A Distributed Precedence Queue Mechanism to Assign Efficient Bandwidth in CAN Networks

Ho-Seek Choi, Jang-Myung Lee

Department of Electronics Engineering, Pusan National University
San 30 Jangjeon-dong Kumjeong-ku, Busan, 609-735, Korea
chs_chs@hanmail.net, jmlee@pusan.ac.kr

Abstract

This paper presents a distributed precedence queue mechanism to resolve unexpected transmission delay of a lower priority transaction in a CAN based system, which keeps a fixed priority in data transactions. The mechanism is implemented in the upper sub-layer of the data link layer (DLL), which is fully compatible with the original medium access control layer protocol of CAN. Thus the mechanism can be implemented dynamically while the data transactions are going on without any hardware modification. The CAN protocol was originally developed to be used in the automotive industry and it was recently applied for a broader class of automated factories. Even though CAN is able to satisfy most of real-time requirements found in automated environments, it is not to enforce either a fair subdivision of the network bandwidth among the stations or a satisfactory distribution of the access delays in message transmissions. The proposed solution provides a superset of the CAN logical link layer control, which can coexist with the older CAN applications. Through the real experiments, effectiveness of the proposed mechanism is verified.

1. Introduction

The controller area network protocol (CAN) was developed to solve complex cable problems and reliability reduction in automotive [1,2]. This availability was built network of high reliability applied various industry environment [3,4,5].

Unlike the IEEE 802.3 standard-access-technique-based CSMA/CD protocol[6], CAN's medium-access control mechanism ensures that when collision occurs a non-destructive contention-based arbitration is initiated that stops all of the transmitting stations except the one which is sending the frame having the highest priority. The frames that are transmitted are not addressed to a specific destination, but they are considered as global objects, each of which is associated with a network-wide unique identifier. CAN allocates absolutely priority to messages or objects transmitted in a network using ID. This mechanism is a good method to manage collisions in network.

If a network is overloaded, the data transmission quantity is rapidly decreased to increase data transmission collision. If this state is continuous, a network may be groggy and the state of non-transmission may continue over a long period of time. This paper presents a mechanism that can create a fair transmission chance and can reduce delay time [7,8,9] using a distributed precedence queue, and assigning a precedence queue to relatively low priority and objects having similar transmission purposes when a network is overloaded, and which can compensate a maximum tolerance delay time and to remove ineffectiveness for an identifier assigned statically into an overload condition [10].

This precedence queue is not assigned statically but assigned dynamically According to transmission quantity, so that the transmission efficiency can be optimized in network. And each queue can independently assign transmission sequences of data of a relative priority.

So, this paper can contribute to the mechanism that can transmit data within a constant time to adjust its priority dynamically based on an extended CAN protocol when a low priority object delays transmission because of an overload in a network.

Identifier is assigned statically in the CAN protocol, the two requirements of a fair transmission chance and delay time, can not be satisfied because this solve collision problem by a static identifier. In this paper it is shown that the problem can be solved collision by the filtering of input frames according to the identifier of each object and by redefining the identifier in the identifier field. By redefinition of the distributed precedence queue (DPQ) to use the identifier field of the extended CAN, each object can be transmitted according to a fair transmission sequence and can thus satisfy the maximum tolerance delay time.

2. CAN Analysis

2.1 A basic CAN protocol

The CAN is based on a CSMA/CD channel access technique. It uses a priority modification mechanism for transmitted-received messages to resolve collisions in a network. The CAN protocol adopts a layered architecture that is based on the OSI reference model, even though it is not fully OSI compliant, and the architecture is composed of three layers the factory automation environment.

1. The Application Layer
   ; Support to access on a Network
2. The Data Link Layer
   ; Connection physical address to the upper-low layer
3. The Physical Layer
   ; Transmission bit stream to physical medium

This paper resolves the transmission delay time problem using the data link layer and the only LLC sub-layer between the MAC (Medium Access Control) and the LLC (Logical Link Control) of the data link layer.

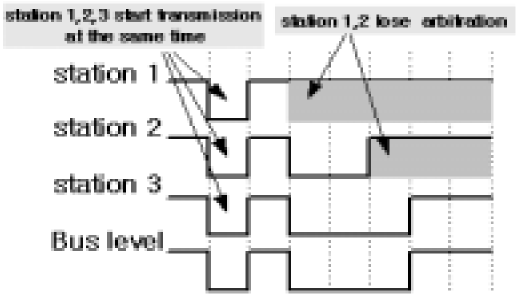

Fig 1. CAN arbitration phase

3. A Distributed Precedence Queue Mechanism (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.

3.1 DPQ 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 as shown in Fig 1, 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.

