

RESEARCH ARTICLE

Seamless CAN: A Novel Fault-Tolerant Algorithm and Its Modeling

JONG MYUNG RHEE¹, (Member, IEEE), IBRAHEEM R. A. ALTAHA¹,
DUC N. M. HOANG², (Graduate Student Member, IEEE), DONG HWAN KIM²,
JIN SEOK YANG², AND SANG YOON PARK², (Member, IEEE)

¹Department of Information and Communications Engineering, Myongji University, Yongin 17058, South Korea

²Department of Electronic Engineering, Myongji University, Yongin 17058, South Korea

Corresponding author: Sang Yoon Park (syPark@mju.ac.kr)

This work was supported by the National Research Foundation of Korea (NRF) funded by the Korean Government (MSIT) under Grant 2020R1A2B5B02002201 and Grant 2021R1F1A1050040.

ABSTRACT Because there is always a demand for robust fault-tolerant automotive networks, recent works have focused particularly on replacing controller area networks (CANs) with more reliable protocols for time-critical bandwidth-demanding in-vehicle network (IVN) systems. However, as the automobile industry normally relies on proven technologies, an interim solution to fault-tolerance capability for CAN systems is still essential during the transition phase to other protocols. In this paper, we propose a novel fault-tolerant algorithm for the CAN protocol that we refer to as *seamless CAN*. Its operating concept is based on a high-availability seamless redundancy (HSR) implementation inside the vehicle network's ring structure, and the fault tolerance is achieved by encapsulating the CAN frame inside the HSR frames and sending them in the ring topology. The simulation results showed that seamless CAN provided better performance regarding no frame loss in the case of link failures and up to ten times lower number of error occurrences.

INDEX TERMS High-availability seamless redundancy (HSR), fault-tolerant CAN, seamless CAN, in-vehicle network (IVN).

I. INTRODUCTION

Ever since the creation of the first successful internal combustion designs, vehicle technology (VT) has developed primarily in the direction of improving the performance of automobiles of a similar design, including faster speeds and better fuel efficiency. But with the emergence of the Fourth Industrial Revolution, that trend has changed drastically. VT is innovating with the integration of cutting-edge information and communication technology (ICT), as can be seen in electric vehicles, self-driving automobiles, and flying car trials [2]. As a result, ICT industries have now begun to play an important role in the automotive industry instead of an ancillary one. For example, unlike in the past when motorized vehicles served merely as a means of transportation, nowadays, people can enjoy various forms of entertainment, such as karaoke and video playback, while on board. Google and

Apple have formed partnerships with existing automakers to manufacture automobiles with numerous ICT advances [3] so access to multimedia resources by passengers and a driver (when appropriate) is merely a few touches away. The in-vehicle infotainment system on Tesla models is so powerful that it is even capable of delivering a smooth state-of-the-art gaming experience [4]. More importantly, thanks to advanced driver-assistance systems (ADAS), recent commercial vehicles are able to operate with little to no human input [5]. However, in order to realize this ICT technology, the number of electronic control units (ECUs) and various sensors (e.g., LIDAR, proximity, and airflow sensors) embedded in vehicles has increased dramatically. Considering the current trend, advanced automobiles are expected to eventually contain up to hundreds of ECUs and sensors [6].

Normally, electronic devices in the vehicles exchange information via a shared in-vehicle network (IVN). Current IVNs have a heterogeneous structure, which is a mixture of

The associate editor coordinating the review of this manuscript and approving it for publication was Giovanni Pau¹.

various protocols, such as controller area network (CAN), local interconnect network (LIN), FlexRay, and media-oriented systems transport (MOST). Such a structure makes a vehicle's design and maintenance work more difficult and causes a significant rise in its weight. This is because additional devices then require being equipped for inter-protocol switching, resulting in a complicated wiring harness. In addition, although CAN shows dominant usage over other protocols, in terms of meeting the growing demand for higher data rates, it has reached its limits in terms of speed and bandwidth [7], [8]. These limitations mean that in-vehicle network failures and interruptions directly linked to fatal accidents might occur more frequently. In reality, as more and more electronic devices are being placed inside vehicles, the number of accidents due to sudden unintended acceleration (SUA) and electronic malfunctions has increased [9].

To address the aforementioned issue of conventional IVN protocols, leading industries have begun to adopt other alternatives, and due to its popularity and high compatibility, Ethernet is one of the prospective candidates [10]. In fact, standard Ethernet also has built-in mechanisms for redundancy in the case of network failure. For example, the rapid spanning tree protocol (RSTP) builds a logical topology for the network and reserves redundant links for faulty network situations. However, IVN systems usually require zero recovery time, while RSTP is subject to a certain amount of delay to reconstruct the network links between devices. Therefore, several approaches have been proposed to enhance the reliability of Ethernet-based networks, including the IEEE 802.1CB standard by Time-Sensitive Networking (TSN) Group [11]. In this standard, redundancy is realized by making copies of the same frame and sending them over disjointed paths, which is similar to the operation concept of high-availability seamless redundancy (HSR) [12]. It is likely that IEEE 802.1CB will be the next-generation protocol for network systems inside vehicles in the future. However, HSR has been commercialized and is now readily available on the market; therefore, it is more convenient to take advantage of HSR for the time being.

