Performance Evaluation of a Method to Improve Fairness in In-Vehicle Non-Destructive Arbitration Using ID Rotation

Pusik Park[1, 2, *], Rustam Rakhimov Igorevich[1], and Jongho Yoon[2]
[1] Embedded & Software Research Center, Korea Electronics Technology Institute
[2] Dept. of Telecommunication & Information Engineering, Korea Aerospace University
South Korea
E-mail: parksik@keti.kr, rusyasoft@gmail.com, and yoonch@kau.ac.kr
*Corresponding author: Pusik Park

Received June 15, 2017; revised August 16, 2017; accepted September 6, 2017; published October 31, 2017

Abstract

A number of automotive electronics—safety, driver assistance, and infotainment devices—have been deployed in recent vehicles. This raises new challenges regarding in-vehicular network arbitration. A performance analysis of non-destructive arbitration has revealed a fairness issue. The arbitration prioritizes without collisions, despite multiple simultaneous transmissions; however, the performances of the highest priority node and the lowest priority node are very different. In this paper, an ID-rotation arbitration method to solve the arbitration-fairness problem is proposed. The proposed algorithm was applied to several engine control units (ECUs), including a controller area network (CAN) controller. Experimental results showed that the algorithm improved the fairness as well as the total throughput within a specific performance constraint.

Keywords: Controller area network, in-vehicle network, MAC, media access control, non-destructive bitwise arbitration, automotive

A preliminary version of this paper appeared in ICONI 2016, Dec. 16–19, Hong Kong. A preliminary version of this paper was presented at ICONI 2016, and was selected as an outstanding paper. This version includes performance-evaluation results on non-destructive bitwise arbitration in a CAN network. This research was supported by research grants from KAIA, No. 15RTRP-B109166-01, and KEIT, No. 10050293. This paper was supported by Konkuk University in 2016.

https://doi.org/10.3837/tiis.2017.10.022 **ISSN : 1976-7277**

1. Introduction

It is impossible to imagine today's cars without in-vehicle networks that connect hundreds of circuits, sensors, and other electrical components. If each component interconnection occurred via a dedicated wire through a point-to-point connection, the result would be an enormous number of wires. In-vehicle networking provides a more efficient method for complex communications.

The reliability and stability of an in-vehicle network becomes a vital issue when the electrical components and sensors control life-critical devices. A failure in the communication network, for any reason, would be crucial. Faults occur for various reasons in systems and networks. In this paper, we would like to concentrate on faults occurring in shared networks.

When the network has a ring topology and the network bandwidth is insufficient, sharing the token becomes a crucial problem. The cause of the problem is mostly related to fairness in token arbitration. The huge number of diverse engine control units (ECUs) in the modern vehicle cannot utilize a simple Round-Robin algorithm to share the token. Some ECUs generate and consume messages much more frequently than their counterparts do. Other ECUs generate messages very rarely, or even on user demand. This type of diversity between devices could cause a problem if no prioritization rules were applied.

As was mentioned, device prioritization is an important factor since some devices have a life-critical impact while others do not. A comparison of the brake and window-control systems could serve as a good example. Scheduling and priority-based network-access token distribution have been addressed by many researchers and defined by standards. Some details about these techniques will be given in Section 2.

In this paper, we try to address the fairness issue among prioritized nodes. The core of the problem entails solving a logical prioritized-lock issue. A logical prioritized lock is caused when a high priority node obtains the token and does not release it. Actually, the node releases the token when it finishes its transmission, but then requests the token again. Following most standards, it will win the competition and regain the token. That causes a long wait period for the other nodes. If all the remaining nodes have a low priority, it will not be an important issue. However, some of the nodes may also have a high priority task that is vital for human life. In the situation described above, those life-critical nodes would also wait. To avoid this, we suggest a method for improving fairness in in-vehicle networks by rotating IDs.

The proposed method has been was simulated using the OMNeT++ discrete event simulator and deployed on real devices interconnected using a controller area network (CAN). In Section 2, we introduce the technologies used, including a CAN bus, and other related research on this topic. In Section 3, the Non-Destructive Bitwise Arbitration Using ID Rotation (NDBA-IR) method is described and explained. Section 4 contains the OMNET++ simulation model of the proposed idea and the experimental simulation results from simulation. Section 5 describes the physical device-deployment experimental environment and the real experimental results. Section 6 compares and analyzes the experiment both sets of results from simulation and real device deployment. Finally, Section 7 concludes the paper.

2. Related Work

2.1 CANs (Controller Area Networks)

A controller area network (CAN) is an ISO-standard computer-network protocol designed for microcontrollers, and devices that communicate with each other without a host computer. CANs have gained widespread popularity for embedded control systems in certain areas; e.g., industrial automation, automotive, mobile machine, medical, military, and harsh-environment network applications.

The CAN bus was originally developed by Bosch as a multi-master message-broadcast system that specified a maximum signaling rate of 1 Mbps [1].

Bus topology-based networking, e.g., CAN, is simple and low-cost. However, a collision occurs when two or more nodes transmit packets at the same time. Collisions are a critical problem because they decrease the network performance. Even though a star topology provides better network performance than a bus topology, the additional equipment increases the cost. Thus, the bus topology has been deployed in various simple networking applications, although the total network bandwidth cannot be extended flexibly.

The CAN protocol utilizes Carrier Sense Multiple Access/Collision Detection (CSMA/CD) with a Non-Destructive Bitwise Arbitration (NDBA) method, which is possible through NRZ (Non-Return to Zero) digital signaling.

A CAN's identification numbers are prioritized. Smaller identification numbers have a higher priority. A message with the highest priority will never back off when a collision occurs. On the other hand, messages with a lower priority will back off and retransmit when the higher priority message has completed its transmission. The NDBA provides need-based bus allocation and delivers efficiency benefits that cannot be gained from either a fixed time-schedule allocation or destructive bus allocation.

Fig. 1 shows the complete network diagram of the Hyundai GENESIS DH sedan, first introduced in 2014. The automotive network consists of several network domains according to safety level. Six types of CAN network—D-CAN (Diagnosis CAN), P-CAN (Powertrain CAN), C-CAN (Chassis CAN), B-CAN (Body CAN), and M-CAN (Multimedia CAN)—interconnect with each other through a central gateway. Several local CANs connect the different devices.