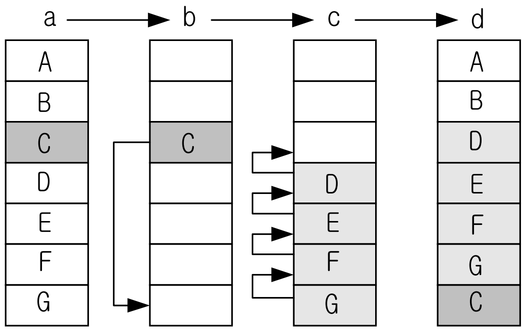

Fig 2. Generation of a precedence queue in DPQ mechanism

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 shown in the Fig 4, 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.

We suppose a network that is composed of Nodes A to G as shown in Fig 2. 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 3; their queues will be dissolved or maintained using the upper 1 byte, as shown the Fig 3 after all transmissions are completed.

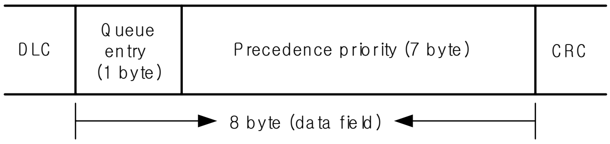

Fig 3. Structure of a data field for DPQ

3.2 DPQ Realization Method

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	11bits base ID	SSR	IDE	18 bit ID EXT	RTR	r0	r1	DLC

Fig 4. Format of the header of extended CAN frames

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 bit EID	RTR	r0	r1	DLC

Fig 5. Format of the header of DPQ frames

The first two bits (t0, t1) must be set at the logical value of zero as shown in Fig 5. 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.

4. System Architecture and Experiments

To verify the usefulness of the mechanism presented in this paper, actuator ECU that are used in throttle-body controllers of vehicles and portable inspection equipment ECU that can set sensor limit values and can diagnosis vehicle problems, established the basic nodes.

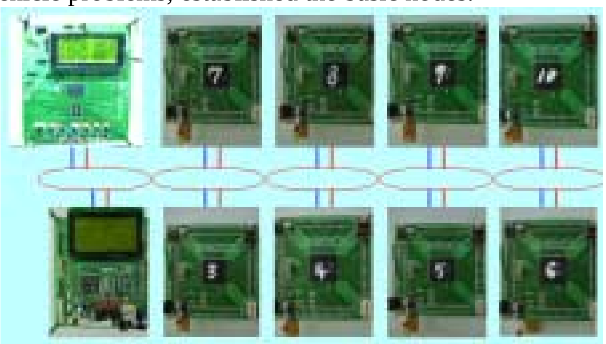

Fig 6. Total system organization

The total system consisted of additional virtual ECU of 10 nodes used in many parts of the vehicles as inhalation fuel ECU, lighting ECU, side-mirror ECU, and exhaust port ECU.

Each node used TMS320LF2407 with the CAN module and PCA82C251 with the CAN transceiver. Each node was set to a 250 Kbps transmission time.

The transmission period for the total 10 nodes was set to two states, 10ms and 2ms. When the transmission period was 10 ms, Collisions did not often occur. But when it was 2 ms, collisions often occurred. The transmission message priority was arranged as Node 1 (portable ECU) and node 2 (main ECU) for each transmission period and this priority decreased gradually. When the transmission period was 2 ms, Node 8,9,10 suffered a long transmission delay because of message collision on the bus, and the DPQ mode was applied to resolve this problem at Node 8,9,10.

Table 1. Identification Definition (ID)

node	standard CAN	DPQ
PORTABLE	11 1 0000 0001	11 1 0000 0001
MAIN ECU	11 1 0000 0010	11 1 0000 0010
3	11 1 0000 0011	11 1 0000 0011
4	11 1 0000 0100	11 1 0000 0100
5	11 1 0000 0101	11 1 0000 0101
6	11 1 0000 0110	11 1 0000 0110
7	11 1 0000 0111	11 1 0000 0111
8	11 1 0000 1000	00 1 1111 1101
9	11 1 0000 1001	00 1 1111 1110
10	11 1 0000 1010	00 1 1111 1111

5. Result and Analysis

Fig 7 shows the transmission delay time of the Node 1. From 1 to 50, the X axis values show the transmission delay time when the transmission period was 10 ms. And from 51 to 100, the values show the transmission delay time when the transmission period was 2 ms. And from 101 to 150, the values show the transmission delay time when the DPQ mode was applied.

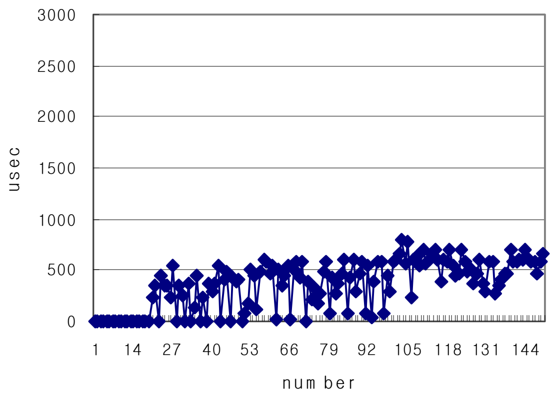

Fig 7. Transmission delay time of node 1

From the Fig 7 results, we know that Node 1 increased the delay time more when the transmission period was 2 ms than when the transmission period was 10 ms. And additional delay time occurred for Node 8, 9 and 10 in DPQ mode.

