Second International Conference on Electrical Engineering
25-26 March 2008
University of Engineering and Technology, Lahore (Pakistan)

Starvation Free Controller Area Network using Master Node

Ali Faisal Murtaza and Zubair A. Khan
Department of Electrical Engineering,
University of Engineering & Technology,
Lahore-54890
Email: ali_faisal355@hotmail.com, zubair.khan@kics.edu.pk

Abstract - Controller Area Network (CAN), a serial communication protocol, is widely used in industry in embedded applications. The fixed priority arbitration mechanism in CAN results starvation of lower priority nodes because in the presence of a set of higher priority messages, the low priority nodes are restricted to appear on a CAN network. Since lower priority messages can not change their priorities so they suffer from starvation and continuously lose arbitration due to the presence of higher priority messages. This paper addresses this problem and proposes a method to make a CAN network starvation free by adding a Master node in a CAN network. This Master node is fully capable of changing priority of starving node and allows the starving nodes to appear on a CAN network. It makes the CAN network starvation free and provides the ability to change priority dynamically. Master node monitors CAN network and remains silent until it keeps receiving messages of all the nodes and become active on detection of starving node, a node not contributing in communication. In active mode, this node tries to make sure that each node keeps participating in communication.

A hardware setup along with CAN king Software has been used to verify the results.

I. INTRODUCTION

Controller Area Network (CAN) is a multi-master message broadcast system originally developed in 1986 by Robert Bosch Gmbh [1]. Controller Area Network basically consists of a two wired copper bus normally known as a CAN bus, which acts as a transmission medium for a CAN network. Each node can transmit message on CAN bus without taking permission from any other node on a CAN network. Each message has the identifier field. This identifier field has two functions. It decides priority of message on a CAN bus. And second is that it defines meaning of the message so that other nodes apply filter criteria on it whether to accept or reject it.

If two or more nodes are trying to access the bus at a same time then their collision is resolved through a non-destructive bitwise arbitration mechanism. Non-destructive means that no data will be lost. And arbitration mechanism is described in figure below. In AND-wired setup, bus takes bit '0' as a dominant one. As it is clear from the figure that both Node1 and Node2 try to access the bus at a same time, At point where Node2 finds that it transmits bit '1' and state of CAN bus is '0' so it will halt its transmission and enters into receiving mode as it loses arbitration from Node1, which has higher priority than Node2. Therefore, Node2 waits until the bus is free to resend its message. When bus is free, Node2 again transmits its message but again some higher priority message comes in its way then again it struggles to access the bus. As CAN works on a fixed priority based arbitration mechanism. It means that each message has a single unique identifier. As priority of each message on CAN bus is decided by its identifier. Therefore priority of each message on CAN bus is fixed. So, one can say, in that regard CAN is not that much flexible. And due to which sometimes CAN fixed priority based arbitration mechanism becomes controversial due to its limitations. As low priority messages do not change their priorities therefore they suffer from starvation due to higher priority messages [2]. In that respect CAN becomes non-versatile and a question arises on the adaptability of CAN.

So this paper works on this issue that in some conditions CAN network will switch from fixed priority based arbitration mechanism to variable or dynamic priority based arbitration mechanism. Therefore, there will be some mechanism which helps these low priority messages to increase their priorities when they struggle to gain access to the bus thus making CAN network starvation free by giving it the ability of dynamic priority based message scheduling. So for that purpose, Master node is introduced in CAN.

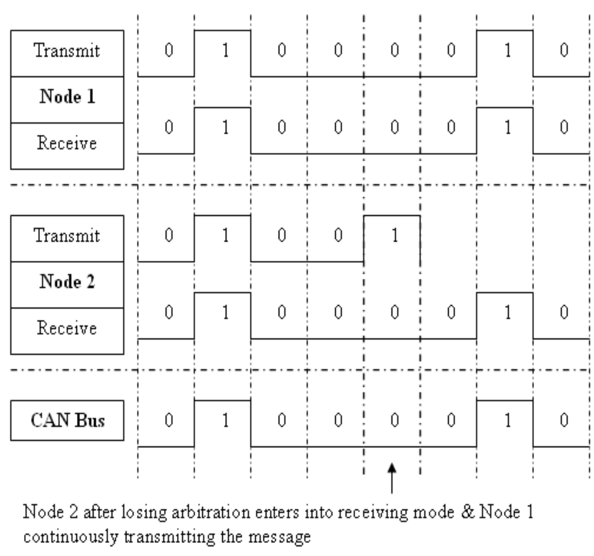

Figure 1: Arbitration Mechanism [1]

978-1-4244-2293-7/08/$25.00 ©2008 IEEE.