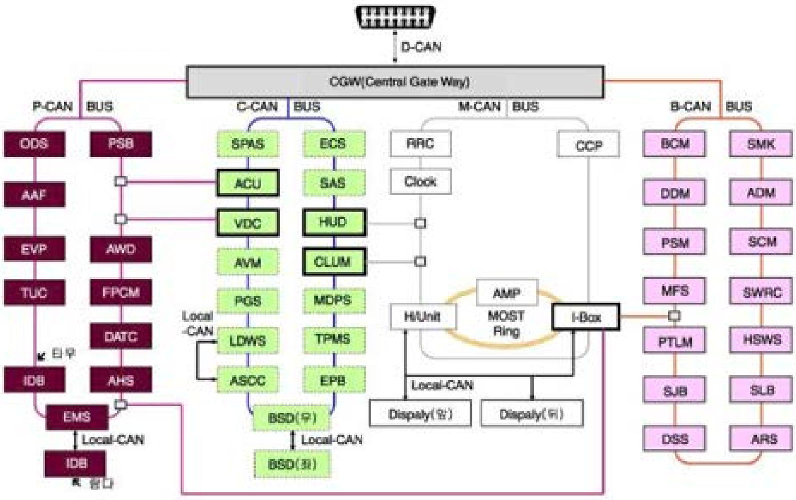

Fig. 1. Network Architecture of the Hyundai Genesis DH Sedan

The CAN buses are separated according to their purposes for safety, security, and efficiency. The all-wheel-drive (AWD) system in the P-CAN bus, which is responsible for the powertrain system, should have better reliability than the body-control module (BCM) in the B-CAN bus. This means that the current automotive network already provides independence across systems with different safety levels; however, the automotive network designer should not consider the interference problem of two different safety-level systems less than the one that uses only one network.

If a car consisted of a single CAN network and every device communicated through it, the interference from the extensive traffic would cause the devices' message deliveries to fail. For example, while the driver pushed a button to open a window, the automatic anti-skid braking system (ABS) could not work. To prevent a lower-priority device's busy status from interfering with a higher-priority device's transmission, the CAN protocol provides a non-destructive bitwise priority-arbitration mechanism. The CAN's priority feature takes an active part in the environment described above, which includes many different safety-level devices in the same network.

However, as shown in **Fig. 1**, recent carmakers have identified the priorities within the automotive systems and have decoupled them into different independent networks according to priority level. The higher priority devices are installed in a high-priority CAN network, e.g., the P-CAN bus in **Fig. 1**, and the lower priority devices are in a low-priority CAN network, e.g., the B-CAN bus. This networking strategy removes the interference caused by the low-priority devices.

The P-CAN network includes the devices most important for safety. The B-CAN network includes less safety-critical devices. Each network consists of similar safety-critical devices. In this network, devices in the same safety level should not compete for the same mission because every device should play an important role in the network. Thus, in a network that is aligned by priority, fairness and efficacy are more important than priority competition.

2.2 CAN FD (Flexible Data-Rate)

In the early 21st century, the lack of speed in the CAN data-link layer protocol needed improvement. Much research was devoted to optimizing CSMA using various methods. In 2011, Bosch started CAN FD (Flexible Data-Rate) development in close cooperation with carmakers and other CAN experts.

CAN FD was supposed to overcome the CAN limitation that the data transfer could not be faster than 1 Mbps. It also upgraded the protocol payload to 64 bytes (**Fig. 2**). The idea behind CAN FD is quite simple. When a single node is transmitting, the bit rate can be increased. Before transmitting the acknowledgment (ACK) slot bit, the nodes need to be re-synchronized.

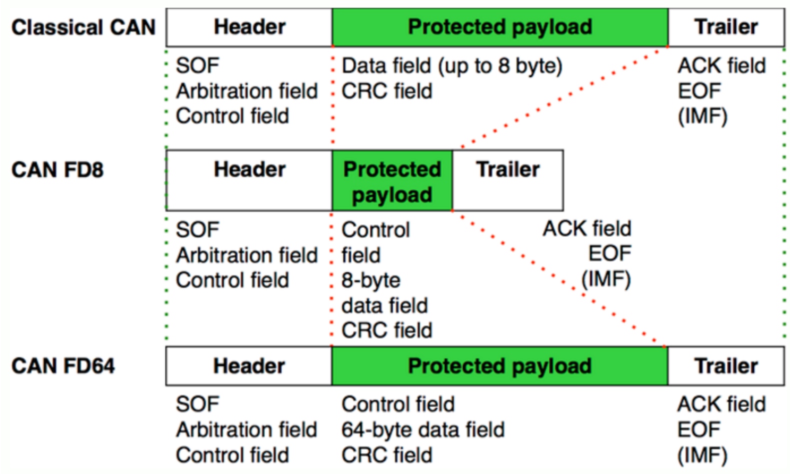

Fig. 2. Comparison of Classical and FD CAN Packet Structures

The new CAN FD standard is not fully deployed and most existing systems still run the classical CAN bus. Multi-master capability has been introduced to provide an evolutionary move to CAN FD.

It allows two CAN data-link layers to exist in the same network. This means that any node is allowed to access the bus at any time, if it is idle. When several nodes want to communicate at the same moment, the message with the highest priority wins the bus arbitration and secures the right to transmit. The system designer assigns a unique priority to each message. The CAN identifier (CAN-ID) part of the message indicates the priority. The lower the CAN-ID number, the higher the priority; thus, ‘0’ is the highest priority. One of the formerly reserved FDF (FD frame) bits is used to distinguish between classical and CAN FD data frames.

Even though the CAN FD contains improved features, the classical CAN and the CAN FD have the same header including the same arbitration field. It means that the CAN FD could get the same arbitration issues like the classical CAN.

2.3 Rotation ID

The CAN frame identifiers are the most important information for the non-destructive bitwise arbitration to provide a priority-based media access control (MAC) mechanism. Typically, all identifiers are fixed when the automotive engineers design the car. A related work in another technology field can help improve arbitration fairness.

