2nd International Conference on Electrical Systems Design & Technologies, Hammamet Tunisia, Nov. 8-10, 2008

NEW TECHNIQUE TO ENFORCE A FAIR BEHAVIOR IN ACCESSING THE MEDIUM IN THE CONTROLLER AREA NETWORK

Tamer A Beitelmal	Ahmed M. El-Mahdi	Hassan S. Abdulkarim
GECOL - Libya	Garyounis University	Garyounis University
	Faculty of Engineering Department of Electrical Engineering	Faculty of Engineering Department of Electrical Engineering
Tmr7813@hotmail.com	ahmedelmahdi@garyounis.edu	Hsa333lby@yahoo.com

***Abstract*--Controller Area Network (CAN) is a serial network technology that was originally designed for the automotive industry. Fundamentally, CAN is a type of Local Area Network (LAN), which is based on the collision-detection method similar to Ethernet. However, in Ethernet collision-detection forces conflicting message senders to stop and resubmit their messages after a random interval. In CAN collision-detection, the message senders go into a non-destructive arbitration process. CAN data transmissions are distinguished by a unique message identifier, which also represents the message priority. Since each application must have its own unique identifier, it is not possible to make the network consider two or more applications as equally urgent. In the high traffic condition, network messages with low relative priorities messages may be delayed by any higher-priority message, even it could be delayed more than once by the same higher priority message. In this paper we propose a modified method (*Taking Multiple Requests* TMR method) that has the ability to enforce a fair behavior in accessing the bus with respect to original CAN. A comparison with the original CAN is illustrated and the results of simulations are shown.**

***Index Terms*— CAN, MAC, Embedded System, Control Networks.**

1. INTRODUCTION

Recently, CAN has emerged from its home environment "vehicle" to a broader class of applications in various automated factory environments. In the low traffic and for the systems where the messages are clearly organized in a priority sequence “like in the cars", the CAN access protocol is working properly by addressing the priority depending on the importance. But when it is used in a control network where some messages need the same importance or where there is no explicit priority classification, CAN access protocol fails to satisfy the timing conditions for the low priority messages[1]. This paper proposes a method called Taking Multiple Request (TMR) method, where a group of messages which request access of the bus at the same time are considered as a group. Each message in the group is allowed to access the network bus according to its priority for one time only till all messages, within the group, are processed. The proposed TMR method applies bus access priority without the unfair behavior exhibited by the formal CAN protocol under heavy traffic.

The remainder of the paper is organized as follows. In section 2, we make a brief description of the arbitration process in the CAN. The proposed technique is introduced and the modifications of the original CAN are discussed in section 3. Section 4 discusses simulation results of comparing TMR technique with the medium access control protocol that used in the traditional CAN method. A conclusion is given in the last section.

2. CAN MEDIUM ACCESS CONTROL PROTOCOL

The CAN protocol implements most of the lower two layers of the ISO/OSI reference model (Data Link and Physical layers). The medium access control protocol employed in CAN can be considered as one type of carrier sense multiple access with collision detection (CSMA/CD) with an additional feature may be named collision resolution (CR). Unlike CSMA/CD where the conflicted message senders are enforces to stop and resubmit their messages after a random interval, in CAN collision-detection, the message with the highest priority will gain access to the bus, while all other nodes switch to a “receiving mode". Thus CAN provides a non-destructive bus arbitration [2][3].

2.1 CAN International Standards

The ISO 11898 standard, titled “Road vehicles – Controller area network (CAN), was first published in 1993, followed in 1995 by an amendment that describes the extended frame. There are two ISO standards classifying the CAN protocol in terms of data rate; ISO 11898 which can handle speed up to 1 Mbit/sec and ISO 11519 that can handle speed up to 125 Kbit/sec.[4][5]

2.2 Message Broadcasting

CAN protocol is a message-based protocol not an address based protocol. Communication is addressed by message identifiers instead of station identifiers as in normal LAN. Each message has an identifier that is unique throughout the network; it defines the priority and the content of the message. [2][6]

2.3 Message Frame Architecture

CAN provides four different types of message frames:

Data Frame – Sends data
Data transfer from one sending node to one or several receiving nodes.
Remote Frame – Requests data
Any node may request data from one source node.
Error Frame – Reports error condition
Any node may signal an error condition at any time during a data or remote frame transmission.
Overload Frame – Reports node overload
A node can request a delay between two data or remote frames. For demonstration purpose, the data frame architecture is illustrated in figure1.