Authorized licensed use limited to: University of Shanghai For Science and Technology. Downloaded on April 09,2021 at 13:14:01 UTC from IEEE Xplore. Restrictions apply.

II. Proposed Dynamic Priority Based Message Schedulig For Starvation Free CAN Network

In this paper, the scheme is proposed that there will be a Master node in CAN network which will monitor CAN network all the time and remains silent as far as it receives messages of all the nodes but become active as it finds that some nodes are not contributing in the communication. Like low priority messages when try to access the bus, higher priority messages come in their way. Therefore, low priority messages lose arbitration from these higher priority messages and suffer from starvation. So, in active mode, Master node is fully capable to change the priorities of these low priority nodes to such that they can easily appear on the bus thus making a CAN network starvation free. In active mode, Master node is fully capable of changing the priorities of each node on network thus providing CAN the capability of dynamic priority based arbitration mechanism. This Master node and whole procedure is implemented in CAN network without modifying anything in the CAN communication protocol. The proposed scheme will reduce total number of application messages in CAN network but it gives relief to those CAN messages which are suffering from starvation due to higher priority messages and do not appear on CAN bus.

A. Introduction of Master Node in CAN

With the induction of Master Node in a CAN network, it gets the ability to switch from fixed priority based arbitration mechanism to dynamic priority based arbitration mechanism. It is basically Master node that will judge those critical conditions in which lower priority messages do not appear on CAN bus, and therefore increases priority of only those messages that struggle to appear on CAN bus. In this way, Master node will not increase priority of all messages on the CAN bus. So, with Master node, those low priority messages which are suffering from starvation due to higher priority messages now become starvation free.

B. Characteristics of Master Node

The Characteristics of this node be:

1. It monitors CAN network all the time.
2. Master node has information of all nodes in the CAN network and if an extra node is added in the CAN network about which Master node does not know then even in that case Master node is fully capable to enter the information of that node in its memory.
3. This node is free means that it does not perform any application i.e. it does not attach to any Sensor or Actuator.
4. This node has the highest priority so it can interrupt the CAN network.
5. Whenever Master node sends message on CAN Bus, every node is bound to accept that message.
6. Master node recognizes each node with its message identifier.
7. Master node has got the ability to change the priority of any node on a CAN network.