Carrier Sense Multiple Access / ID Countdown (CSMA/IC) is another type of ad hoc collision-solving MAC protocol. The simultaneous medium access competition is solved by assigning a local unique ID to each node, which is a quite challenging and complicated problem.

Lee and Kim [2] demonstrated a distributed ID-assignment scheme for CSMA/IC. They used an “ID screen effect” (the ability to find the largest ID among contenders) to assign a new ID individually in each node. Additionally, the authors proposed a “well-arranged ID-rotation system” to address the ID starvation problem. As they mention, the proposed ID-rotation system cannot provide perfect fairness, but it distributes the medium access considerably more fairly. To measure the efficiency of the proposed methods, they defined two metrics: the medium utilization ratio and the ID assignment delay.

The Well-Arranged Rotation ID system uses rotated IDs for every medium-access contention. The ID rotation rules are as follows:

- Change the first (most significant) ‘0’ bit to ‘1’
- Set all previous bits to ‘0’
- If no ‘0’ bits exist, set all ‘1’ bits to ‘0’

If the binary form of an ID is “11001,” the result of the rotation becomes “00101.” Because the third bit is the first ‘0’ bit, it is set to ‘1’ and the first and second bits are set to ‘0’. **Fig. 3** illustrates an ID-rotation example for a three-bit set.

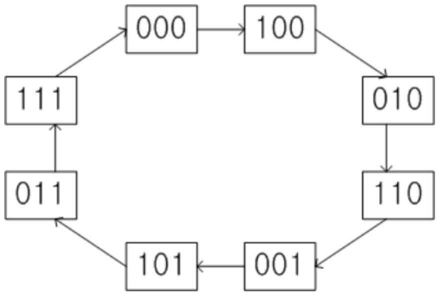

Fig. 3. Well-Arranged Three-bit ID-Rotation Example

3. NDBA-IR Method

The CAN messages’ timing behavior is influenced by several factors, including different message priorities and the type of queuing policy implemented by the CAN device drivers and communication stack.

NDBA (Non-Destructive Bitwise Arbitration) is a typical priority-based arbitration method implemented by the CAN protocol for transmitting messages on a network. The highest priority message of those sent from all nodes wins the arbitration, i.e., the right to transmit over the network. During the competition to win the arbitration, messages are sent on the network at the same time. Despite the simultaneous transmission, the NDBA avoids the collision and destruction of messages using the electrical mechanism between a dominant bit and a recessive bit [1]. A retransmission caused by the simultaneous transmission of two or more nodes would not be a consideration.

Fig. 4 shows an example of the effect of the conventional NDBA policy [3]. Node A is the highest priority node transmitting the highest priority messages, and node G is the lowest priority node transmitting the lowest priority messages.

According to the fixed priority assignment, the average end-to-end delay is proportional to the static priority order [4, 5]. The average end-to-end delay of node A is better than node B, which is better than node C, and so forth. Thus, the performance difference between node A and node G is quite large.

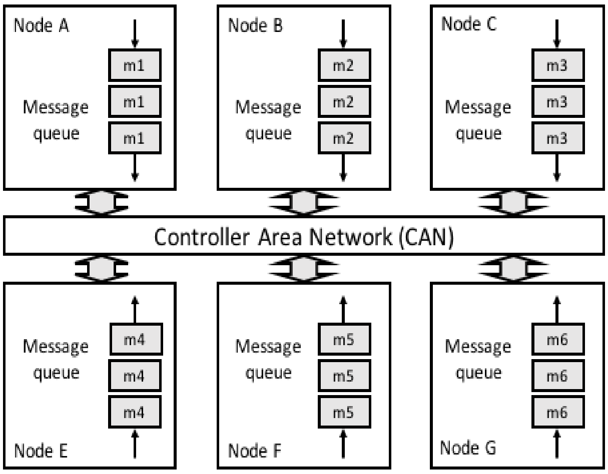

Fig. 4. Example of the conventional policy

This feature is not always the best solution because NDBA's performance evaluation shows a proportional distribution of priorities. There is a big difference between the highest priority node and the lowest priority node. It would be unfair for similar priority nodes to distribute messages unequally.

To build a non-destructive and fair networking system, the fixed-priority NDBA should be improved. **Fig. 5** shows an example of the NDBA-IR (ID Rotation) policy that provides fair message treatment.

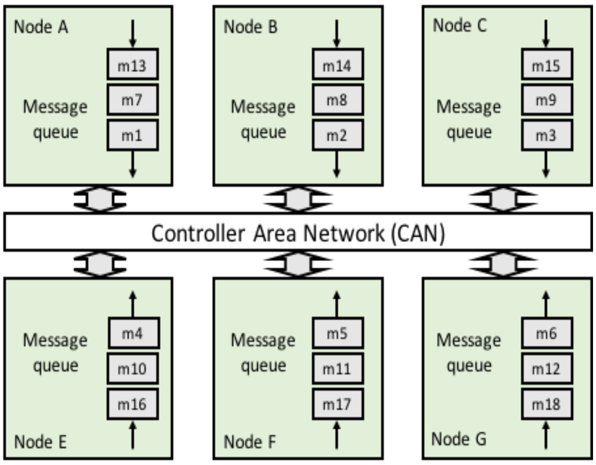

Fig. 5. Example of the NDBA-IR policy

The NDBA-IR node does not use fixed identification. As soon as a node transmits a message, it changes its ID to the next ID, which has the lowest priority. In **Fig. 5**, node A sends message m1 and node B waits with message m2. After sending message m1, node A prepares its next message m7 with the lowest priority and node B starts to send message m2. (Unlike **Fig. 3**, this is not a three-bit rotation, so it does not reset at seven.)

In **Fig. 5**, when node A finishes sending message m1, the other nodes try to send their messages. Thus, they cannot recognize whether another node is sending a message. The other nodes send their messages, although nodes C, D, E, F, and G soon stop sending. Like node A, node B changes its ID to the next ID, which is the largest number (lowest priority) among all nodes on the network.