Bus Idle	S O F	Arbitration Field	Control Field	Data Field	CRC Field	ACK Field	EOF	IFS	Bus Idle

Fig. 1. The Standard CAN Data Frame Architecture [1] [2]

2.4 Dominant and Recessive Bus Level

The physical CAN bus uses a differential voltage between two wires, CAN_H and CAN_L. The dominant level (logic 0) always overrides a recessive level (logic 1), which is important especially during bus arbitration, the CAN bus level will be dominant in case any number of nodes in the network output a dominant level. The CAN bus level will only be recessive when all nodes in the network output a recessive level. [2][3][5]

2.4 Bus Arbitration

CAN avoids message/data collisions by using the message ID. A CAN node checks if the bus is busy (Carrier Sense) before sending a message. If the bus is free, several nodes could be sending at the same time (Multiple Access).

According to the flowchart shown in figure 2, each transmitting node sends the message ID "arbitration field" bit by bit. After sending each bit, each node compares its output signal with the actual bus level. If a node found that the bit value it has written is different from the bit value it read back, then it will stop transmitting, it has detected a collision and has lost in the arbitration process, consequently the node will switch into receiving mode. If the node has finished sending all arbitration bits without losing the bus arbitration, it will transmit the rest of the message. At this time all other CAN nodes in the network will have switched to receiving mode. [2][5]

Figure 3 shows an example where three nodes in a four node CAN network try to access the bus at virtually the same time. In this example node C will win the bus access.

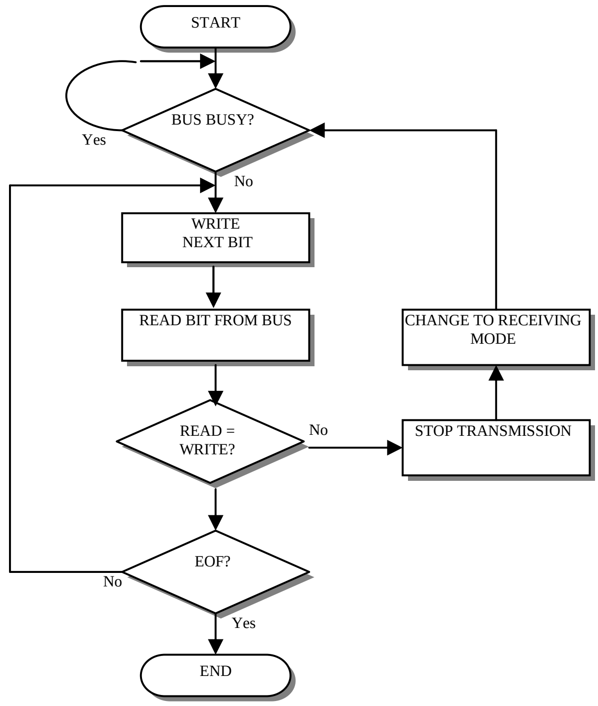

Fig. 2. Arbitration Flow Chart

The nodes in this example have the following message IDs:
A 1100101100 = 32C hex

B 1100110000 = 330 hex

C 1100101000 = 328 hex

The message ID of node D is of no significance, since it is not requesting bus access. According to this example and the CAN specification (lowest message ID represents highest message priority) node C must gain the bus access.

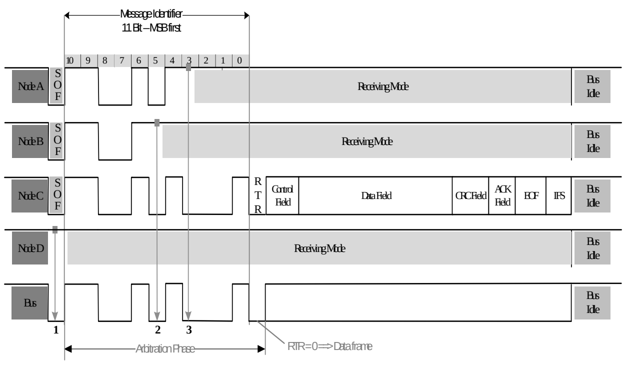

Fig. 3. Bus Arbitration Example.