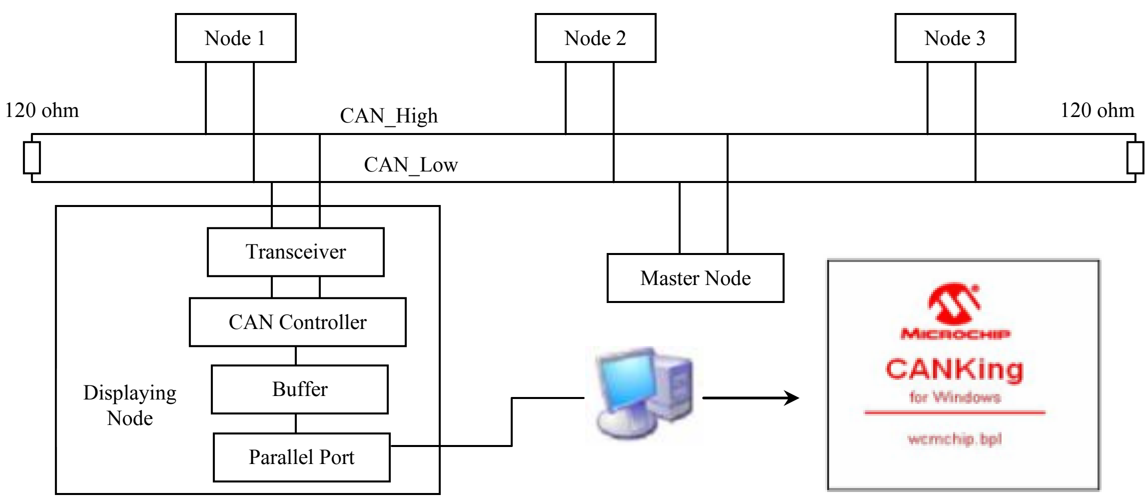

Figure 2: Complete Setup

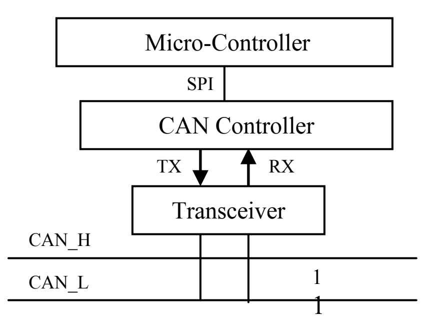

Figure 3: Internal Structure of Nodes 1, 2, 3 & Master Node [9]

Authorized licensed use limited to: University of Shanghai For Science and Technology. Downloaded on April 09,2021 at 13:14:01 UTC from IEEE Xplore. Restrictions apply.

III. Allocation Of Dynamic Priorities

Messages are basically generated by the nodes on CAN network, therefore when we are assigning priorities to different messages then those priorities are saved in the nodes. Whenever a node wants to transmit its message then it will send it with its pre-programmed priority. As CAN works on a fixed priority based message scheduling therefore whenever a CAN node generates a certain message, it will always generate it with a single priority. And this is due to the fact that each message on CAN network has identifier field, which decides the priority of that message on CAN bus and also defines the meaning of that message. Every node on CAN network applies filter criteria on this identifier field whether to accept it or reject it. Like, if 1300 is identifier of tachometer speed. When all nodes receive this message then with the help of this identifier i.e. 1300 they know that this is tachometer speed message. So, if any node requires this message then it will accept it otherwise reject it.

Therefore, following rules are applied during assigning the dynamic priorities to a CAN network.

Rule # 1: If we are assigning more than one priority to a single message then all the other nodes should know that changed priority describes the same message. Like we assign two priorities to a tachometer speed message i.e. 1300 and 15, then all other nodes should know that meaning of both identifiers is the tachometer speed message.

Rule # 2: If 1300 and 15 are tachometer speed message identifiers as in the above case, therefore these two priorities should not assign to any other message. Like, if we assign 1300 identifier to both tachometer speed and winding current. Then all nodes will confuse that this is tachometer speed message or winding current message. Therefore all messages on CAN Network should have distinct priority.

Conclusion: It is decided that only two priorities are given to a single message. Therefore, two pools are created for this purpose, one is known as original priority pool and other is known as changed priority pool, which are discussed in the upcoming sections.

Note: No node cannot change its priority without receiving message from Master node.

A. Algorithm of Assigning the Priority

The algorithm is applied on Standard identifier but it can easily apply to Extended identifier as well. Standard identifier consists of 11-bits, therefore there are 2^11 = 2048 number of messages. First two priorities are assigned to Master node. Remaining priorities are divided into two pools, one is known as original priority pool which consists of all lower half priorities and the other is known as changed priority pool which consists of all upper half priorities. Whenever a priority is assigned to a message, then it takes first priority from original priority pool and the other priority from changed priority pool.

Priority of the Master Node: {00,01}, where 00 priority belongs to Master node and 01 priority is the reserved one and has not been used.

Original Priority Pool: All nodes take original priority from this pool: {1025, 1026, 1027,………...,2045, 2046, 2047}

Changed Priority Pool: All nodes when they changed their priority take their changed priority from this pool.
{02, 03, 04, 05, 06, ….......………………….1022, 1023, 1024}

Guideline: Whenever any node takes priority from changed priority pool, it will take the next free priority.