Every node increases its ID after every successful transmission. Before long, the IDs are exhausted because of the arbitration-field length. Whenever a node's ID reaches the maximum number, it resets to its initial ID and the ID rotation restarts. The node obtaining the initial ID owns higher priority again.

If there is no reset process, the last node with the maximum ID becomes the overwhelming node with ID zero. Until the ID reaches quite a large number, the node monopolizes the network. The NDBA-IR allows every node to attain a fair performance.

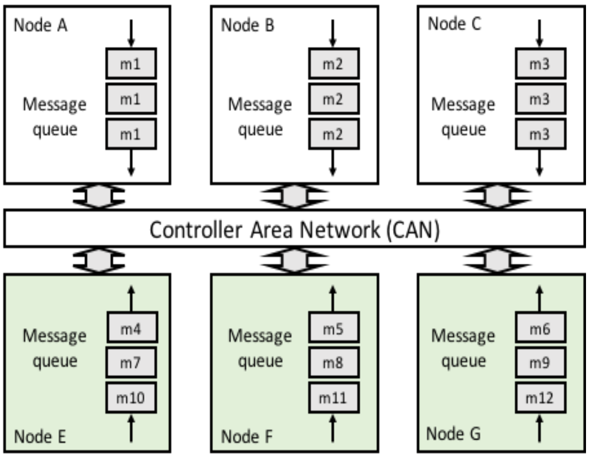

Fig. 6. Example of mixed policies

Fig. 6 shows the example to demonstrate the effect of a mixed policy consisting of the conventional fixed priority and the ID Rotation supporting fairness. Nodes A, B, and C are the fixed priority group, which would be high or low. The other nodes follow the NDBA-IR policy. The IDs of nodes E, F, and G should change within some ID range and reset their IDs whenever an ID reaches the maximum number. The mixed policy can support various network configurations.

Fig. 7 shows a flow chart of the NDBA-IR node's ID management module. The four dotted boxes at the left are data variables. At the beginning, every node sets them to constant values, e.g., an initial ID, the maximum ID on the network, and the number of nodes. The current ID is a variable that changes during operation.

Whenever a node fails to send a message, it waits for another node's transmission and tries again, using the same ID. If successful, the node updates its current ID by adding the number of nodes, or a bigger number.

If the new current ID is smaller than the maximum ID, the node's priority becomes the lowest one and the node makes way for the other nodes. Even though the node's priority is the lowest, it can send new messages when the network is idle; i.e., no other node is sending a message. The more messages the node sends, the lower priority it is assigned. Each node should yield to another node according to the number of transmissions. As a result, every node fairly earns a similar chance for transmission.

核心：初始给予一个固定的Id，每个报文发送一次，其报文的优先级便下降到当前设置的优先级最低的下一个等级，以此类推，直到Id全都被使用以后，每个报文均从初始Id开始循环继续发送。

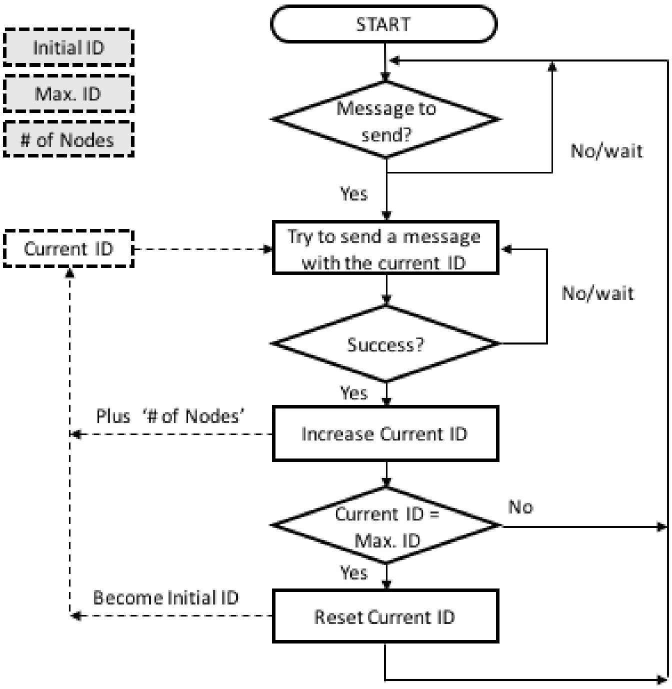

Fig. 7. Flow chart of an NDBA-IR node. Solid lines indicate the flow. Dashed lines indicate changes to a variable.

4. NDBA-IR Simulation

The performance was evaluated based on the average end-to-end delay on the network. The end-to-end delay, representing the latency of transmitting a message between a station and a receiver, is measured as the criterion. A typical vehicle end-to-end delay requirement is strict, and usually required to be less than 10 ms [6, 7].

The network topology is a bus topology, as in **Fig. 8**. Therefore, every message is transmitted as a broadcast message and every node receives every message on the network. A node consists of two modules, as shown in **Fig. 9**. 'ctl' is the NDBA-IR controller. 'srv' generates the messages and 'rcv' receives them.

To evaluate the NDBA-IR performance, the OMNeT++ simulation tool with the INET framework was used. The CAN controller simulation model was written and modified based on the simulation environment developed by Jun Matsumura et al. [8].

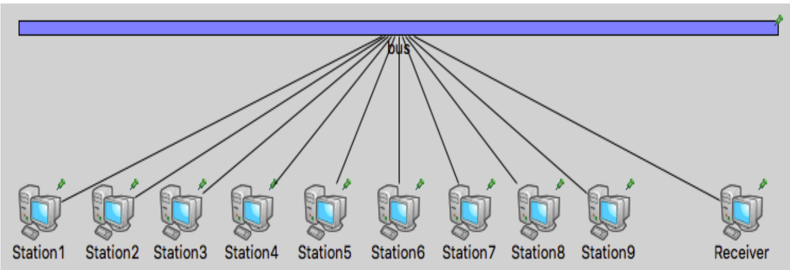

Fig. 8. Network configuration of the OMNeT++ simulation

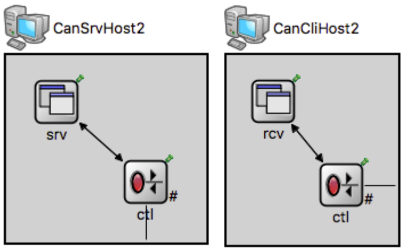

Fig. 9. Server and client host configurations

The simulation environment has a performance-evaluation assumption defined at [9]. Every node sends messages periodically with a random interval between adjacent messages. Once the message is sent, the node waits for a random time. As soon as the waiting timer expires, the node tries to send the next message. The data length is 8 bytes and the network bandwidth is 1 Mbps.