3. THE PROPOSED TECHNIQUE: TAKING MULTIPLE REQUESTS "TMR"

In the industrial control networks, where the same quality of service should be ensured to a number of different applications, the original CAN protocol is not able to enforce either a fair division of the network medium among the nodes or a satisfactory distribution of the access delays experienced when transmitting messages. In the high traffic condition, network messages with low relative priorities messages may be delayed by any message with a higher relative priority wants to access the bus, even it could be delayed more than once by the same higher priority messages; irrespective of how many times it tries to access the bus and loses the arbitration. [4][5]

3.1 Traffic Pattern in the Original CAN

The traffic pattern in the original CAN protocol can be illustrated by an example. Figure 4. illustrates a situation where a message with identifier 7 becomes ready for transmission. In this example message 7 becomes ready for transmission, while at the same time instance message 3 becomes ready. The CAN arbitration mechanism ensures that message 3 is allowed to transmit while message 7 is forced to wait for the bus to become idle. While serving message 3 and before the bus becomes idle, message 4 appears which has the priority over message 7 and will be sent before it.

Obviously, the example illustrates the possibility of delaying message 7 for a very long time, although it was ready before many other messages. This delay is caused by its relative low priority compared with them. Message 7 is prevented from being transmitted by a higher priority messages, even by messages 3 and 4 more than once, and it takes a long time before it is eventually served.

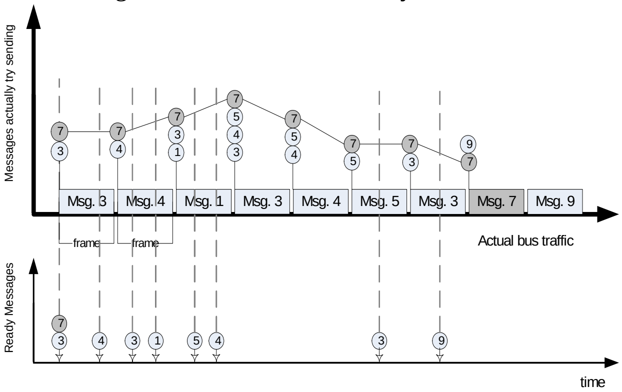

Fig. 4. A traffic pattern in a CAN network

3.2 Traffic Pattern in the TMR-CAN

A modification of the original CAN protocol is introduces, named taking multiple requests (TMR), which enforces a fair behavior in accessing the network bus, and it reduces the low priority messages' response time, by increasing its opportunity to get the same QoS that is given to the other higher priority messages, even in the case of high traffic. With the proposed technique, the network is enforced to work in a way that is similar to which is illustrated in figure 5. Where a group of messages which request access of the bus at the same time are considered as a group. Each message in the group is allowed to access the network bus according to its priority for one time only till all messages, within the group, are processed.

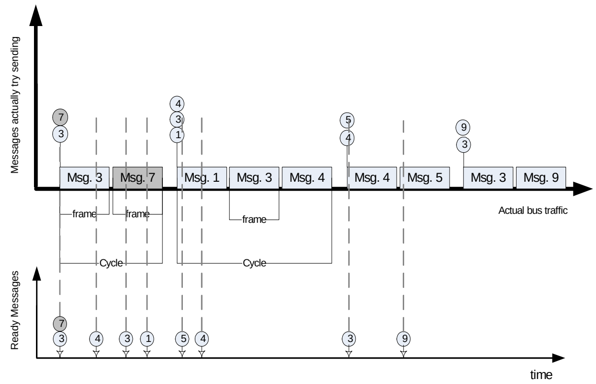

Fig. 5. A traffic pattern in the TMR-CAN.

This behavior could be achieved by limiting the number of times each message can be sent within a specific period of time, known as a cycle. Each node will determine if it is being accepted as a member of the selected group of the current cycle and if it has the right to access the network bus.

3.3 The Proposed Modification

Before going into the explanation of the proposed method, we need to have a closer look at the original CAN frame and review some of its important fields. The intended fields are the SOF, EOF and IFS, where they play a major role in the proposed technique.

SOF (1 Bit): The dominant Start of Frame (SOF) bit represents the start of the frame and the arbitration field follows right after the SOF bit. A CAN node, before attempting to access the bus, must wait until the bus is idle.

End-of-Frame Field (EOF): Each frame is terminated by a bit sequence of 7 recessive bits.

IFS (3 bits, recessive): The Interframe Space (intermission) represents the minimum space between frames, thus indicates the end of transmission and the bus becomes idle.

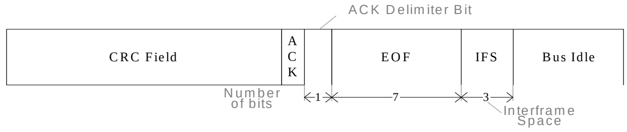

Fig. 6. The frame termination fields

Each CAN message frame will be terminated by a sequence of 11 recessive bits: The ACK Delimiter bit in the

Acknowledgement Field (1 bit), the EOF Field (7 bits) and the IFS Field (3 bits). As shown in figure 6.