Procedure: Node1 takes one priority from original priority pool (O.P.P) and one from changed priority pool (C.P.P).
O.P.P: {1025, 1026, 1027, ……………….., 2045, 2046, 2047}
C.P.P: {02, 03, 04, 05,……………………..,1022, 1023, 1024}

Node1 takes original priority of 1027. As this is the first node to take priority from C.P.P so it will take first priority from C.P.P which is 02.

Node1 = {1027 / 02}

The original priority of Node1 is 1027 therefore it always sends message on CAN bus with priority 1027 and it will not change its priority from 1027 to 02 unless it receives message from Master node to do so. Detail of this changed priority method is given in the upcoming sections.

Now, Node2 when take priorities, then it has same pools but with one priority less in each pool as these priorities are already assigned to Node1.
O.P.P: {1025, 1026, ~~1027~~, ……………….... 2045, 2046, 2047}
C.P.P: {~~02~~, 03, 04, 05, ……………….……….1022, 1023, 1024}

Node2 can not take priority 1027 from O.P.P (Original Priority Pool) and 02 from C.P.P (Changed Priority Pool), as these two priorities are already assigned to Node1.

Node 2 takes 1029 as its original priority. And as next free priority in C.P.P is 03, so Node2 takes 03 as its changed priority.

Node2 = {1029 / 03}

So, for Node3 following pools are left:
O.P.P: {1025, 1026, ~~1027~~, 1028, ~~1029~~,…, 2046, 2047}
C.P.P: {~~02~~, ~~03~~, 04, 05,…………..……,1022, 1023, 1024}
And so on.

IV. Working Of Master Node

This node works in two modes.

Silent Mode: In this mode, Master node monitors CAN network and remains silent as far as it receives messages of all the nodes. i.e. If there are 'N' no of nodes in a CAN network, and after a specific period of time or specific number of messages received, 'N' no of different messages received.

Active Mode: In this mode, Master node becomes active as it realizes that it does not receive messages of all the nodes. i.e. If there are 'N' no of nodes in CAN network, and after a specific period of time or specific number of messages received, 'N-1' no of different messages received.

V. Algorithum Of Master Node

A. Master Set of Nodes

Master node has a set of nodes. In this Master set of nodes, identifiers of all the nodes are included. Whenever, Master node receives some message then Master node only checks the identifier of that message and removes that identifier from its Master set of nodes as message of that node appears on the bus

Authorized licensed use limited to: University of Shanghai For Science and Technology. Downloaded on April 09,2021 at 13:14:01 UTC from IEEE Xplore. Restrictions apply.

so there is no need to change the priority of that node. Similarly, this procedure goes on unless the time limit is crossed. The time interval can be set by means of time i.e. 10 minutes etc or even after specific number of messages, like 500 messages etc (depends upon the system designer). Therefore, after the time limit is crossed, Master node becomes active and applies the algorithm to those nodes only which are not excluded from Master set of nodes or in other words which do not appear on the bus. This procedure is discussed in the next section. After completing its job, Master node becomes silent again, reset its Master set of nodes, interval limit and restart checking a CAN network, now again time limit is crossed and Master node receives messages of all the nodes i.e. all nodes are excluded from the Master set of nodes, then there is no need to change priority of any node, therefore Master node remains silent. And move on to next cycle and so on.

B. *Criteria to Change the Priority*

Master node changes priorities of those nodes only which do not appear on CAN bus, in other words, those nodes which lose bus arbitration to other nodes and suffer from starvation. Therefore, there is no need to change priorities of those nodes which appear on the bus.

After the time limit is crossed, Master node picks a single node from those nodes which do not exclude from its Master set of nodes. Master node twice changes its priority and then picks the next struggling node from its Master set of nodes apply the same procedure and so on. By twice changing the priority means that Master node first generates message for struggling node to change its original priority (taken from original priority pool) to changed priority (taken from changed priority pool) so that it can easily appear on CAN bus and after sometime Master node changes priority of that node back to original one and then immediately picks the next node, apply the same procedure and so on. This is due to the fact that for example if twenty nodes struggle to appear on the bus, and Master node changes priority of all nodes at a same time, then Master node is just transforming these nodes from lower priorities to higher priorities. The difference is that previously these nodes struggle to appear on the bus and now these nodes create problem for other nodes to appear on bus.