Two simulation configurations were considered. In the first, the average interval of every message is 4 ms, distributed exponentially. In the other, the average interval of every message is 2 ms, also with an exponential distribution. The first simulation means that the input load is about 25%; the other means that the input load is about 75%.

Fig. 10 and **Fig. 11** show the results of the two simulations. The Y-axis shows the end-to-end average delay between the station and the receiver in **Fig. 8**. The Y-axis data unit is μs. The X-axis means the node. In the result graphs, the striped bar means the conventional NDBA and the solid bar means the NDBA-IR.

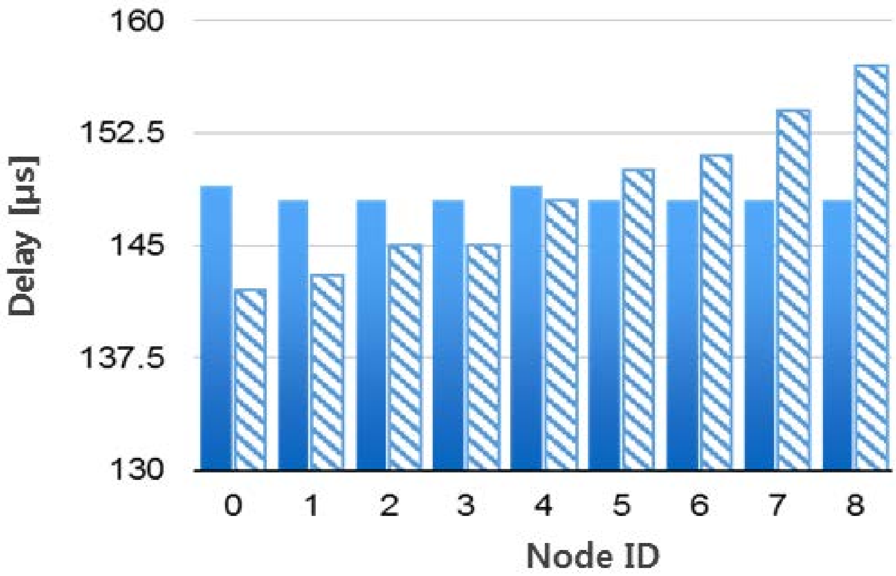

Fig. 10. End-to-end delay result with a light load

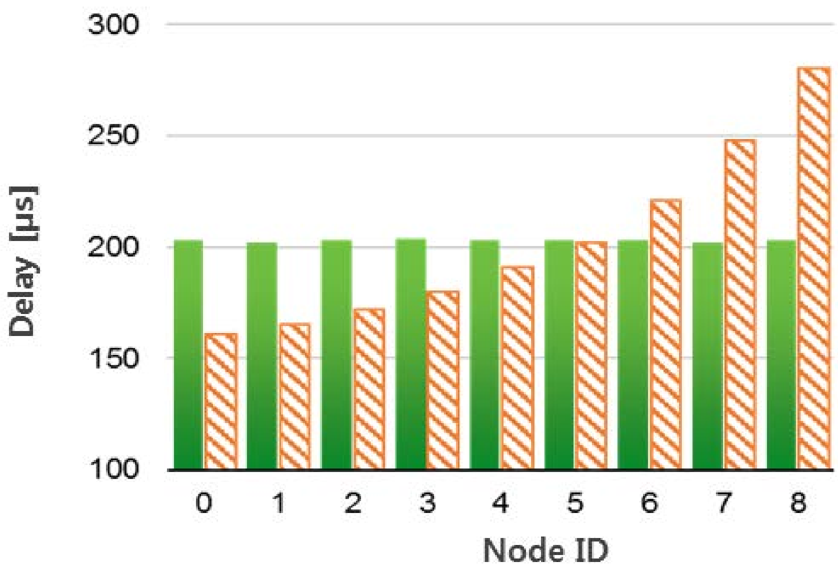

Fig. 11. End-to-end delay result with a heavy load

Regardless of the load, NDBA and NDBA-IR show similar patterns, but the two policies show different results. For example, NDBA shows a big difference between the lowest priority node 8 and the highest priority node 0. The results of both **Fig. 10** and **Fig. 11**, of course, show the same pattern. However, the results of the NDBA-IR show a very small difference between the nodes.

5. Experiment

This section presents the real physical device implementation of the proposed idea. For our experiment, we used an Atmel-based AT90CAN128 microcontroller. It is a high-performance, low-power Atmel 8-bit Advanced Virtual RISC-based microcontroller with 128 KB in-system programming flash memory, 4 KB electrically erasable programmable EEPROM, 4 KB SRAM, 53 general-purpose IO lines, and a CAN controller. More details regarding this chip can be found at [10].

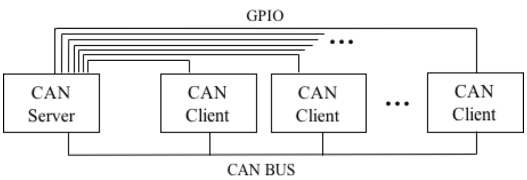

Fig. 12. Experimental environment / architecture

The experiment was performed with eight AT90CAN128 microcontroller-based devices; one behaved as a server and the remaining seven nodes behaved as clients. A simplified experimental architecture is illustrated in **Fig. 12**. **Fig. 18** shows a photo of the experiment environment.

