as a low priority frame (P=1). When t1 and P are used, the priority can be assigned a maximum 4 queues.

The next 8 bits represent the precedence level of the frame. Namely, these 8 bits show the transmission queue order. The DPQ, which was used in this research, uses t0, t1, and then distinguishes the standard CAN mechanism, and sets each queue using P, and concludes the precedence in the queue using 8 bits.

V. Experiment and Result

In this paper, we experimented via Engine State Monitoring System which is composed of ten CAN Modules and one SPU to evaluate the effectiveness of the proposed method.

CAN Module uses sensors (Resistance Bulb, Thermocouple, Pressure Sensor...) actually used in marine engine. And it uses data from measuring elements composed as shown in Fig.9. And CAN Module use Microchip's dsPIC30F4012 and SPU uses TI's TMS320F28335.

Fig. 9 shows the monitoring program for PC via LabVIEW to monitor the state of network and data in real-time. Each of vertical fill slides indicate occupied bandwidth of node and gauge indicates shows temperature of measuring element. We set the bitrate of network at 500Kbps considering the length of network installed in the low-speed marine engine.

To verify the effectiveness of DPQ mechanism, we presented transmission result comparing normal CAN transmission with transmission in DPQ mode. For actual monitoring system for the Low-speed engine in large vessel, they use about hundreds of sensor; we set the transmission period at 500us for more possible transmission collisions.

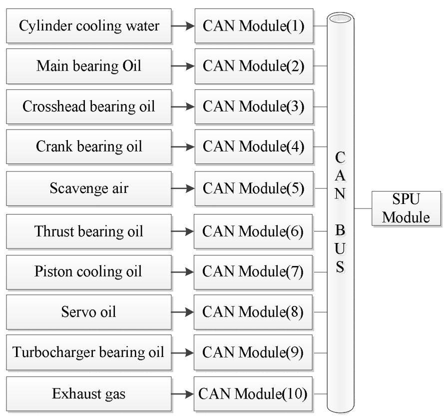

Fig. 8. Experiment System Architecture

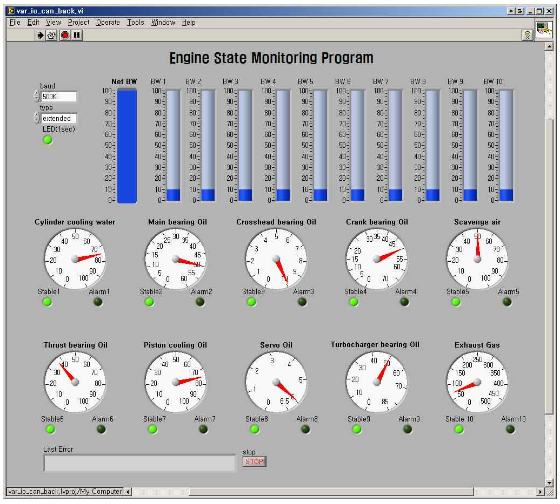

Fig. 9 Engine State Monitoring Program

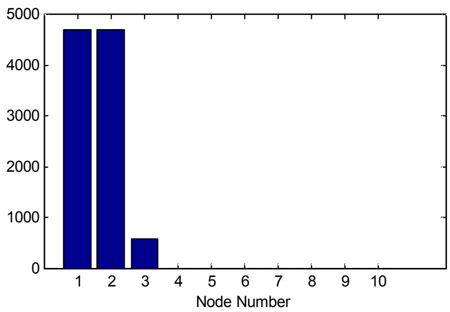

Fig. 10 Received Messages in normal CAN transmission

Fig. 10 shows received message numbers for each sensor nodes when network received 10000 messages. Because of consecutive collision and short transmission period, messages with low-priority are not transmitted while messages with high-priority occupy most of bandwidth.

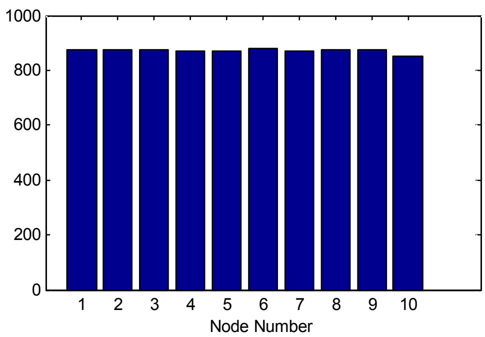

Fig. 11. Received Messages in DPQ mode

Fig. 11 shows result of transmission of DPQ mode in identical condition. It shows evenly distributed received messages.

But data frames derived from the process of making and dissolving the precedence queue, actual state monitoring messages are decreased about 10 percent.

VI. Conclusion

In this paper, we proposed the Marine Engine Sate Monitoring System using DPQ mechanism to improve fairness in network.

Through these methods, messages with low-priority were ensured to transmit message and delay its time was improved. Features of the Marine engine monitoring system are as follows.

1. CAN communication module is connected to sensor used in actual Marine, is distinguished according to set ID and is designed for adding extra modules easily.
2. SPU (Signal Processing Unit) module for processing total data which transmit in CAN module is able to check Whole engine, bearing state and etc.
3. Whole states of Marine engine is able to be checked through the output values (voltage, current, resistor) of all sensors using the SPU (Signal Processing Unit) module.
4. Conventional communication method as RS232 and RS485 is supported for compatibility with the existing Marine engine system. Therefore, state of Marine is able to be checked in ECR (Engine Control Room).

ACKNOWLEDGEMENT

This research was financially supported by the Ministry of Knowledge Economy (MKE) and Korea Institute for Advancement of Technology (KIAT) through the Research and Development of Regional Industry.

This research was supported by the MKE (The Ministry of Knowledge Economy), Korea, under the Human resource development program for robotics support program supervised by the NIPA (National IT Industry Promotion Agency). (NIPA-2010-(C7000-1001-0009))

REFERENCES

[1] Bosch, "CAN Specification Version 2.0.", Robert Bosch GmbH. Stuttgard, 1991.

[2] International Standard Organization, "Road - vehicles Interchange of digital information - Controller area network for high-speed communication", ISO 11898, November 1993.

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

[4] S. Corriganm, Intriduction to the Controller Area Network, TI, Application Report. Texas Instruments, 2002.