Another advantage of changing the priority one by one is that when any node changes the priority then it always have the highest priority in CAN network, for example Node1 has original priority of 1027 and has changed priority of 05, and another Node6 has original priority of 2045 and has changed priority of 1024 which is the lowest priority in changed priority pool. Now, after the time limit is crossed, if Node1 and Node6 do not appear on the bus then Master node changes Node1 priority first from 1027 to 05 and then after a short interval of time, Master node again says to Node1 to go back to your original priority. And then immediately says to Node6 that change your priority from 2045 to 1024. It means that only one priority is active from changed priority pool at a time, which is the highest priority on CAN bus. It shows two facts, one is that when ever any priority is active from changed priority pool, it always wins the bus arbitration and second is that next free priority rule can easily be applied to pick the priority from changed priority pool as only one priority is active from changed priority pool at a time. So, there is no competition occur between the priorities of changed priority pool.

C. *Master Node Adaptability*

Also in case if an extra node is added in a CAN network about which Master Node does not know anything even then Master Node can easily add that node in its memory and can easily apply the whole procedure to this new node as well. Like if a message with identifier '2031' comes about which Master node does not know, Master node adds this '2031' identifier in its Master set of nodes directory, knowing that an extra node is added in a CAN network, and now from onwards Master node also checks the condition of this node whether messages of this node appears or not.

VI. Mathematical Model Of Master Node

Master Node has the following set of Nodes.

$$MASN = \{N1, N2, N3, \ldots\ldots\};$$

Remember each node is recognized by its identifier.

Where,

MASN = Master set of Nodes.

N1 = Identifier of the Node1.

N2 = Identifier of the Node 2.

N3 = Identifier of the Node3.

& so on.

N = No of Nodes

Suppose there are three nodes in the Master set of nodes, so N=3.

Note: All these information will be changed after the Upgraded Step.

Step#1: If the message is received then move to Step2, otherwise remain in this step.

Step#2: In this step, Master node checks that the message received belongs to which Node.

As N = 3; so there are three nodes in the Master set of nodes, so Master node checks that the received message belongs to which node.

$$N1,i = [(N1 - IDMi) * 2]\text{^}2;$$

It checks the first node here

Where,

N1,i: N1 = Identifier of Node1 & i = Message No

IDMi = Identifier of the Message No i.

If N1,i = 0, then $MAS = \{\cancel{N1}, N2, N3\}$

It means that the message received belongs to this node, so this node should be excluded from the Master set of nodes and then move to Step3 otherwise check the next node.

Next Node: $N2,i = [(N2 - IDMi) * 2]\text{^}2;$

It checks the second node here

If N2,i = 0, then $MAS = \{N1, \cancel{N2}, N3\}$

It means that the message received belongs to this node, so this node should be excluded from the Master set of nodes and then move to Step3 otherwise check the next node.

Next Node: $N3,i = [(N3 - IDMi) * 2]\text{^}2;$

It checks the third node here

If N3,i = 0, then $MAS = \{N1, N2, \cancel{N3}\}$

It means that the message received belongs to this node, so

Authorized licensed use limited to: University of Shanghai For Science and Technology. Downloaded on April 09,2021 at 13:14:01 UTC from IEEE Xplore. Restrictions apply.

this node should be excluded from the Master set of nodes and then move to Step3 otherwise check the next node.

Next Node: As there is no node left, so move to Upgraded Step.

Upgraded Step: If the Master Nodes reaches this point then it means that the message received by Master Node does not belong to its Master set of nodes, in other words, an extra node is added in the CAN network, so it will add that node in its set of nodes and therefore N i.e. number of nodes is increased by one every time if Master Node enters in this step. After this up gradation, the Master node moves into step 3.

MASN = {N1, N2, N3, N4};

N = N+1

Note: After the upgraded step Master node has the new set of nodes and therefore all the criteria is applied to this new Master set of nodes. Like, previously the criteria is applied to only three nodes, now as extra node is added therefore the criteria is applied to the four nodes.