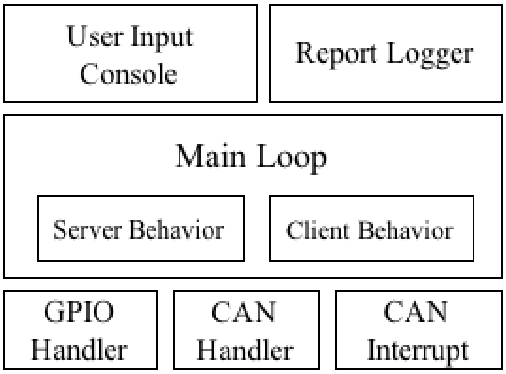

Fig. 13. Server/client-node software architecture

The firmware architecture was designed to be reconfigurable by console at runtime. Since the AT90CAN128 microcontroller only runs an 8-bit AVR core, deploying it with an OS would not be an appropriate solution. The idea was implemented in firmware with a single solution. The single binary solution could be deployed on each ECU device and initialized using a universal asynchronous receiver/transmitter (UART) communication channel.

The simplified software architecture of the experimental node is depicted in **Fig. 13**. At the bottom, the GPIO and CAN related interrupt and handler codes are displayed. On top, the user input console and logger output codes are shown. The core logics for various types of experiment are deployed in the main loop section. The main loop section can differ depending on the experimental node configuration and behave as a server or client.

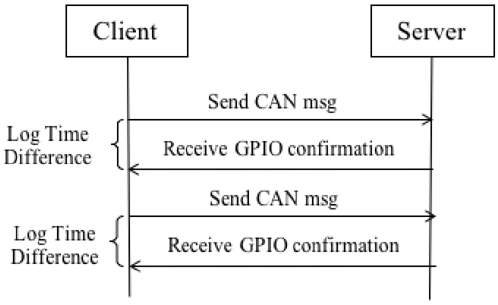

Fig. 14. Sequence diagram of message exchange

The overall behavior of the system is quite simple, and illustrated in the sequence diagram in **Fig. 14**. First, a client device generates a CAN message that contains its ID number. It fixes the time stamp when sending the message to the server. When the server receives the CAN message, it extracts the ID number and triggers the proper GPIO wire. Our experimental system has eight GPIO wires and each wire is connected to a CAN client with a one-to-one connection.

For example, the GPIO-2-connected client-node device defines its ID number as 2 and sends a CAN message that contains a 2 as its second byte value. The server simply extracts the ID information from the received packet and changes the GPIO-wire value from low to high, or vice versa. When the GPIO-wire value (voltage) is changed, the client checks the timestamp and calculates the difference. The timestamp difference is logged and saved for future analysis. Then, the next CAN message is generated and sent to the server. At this step, the ID number of the client device would differ depending on the experiment type. For our study, we defined three types of experiment:

1. No grouping, without ID rotation
2. Two groups, with ID rotation
3. No grouping, with ID rotation

The list of experiment types was not selected randomly. It is important to show the performance difference in each type of experiment.

Table 1. Throughput delay for each node [ms]

Node ID	Waiting Interval [ms]		
	100	10	5
1	1.000	1.168	1.981
2	1.005	1.260	2.031
3	1.005	1.267	2.412
4	1.006	1.277	361.950
5	1.005	1.365	580.831
6	1.007	1.412	2225.956
7	1.011	1.483	3261.228

It is difficult to load the CAN bus with 1-Mbps throughput, which is why we reduced the throughput to 125 Kbps. To start an experiment between every CAN message, we set a waiting interval. To see the degradation, various waiting-interval values were tried. Table 1 provides a list of value examples. It shows that when the waiting interval is large enough, the delay among the nodes stays almost the same. When the waiting interval becomes 5 ms, the larger-ID numbered nodes (with lower priority) have a longer delay than the smaller-ID numbered nodes. Five ms was defined as a threshold for our experimental setup.

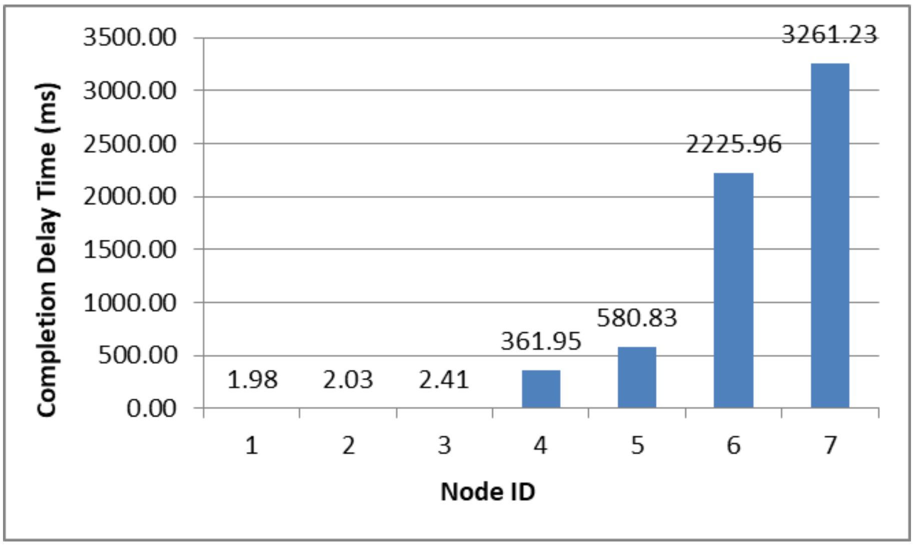

Fig. 15. Experimental results: no grouping, without ID rotation, 5-ms waiting interval

Fig. 15 shows the 5-ms waiting interval results from Table 1. It shows the large detailed delay difference between the nodes. A simple calculation can show that it is impossible to obtain fairness among the nodes because of the bandwidth size.