3.3.1 TMR Frame Format

The TMR frame format follows exactly the original CAN frame format except the following modification: An additional field is attached at the end of the original CAN frame. This new field will indicate the end of cycle (EOC). The EOC field can be made to include only one recessive bit. However, we propose making the EOC field to include three consecutive bits. This is done to ensure robustness in a way similar to the IFS field. Figure 7 illustrates the TMR frame format in comparison with the original CAN frame.

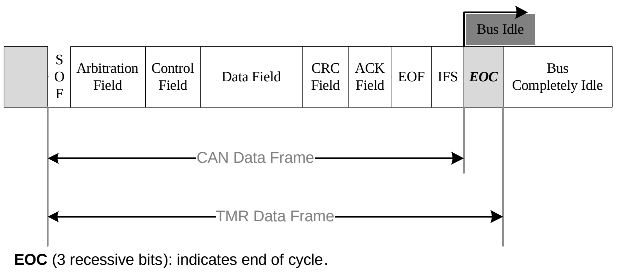

Fig. 7. Sample frame format of TMR including the EOC

Analogous to the CAN message frame, the TMR frame will be terminated by a sequence of 11 recessive bits plus 3 recessive bits for EOC field. On other words the TMR frame format will be terminated by a sequence of 14 recessive bits as shown in figure 8.

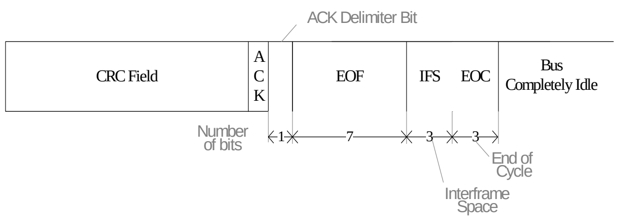

Fig. 8. The modified frame termination fields

3.3.2 Bus idle and bus completely idle conditions.

Referring to figure 7,8, after elapsing of the IFS field the bus is considered idle; and when the bus is found recessive for a given additional bits after the IFS bits, it is assumed to be completely idle. By monitoring the EOC bits, it is possible to detect the bus completely idle condition, with negligible effects on protocol complexity and with almost no effects on communication efficiency.

3.4 TMR Network Traffic Rules

In the TMR method, traffic in the network is divided into transmission cycles. In each cycle a group of messages will be transmitted in sequence according to their priority and without repetition.

Figure 9 shows a flowchart for the arbitration process in the proposed TMR method. Access to the network and arbitration process is described by the following points:

- The appearance of 14 consecutive recessive bits on the network bus indicates that the bus is completely idle and a new cycle which we will define as the “current cycle” can be started if there are nodes requesting access to the network.
- An internal node which loses arbitration because of its low priority will continuously try to access the bus by sending SOF bit whenever it monitors an IFS bits (bus idle condition i.e. 11 consecutive recessive bits).
- Nodes which did not transmit an SOF bit at the beginning of the current cycle are considered as “external” to the current cycle, these external nodes will switch to receiving mode and they will wait till the next transmission cycle.
- An external node can attempt accessing the bus whenever it monitors an end of cycle (completely bus idle condition i.e. 14 consecutive recessive bits). In summary: An internal node waits for a bus idle while an external node waits for a completely bus idle.
- An internal node which succeed in accessing the bus will transmit its message frame according to the formal CAN protocol format and then it will attempt to establish an EOC condition by transmitting three recessive bits. Finally, it will flag itself as an external node.
- The three EOC bits transmitted by an internal node which has successfully accessed the bus will not actually appear on the bus because they will be written over by the SOF bits transmitted by the pending internal nodes which start transmitting immediately after sensing a bus idle condition (IFS bits). Only the EOC bits transmitted by the last internal node will appear on the network bus, thereby, establishing a completely bus idle condition and signaling the external nodes to transmit SOF bits (if they have a ready message) in order to be considered in the next transmission cycle.

In the TMR method higher priority messages, will be enforced to send just once per cycle and allow lower priority messages to be sent. This is quite reasonable, so that high priority messages should not be allowed to occupy the network bus for a long time.

It is obvious that, the duration of the transmission cycle is not constant but depends on the number of message frames inside the cycle.