Step#3: Master Node will check that whether the message limit is crossed or not:

MAS = a – i ;

a = Interval limit

i = Message No

If MAS = 0, then it means that the message limit or interval limit is crossed and move to Priority Changed Step otherwise move to Step 1.

Priority Changed Step: In this step, Master node picks single node at a time present in the Master set of Nodes, apply the procedure and then pick the next node, and so on. The procedure is that Master node changes the priority of node from original to the changed one and after sometime it again changes the priority of that node back to original one. After completing the procedure on first node, it picks the second node apply the came procedure and so on.

After changing the priority of all nodes present in the Master set of Nodes, the Master Node resets its time limit, and its Master set of nodes. And then move onto the next cycle.

Note**:** If a node is added in the Master set of Nodes, then it keeps the last updated Master set of Nodes.

VII. Results

This whole idea and scheme is applied on a hardware setup and the results are taken by using the CANking Software. CAN works on a static priority based message scheduling i.e. each message on CAN network can have only one identifier. This identifier decides the priority of message on CAN bus as well as meaning of the message. Therefore, when higher priority messages and lower priority messages try to access the bus at the same time, then their collision is resolved through a non-destructive bitwise arbitration mechanism. There is a huge possibility that the set of higher priority messages do not allow lower priority messages to appear on bus. Thus these low priority messages suffer from starvation.

A. Graphical Representation

Below are the figures showing results and conclusions in graphical form. In the graphs below, horizontal axis shows the number of messages. As we move towards right, number of messages goes on increasing. While, vertical axis shows the percentage of different messages appears on CAN network. Each message is recognized by its identifier. Different message means different identifier.

As in this paper, three nodes are prepared.

Node1: It has original identifier of 1040 and its changed priority is 2.

Node2: It has original identifier of 1030 and its changed priority is 3.

Node3: It has original Identifier of 1055 and its changed priority is 4.

All three nodes continuously transmitting their respective messages and try to access the bus at a same time. Using the test setup made in this paper, results are recorded on 50 messages at a speed of 125 Kbits/sec. Therefore, if all three nodes contribute in the communication, then percentage of different messages appears on CAN network is 100%. And if two nodes contribute then it is 66.33% and if only one node then it is 33.33%. As this paper addresses the problem of arbitration, therefore, all three nodes continuously transmitting their respective messages and try to access the bus at a same time and results are recoded on 50 messages without Master node and with Master node.

Conventional CAN Network:

(Without Master Node)

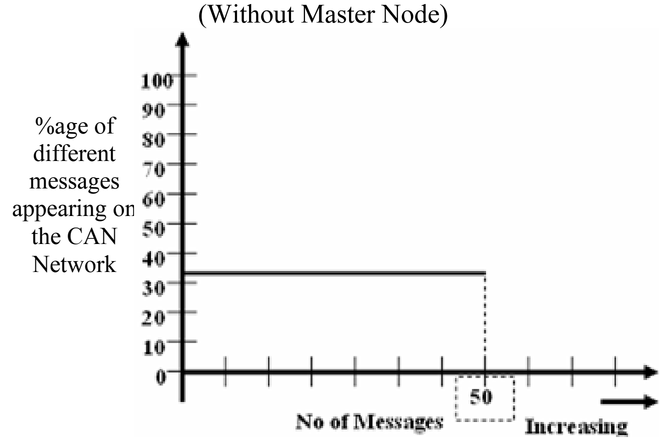

Fig. 4. Conventional CAN Network Results

As all three nodes continuously transmitting their messages at a same time, therefore message with identifier 1030 appears on the bus, and percentage of different messages appears on CAN network is 33.33 and it stays on there. This is due to the fact that whenever two low priority messages i.e. 1040 & 1055 try to appear on the CAN bus, higher priority message i.e. 1030 always comes in their way and dominates the bus.

Proposed CAN Network:

(Induction of Master Node to give Dynamic Priority to CAN Network)

Master node interval limit is set at 25 messages. At 25 messages, when Master node sees that only one message with ID=1030 is received, and other nodes do not involve in the communication. Therefore, after 25 messages, it increases the priority of those nodes one by one which does not contribute in the communication.