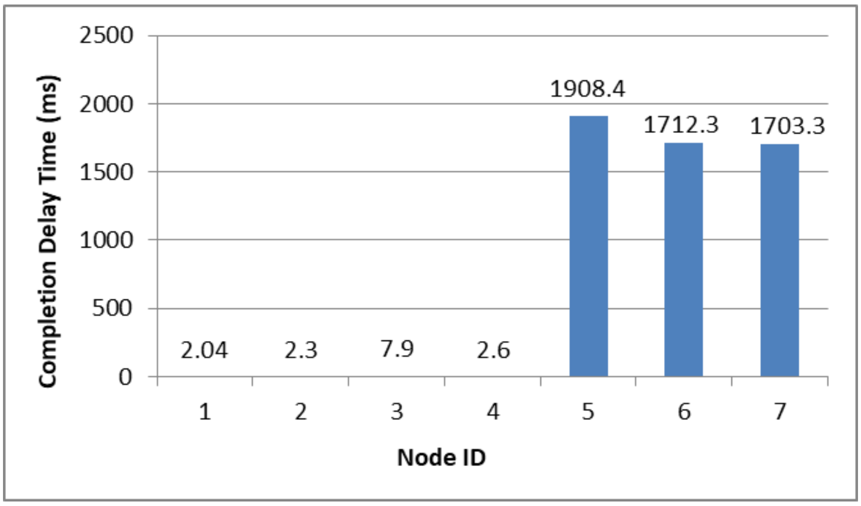

Fig. 16. Experimental results: Two groups with ID rotation, 5-ms waiting interval

To reach some level of fairness, we implement grouping logic with the rotation. **Fig. 16** illustrates a two-group example.

The first group includes nodes 1–4 and the second group contains nodes 5–7. The first group has higher priority than the second group. To implement this grouping and rotation logic, the nodes' ID numbers were distributed.

The first group's address range goes from 1–1000 and the second group includes 1001–2000. The output result in **Fig. 16** depicts that the three last nodes were normalized among themselves. Moreover, the worst delay time was reduced from 3 s to 1.9 s.

As a real, practical use of this idea, we can see that transmission-related device nodes could be merged into the first group, where the response time should stay about 2 ms. The other, window-control type of device node could be allowed to have a 2-s delay. We are considering worst-case scenarios when the network throughput is used maximally.

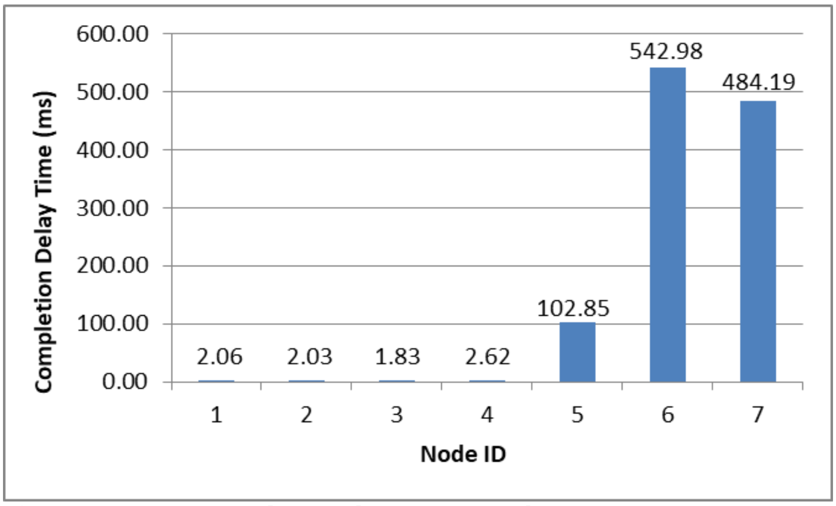

Fig. 17. Experiment results: Single grouping, with ID rotation, 5-ms waiting time

Sometimes, a 2-s delay for window control in the above situation may not be acceptable to the user. To reduce this delay, we continued our experiment and finalized it with a single grouping that distributed the waiting delay fairly among the other nodes. As illustrated in **Fig. 17**, the single grouping with ID rotation decreases the delay on devices with small ID numbers. This time it decreased to 400 ms. Our theoretical assumption and simulation in OMNeT++ gave us an absolute fairness result. However, the real device deployment shows that complete fairness is almost impossible to reach because of the hardware limits of the CAN controller chip.

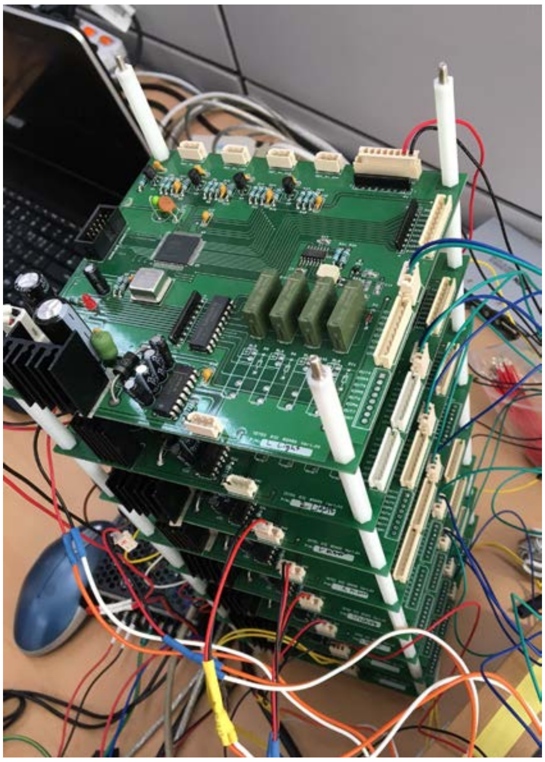

Fig. 18. Test environment

6. Analysis of Results

The main purpose of NDBA-IR is to maximize the total utilization within some system-requirement constraints. **Figs. 10** and **11** show simulation results. **Figs. 15–17** show experimental values. In **Fig. 10**, we assume that the system requirement for the average end-to-end delay is 150 μs. The two striped bars with the lowest priority at the right of the graph exceed the constraint limit. This means that the CAN bus can only use seven nodes; the last two nodes do not meet the requirement. **Fig. 11** shows similar results, but all solid bars meet the requirement; thus, the CAN bus can use all nine nodes. **Fig. 15** and **Fig. 17** show cases similar to **Figs. 10** and **11**. In **Fig. 15**, nodes 6 and 7 show a much bigger latency than in **Fig. 17**.

Fig. 15 shows a relation between the latency and priority that is similar to the simulation results. A high priority node can send a message immediately, but a low priority node must wait for the higher priority nodes to finish their transmissions.