The TMR method implements a more effective control over the collisions resolution which still relies on the conventional arbitration mechanism of CAN.

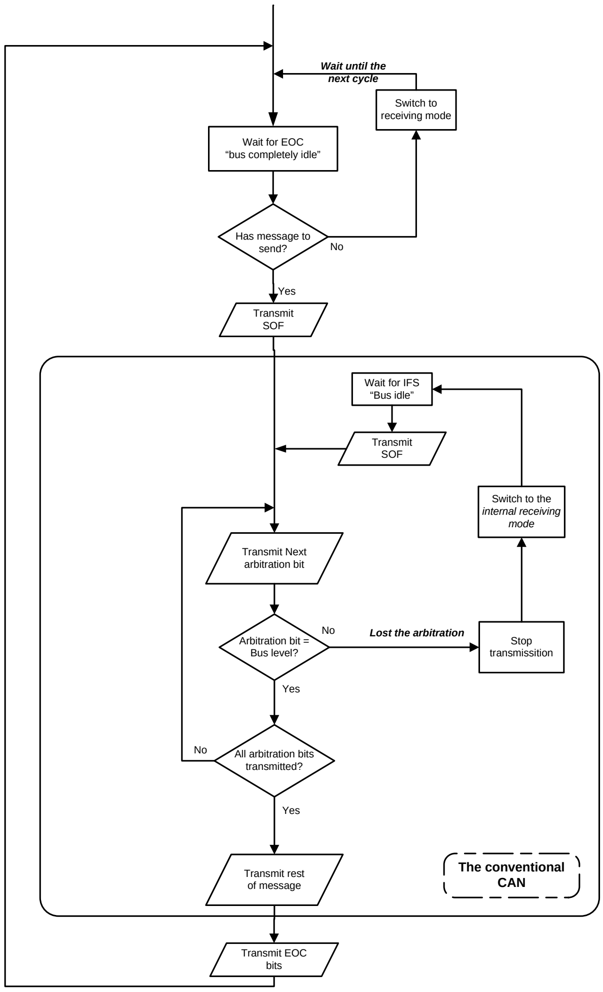

Fig. 9. TMR arbitration process flowchart

3.5 Distinguish Between Original-CAN and TMR-CAN

Fortunately the designer of CAN frame provided reserved bits which may be used for future modifications as shown in figure 10. We can use the reserved bit r0 to distinguish whether the frame format is a CAN format or TMR format. We suggest using the reserved bit r0 as follows:

r0 = 0 conventional CAN frame format

r0 =1 TMR-CAN frame format

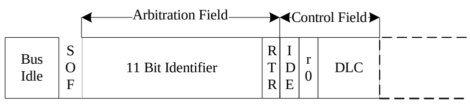

Fig. 10. The reserved bit r0 in the control field.

4. Comparison and Simulation Results

In this section a comparison is made between the original CAN protocol and the proposed TMR technique. The comparison concentrates on the fact that the medium access method for the TMR technique is similar to that used in the original CAN protocol in most aspects; the main difference comes from modifying the way of accessing the bus, by dividing the traffic into transmission cycles. We estimate the improvement in fairness provided by the TMR technique by means of simulated comparison. The simulation compares the average time required for a message to access the network bus (arbitration time) between the two methods. The method which requires lesser time will be considered as having more fairness.

4.1 Modeling The Number of Messages

Estimating the average arbitration time for a given message requires determining the number of messages that will be sent in a time period. This period of time starts when the given message is ready for transmission till the point of time when the message gets access to the bus. In this section the following scheme is followed in modeling the number of these messages:

- The number of messages which are ready for transmission will be denoted by "n".
- The number of messages which are ready for transmission and have a priority higher than the priority of the message under consideration will be denoted by "n_p". The parameter n_p will be given a random value in the range $(0-n)$.
- The number of messages which have been sent and seek repeated transmission before serving the message under consideration will be denoted by "n_r". The parameter n_r will be given a value equal to n_p multiplied by a repetition factor (R_F) which depends also on the degree of network traffic.
- The number of messages which are included in the current cycle which is already in progress will be denoted by "n_c".

4.2 Computation of Arbitration Time t_a

A. For the original CAN protocol

The arbitration time (t_a) includes the time of transmitting n_p messages and the time of transmitting n_r messages.

$$t_a = \sum_{i=1}^{np} tf + \sum_{i=1}^{nr} tf \qquad (1)$$

t_f : is the frame time

B. For the TMR technique. TMR format modification can be use either with the st

The arbitration time includes the time of transmitting n_c messages and the time of transmitting n_p messages. The second term is because the considered message is delayed by n_p messages which are in the same cycle.

$$t_a = \sum_{i=1}^{n} tf + \sum_{i=1}^{np} tf \qquad (2)$$

The term n_p is the same in the two methods. This is necessary to make an equitable comparison between them; where at a certain time, the node joins the same number of nodes in competition either for CAN method or TMR method.