As shown in Fig 8, in the case of Node 8, the state which a transmission period is 2ms, a longer delay time occurred for low priority nodes than other nodes. To overcome this problem, we can verify that a transmission chance was guaranteed and the delay time was advanced outstandingly, when the DPQ mode was applied instead of changing the priority permanently, as shown in Fig 9.

In case of experiment 2 shown in the Fig 11, the graph shows a transmission delay time. From 1 to 50, the X axis values show a transmission delay time for the highest priority Node 1 when the transmission period was 2 ms. And from 51 to 100, the values are shown for that when the node number was 10. From 101 to 150, the values are shown for when the DPQ mode was applied.

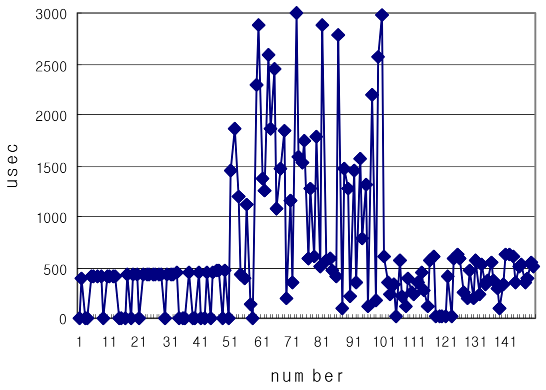

Fig 8. Transmission delay time of node 8

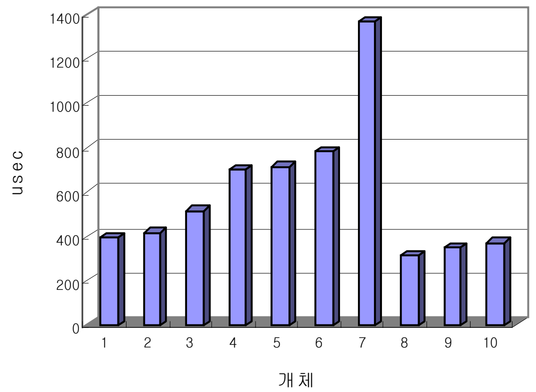

Fig 9. Average transmission delay time of DPQ mode

6. Conclusion

This study applied the DPQ mechanism to correct the ineffectiveness occurring according to a fixed priority mechanism and to arbitrate collisions in a network using a standard CAN protocol. The proposed mechanism established the availability through an experiment of two different states.

The experiment showed that a transmission of a low priority node does not exceed the maximum tolerance delay time using the DPQ mode, despite frequently occurring collisions in transmission and the rapid transmission of each node.

But, in the case of the DPQ mode being applied to high priority object, the effectiveness was lower than that of a standard CAN application. In future research, algorithms will be developed to efficiently manage the time delay of each object, applying the DPQ mechanism dynamically. And it will be shown how these algorithms can be applied conveniently for compatibility with other CAN applications.

Acknowledgments

This work was supported (in part) by the Ministry of Science & Technology, Korea, under the Region Research Center University (RRCU).

References

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

[2] International Standard Organization, “Road - vehicles Interchange of digital information - Controller area network for high-speed communication” Draft Amendment, *ISO 11898*:1993/DAM 1, February, 1994.

[3] CAN in AUTOMATION International Users and Manufacturers Group e. V. “CAN Application Layer (CAL)”, CiA/DS201-CiA/DS205, CiA/DS207.

[4] Jin W. Park, Dong K. No, Jae H. Park, Hwa R. Hur, Jang M. Lee, “Implementation of a Mobile Robot with Distributed Control Structure using CAN” Autumn Combination Conference . pp 251-255.1999.

[5] Sung S. Hong, “A distributed real time control systems,” CASE Technical Special: Real-Time Control System (3), *ICASE*, 1, 1998.

[6] IEEE Standards for Local Area Networks, “Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications”, *ANSI/IEEE Std 802.3*- ISO/DIS 8802/3, 1985.

[7] K. Tindell and A. Burns and A. Wellings, “Calculating Controller Area Network (CAN) Message Response Times”, *in Proc. 1994 IFAC Workshop on Distributed Computer Control Systems*, Toledo, Spain, September, 1994.

[8] K. Tindell and A. Burns, “Guaranteeing Message Latencies on Control Area Network (CAN)”, *in Proc. 1 st International CAN Conference*, Mainz, Germany, September, 1994.

[9] K. Tendell, A. Burns and A. Wellings, “Analysis of Hard Real-Time Communications”, *Report YCS 222, Department of Computer Science*, University of York , to appear in Real-Time Systems, 1994.

[10] “SDS-Smart distributed system specification” *Hineysell Inc.*, Micro Switch Division, Phoenix, AZ, GS 052-103/104/1-5/106/107/108.