Figs. 16 and **17** show different results for the low priority nodes' performance. The latency of **Fig. 17** is better than **Fig. 16** because the low priority nodes in **Fig. 17** will earn a chance to transmit through their ID rotation. In contrast, the low priority nodes in **Fig. 16** never have a good priority ID because the IDs are compartmentalized into two groups. The higher priority group never allows the lower priority group any chance to transmit a message prior to the higher priority group.

An implementation analysis is required to explain the results in **Fig. 17**. As was mentioned above, the throughput of the CAN bus was reduced to 125 Kbps, which is equal to 874 frames per s [11], which is approximately one frame per ms. Hence, when the waiting interval is 5 ms, it is impossible to fit seven packets on a single bus.

Absolute fairness was not reached by ID rotation because of an implementation tradeoff. Every time a node tried to send a frame, it checked the availability of the CAN bus. If the CAN bus was in use by other nodes, the node waited for a certain amount of time. While the node waited, other nodes were able to use the network. Thus, nodes that started with higher priority IDs were able to send packets multiple times, while their priority kept decreasing. When a collision finally happened, the high priority nodes won, and sent their frames. The high, tall delays moved around over the course of the experiments. Sometimes the highest delays shifted from nodes 6 and 7 to nodes 1 and 2. The difference of the total, summarized completion delay between NDBA-IR and the static NDBA decreased 83% (from 3261.23 ms to 542.98 ms). It proves that the ID-rotation concept works and can be applied to real field CAN-bus applications.

7. Conclusion

The paper proposed NDBA-IR (Non-Destructive Bitwise Arbitration-ID Rotation), which improved the fairness of a CAN bus network. NDBA-IR was described and examples were given. The simulation results and experimental measurements were presented to demonstrate the end-to-end delay in various node configurations. From this analysis, NDBA-IR decreased the difference between the high priority nodes' performance and the low priority nodes' performance, thereby increasing the total utilization and fairness.

Acknowledgments

This work was partly supported by a Korea Agency for Infrastructure Technology Advancement (KAIA) grant funded by the Ministry of Land, Infrastructure, and Transport (MOLIT) (No. 15RTRP-B109166-01, Research on the parts and modules standardization of rolling-stock) and a Korea Evaluation Institute of Industrial Technology (KEIT) grant funded by the Ministry of Trade, Industry, and Energy (MOTIE) (No. 10050293, Video Integration Transmission Network System Development for Aircraft based on 10GbE). This paper was supported by Konkuk University in 2016 as well.

References

[1] Robert Bosch GmbH, "CAN Specification, version 2.0," September 1991. Article (CrossRef Link)

[2] K. S. Lee and C. H. Kim, "A Simple Distributed ID Assignment Scheme for CSMA/IC using Well Arranged Rotation ID in Ad Hoc Networks," in Proc. *International*

Conference on Advanced Information Networking and Applications, pp. 624–631, 2009. Article (CrossRef Link)

[3] S. Mubeen, J. Mäki-Turja, and M. Sjödin, “Extending Worst Case Response-Time Analysis for Mixed Messages in Controller Area Network With Priority and FIFO Queues,” *IEEE Journals and Magazines*, vol. 2, pp. 365–380, 2014. Article (CrossRef Link)

[4] H. Zeng, M. D. Natale, P. Giusto, and A. Sangiovanni-Vincentelli, “Using Statistical Methods to Compute the Probability Distribution of Message Response Time in Controller Area Network,” *IEEE Trans. on Industrial Informatics*, vol. 6, no. 4, pp. 678–691, November 2010. Article (CrossRef Link)

[5] J. Hao, J. Wu, and C. Guo, “Modeling and Simulation of CAN Network Based on OPNET,” in *Proc. of Int. Conf. on Communication Software and Networks*, pp. 577–581, 2011. Article (CrossRef Link)

[6] H. T. Lim, L. Volker, and D. Herrscher, “Challenges in a future IP/Ethernet-based in-car network for real-time applications,” in *Proc. of Design Auto. Conf.*, pp. 7–12, June 2011. Article (CrossRef Link)

[7] J. Sommer, S. Gunreben, F. Feller, M. Kohn, A. Mifdaoui, D. Sass, and J. Scharf, “Ethernet – A survey on its fields of applications,” *IEEE Commun. Surveys & Tutorials*, vol. 12, no. 2, pp. 263–284, April 2010. Article (CrossRef Link)

[8] J. Matsumura, Y. Matsubara, H. Takada, M. Oi, M. Toyoshima, and A. Iwai, “A Simulation Environment based on OMNeT++ for Automotive CAN-Ethernet Networks,” in *Proc. of Fourth Int. Workshop on Analysis Tools and Methodologies for Embedded and Real-Time Systems*, pp. 1–6, July 2013. Article (CrossRef Link)

[9] P. Park, M. Kim, and J. Yoon, “Performance Evaluation of CAN Arbitration by Wrap-Around ID Assignment,” *KICS Conf.*, pp. 1–2, 2016. Article (CrossRef Link)

[10] http://www.atmel.com/devices/AT90CAN128.aspx. Article (CrossRef Link)

[11] https://electronics.stackexchange.com/questions/121329/whats-the-maximum-can-bus-frame-message-rate-at-125-kbit-s. Article (CrossRef Link)

Pusik Park received the M.Sc. degree in information and communication engineering from the Korea Aerospace University, Republic of Korea, in 2002. Since then, he has been with Korea Electronics Technology Institute, where he is currently the managerial researcher. His current research interest includes industrial deterministic network and related issues.

Rustam Rakhimov Igorevich received his Ph. D. degree from the Distributed Multimedia Lab, Konkuk University, Seoul, Republic of Korea in 2016. He is currently working as a researcher at Korea Electronics Technology Institute. His current research interest includes IoT and IoT Security related issues.

Jongho Yoon received his Ph. D. degree from the Korea Advanced Institute of Science and Technology, Republic of Korea in 1990. He has been with the Korea Aerospace University, where he is currently a Senior Professor. His current research interest includes telecommunication and deterministic Ethernet related issues