Authorized licensed use limited to: University of Shanghai For Science and Technology. Downloaded on April 09,2021 at 13:14:01 UTC from IEEE Xplore. Restrictions apply.

It first transmits a message to Node1 to change its priority. In response, Node1 changes its preprogrammed original priority of 1040 to preprogrammed changed priority i.e. 02. After 3 seconds i.e. round about 12 messages, Master node says to Node1 to go back to your original priority, and immediately says to Node3, increase your priority therefore Node3 changes its priority from 1055 to 04. Again after 3 seconds i.e. round about 12 messages, Master node says to Node3 to go back to your original priority. After that Master node reset its Master set of nodes and interval limit and move on to next cycle. That's why at 50 number of messages, graph goes to zero, as new cycle will be started and no message is received.

With the induction of Master Node, one can see the difference that percentage of different messages appearing on CAN bus increases significantly as Master node is fully capable of increasing the priority of low priority messages which are suffering from starvation to such that they can easily appear on the bus. Therefore, all nodes contribute in the communication and thus making the CAN communication starvation free.

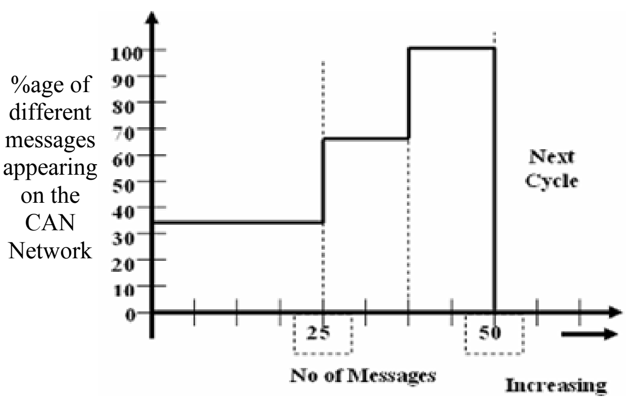

Fig. 5. Proposed CAN Network Results

VIII. CONCLUSIONS

It is concluded that the starving nodes struggling to appear on the CAN bus are able to appear on the bus because Master node is fully capable of changing the priorities of these nodes and allows these node to contribute in communication thus making a CAN network starvation free as shown in Fig. 5. The main purpose of Master node is that it tries to make sure that each node contributes in the CAN communication.

This tested setup contains three nodes however it can be extended to any number of nodes.

REFERENCES

[1] Steve Corrigan, *Introduction to the Controller Area Network (CAN) – Texas Instruments Application Report, SLOA 101*, 2002.

[2] K-Anwar and Dr. Z.A.Khan, "Dynamic Priority Based Message Scheduling on Controller Area Network", in *ICEE-2007*, 2007, ISBN No: 1-4244-0893-7.

[3] Robert Bosch, "CAN specification 2.0", Parts A and B, Sep. 1991.

[4] "Controller Area Network (CAN), an overview", http://www.can-cia.org/can

[5] Home page of the organization CAN in Automation (CiA), "CAN Protocol," 2004, http://www.can-cia.org/can/protocol.

[6] Khawar. M. Zuberi and Kang. G. Shin, "*Design and Implementation of Efficient Message Scheduling for Controller Area Network*", IEEE Tran, Computers, vol. 49, no. 2, pp. 182-188, Feb. 2000.

[7] IEEE Industrial Electronics Society, "IEEE Transactions on Industrial Informatics," http://www.ieee-ies.org/tii/iit02_2.shtml.

[8] ISO. International Standard ISO 11898: *Road Vehicles – Interchange of digital information – Controller Area Network (CAN) for high speed communication*, ISO 11898:1993[E], 1993.

[9] "MCP 2515 CAN Controller Interface with SPI Interface", http://www.alldatasheet.com/datasheetpdf/pdf/166906/MICROCHIP/MCP2515.html.

Authorized licensed use limited to: University of Shanghai For Science and Technology. Downloaded on April 09,2021 at 13:14:01 UTC from IEEE Xplore. Restrictions apply.