4.3 Arbitration Time for 128 Different Messages

In this section we apply equations (1) and (2) to determine the arbitration time assuming the number of messages is 128.

In figure 11, we assume that the value of message repetition factor (R_F) is 0, while in figure 12 shows the arbitration time when the value of message repetition factor (R_F) is 2. In figure 13 the value of repetition factor (R_F) is 4.

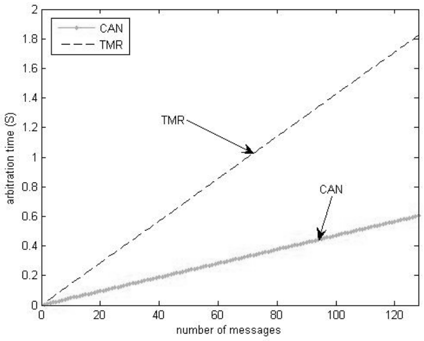

Fig. 11. Average arbitration time with minimum RF,

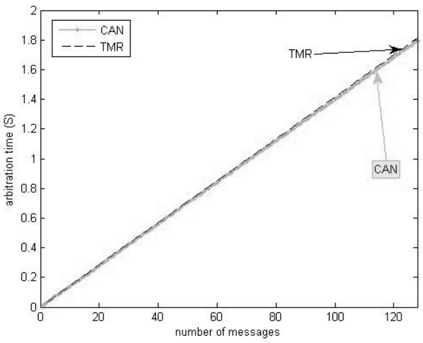

Fig. 12. Average arbitration time with medium RF.

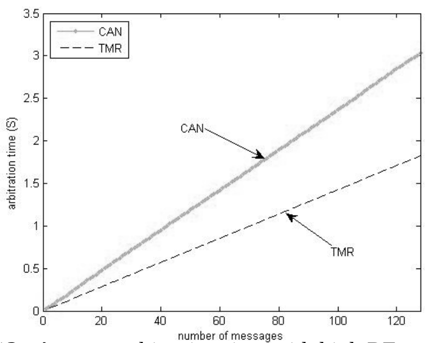

Fig. 13. Average arbitration time with high RF.

From these three figures, it's clear that the proposed technique ensures arbitration time smaller than that appears in the original CAN when the network experienced high traffic.

5. Conclusions

This paper introduces an explanation of the arbitration mechanism that controls the media access used in CAN. TMR technique that improves the fairness of the bus access is illustrated. We also provide an explanation of how the new technique enforces the network to behave in a way that permit an equal chance for all messages. This fairness behavior is ensured by dividing the transmission to cycles, where each message can be sent just once per cycle. This improvement is achieved by adding an additional field to the original CAN frame, called End of Cycle (EOC).

From simulations results, it is clear that when the same message is assumed to be sent more than once " may be called heavy traffic", our technique gives good results and improves the transmission rate.

TMR technique will be useful in networks that contain certain situations. These situations are:

- When the control applications require the same quality of service to be ensured to a number of different messages. Since each application must have its own unique identifier, it is not possible to make the network to consider two or more applications as equally urgent.
- Also when high repetition factor is expected "the message is repeated frequently"

References

[1] Gianluca Cena and Adriano Valenzano, " Achieving round-robin access in controller area networks", Industrial electronics, IEEE Transactions on Volume 50, Issue 4, Nov. 2004 Page(s):1093 - 1100

[2] ROBERT BOSCH GmbH, “CAN specification 2.0”, stuttgart 1991.

[3] “Controller Area Network – A serial bus system – not just for vehicles”, esd gmbh Hannover.

[4] Gianluca Cena, Adriano Valenzano, “A Distributed Mechanism to Improve Fairness in CAN Networks”, Dipartimento di Automatica Informatica, Politecnico di Torino, 1998

[5] G.Leen, D. Heffernan, “Time-triggered controller area network”, computer and control engineering journal, 2001.

[6] H. F. Othman, Y. R. Aji, F. T. Fakhreddin, A. R. Al-Ali "Controller Area Networks: Evolution and Applications", information and communication technologies, 2006. ICTTA'06.2nd.