As mentioned earlier, traditional in-vehicle networking protocols (e.g., CAN, FlexRay, and LIN) are expected to be replaced by novel Ethernet-based approaches due to their higher bandwidth performance, fault tolerance, and network reliability. However, because CAN has long been used as a basic protocol for IVN since its development, a large number of automobile plant facilities have been configured to work with this protocol. Therefore, it would possibly take a long time for the current IVN to evolve into a complete Ethernet-centered network, thus suggesting the need to resolve CAN's existing drawbacks. In this paper, we propose a new fault-tolerant algorithm for CAN referred to as *seamless CAN*, which applies HSR's redundancy mechanism to CAN as an interim solution during the transition period to next-generation IVNs. The main contributions of this paper are summarized as follows:

- A novel network structure is introduced in which the CAN bus responsible for the communication between CAN nodes is replaced with an HSR-based ring topology. Our proposed scheme guarantees zero recovery time in the event of message errors or link failures in a CAN-based network. Because every CAN frame is encapsulated inside a seamless CAN frame, the entire network system becomes more reliable without any modification of the CAN frames, hence improving its compatibility level;
- We carried out validation to ensure that our CAN models would work in accordance with CAN modules implemented in real-world applications. By comparing bit-error-rate (BER) performances adapted from experiments in [13] with our results, we were able to conclude that our models were analogous to those in the real world and were safe for further simulation procedures;
- We conducted an assessment of our proposed scheme against the conventional CAN protocol with several simulation scenarios, featuring failover and BER performance simulations. The preliminary results showed that seamless CAN displayed no loss of frames in the case of link failures and significantly better BER performance, indicated by a lower number of bit error occurrences.

It should be noted that this work is an extended version of our presented conference paper [1], and we have developed the concept of seamless CAN in a much more rigorous manner. In fact, the conference paper is a brief introduction to the basic working principles based on the ring-topology network of HSR protocol. However, in this journal version, we introduce the novel concept of seamless CAN node, which describes in detail the essential components of the node, such as different interface types for both CAN and HSR sides. We also provide in detail the operation concept of seamless CAN node to thoroughly demonstrate how the CAN bus can be transformed into the ring architecture using an illustrative diagram. Finally, we also describe how a conventional CAN bit signal can be encapsulated inside an HSR frame for implementation work.

In addition, we provide a mathematical formulation for the ring-structure network to prove how this type of topology is able to improve the successful transmission rate compared to the conventional CAN bus. We also implement the proposed scheme based on the OMNeT++ framework and validate the simulation model based on previous real-world work to ensure the reliability of our simulation results. In this journal version, we further conduct thorough performance evaluation work with numerous different scenario settings. Finally, compared to the conference paper version, we conduct an extensive literature review of state-of-the-art fault-tolerant solutions for in-vehicle network systems. Relevant background knowledge is also provided to the readers for a better understanding of the fundamental concepts behind how we formulate, design, and realize the seamless CAN mechanism.

The rest of this paper is organized as follows. Section II reviews the related work in the literature. In Section III, we study the conventional CAN and HSR protocols and then propose our seamless CAN scheme in detail. The modeling procedures of CAN and Seamless CAN protocols are presented in Section IV, which also includes the hypothesis and scenarios set up for the simulation. In Section V, we present several simulation scenarios to evaluate our proposed scheme, and the results are extensively discussed. Finally, Section VI concludes this paper and proposes future work.

II. RELATED WORK

A few works have reported designing and achieving fault tolerance for the IVN protocols especially CAN because it is widely used in many industrial areas yet is vulnerable due to many limitations [14]. One of which, as pointed out by Short and Pont [15], is that CAN is a single-bus protocol, and any bus damage may lead to a failure of the whole IVN. For this reason, IVNs with CAN based on multi-bus architecture have been intensively studied in the literature. For instance, Ferreira et al. [16] repurposed the FFT-CAN protocol by presenting a network architecture with duplicated broadcast bus and bus guardians. In addition, the replication of the master node and its synchronization protocol were also proposed to ensure similar scheduling behaviors between the primary and backup masters. However, the extended amount of synchronization process delay might not be suited for time-critical IVNs.

Employing a similar dual redundancy mechanism, Xiang-Dong et al. [17] proposed a dual CAN-bus network architecture with its redesigned controllers based on a field-programmable gate array (FPGA) chip. Due to the fact that redundancy management is handled by hardware logic circuits, this scheme is probably applicable to systems with real-time and high-reliability requirements. In addition, Wang et al. [18] presented a combination of analytical and hardware fault-tolerant redundancy solutions. By using a method of detecting resistance changes and consequently locating the faulty parts of the entire network, the proposed scheme further improves the robustness of CAN bus. Interestingly enough, a simple yet effective solution was proposed by Rufino et al. [19]. In fact, replicated CAN bus approaches are not considered particularly cost-effective, so this approach realizes redundancy in the physical media of only one channel. To this end, a dedicated device is used in conjunction with the CAN controller to implement a voting mechanism on available redundant media interfaces.

As an IVN architecture generally involves manifold networking protocols for various on-board applications, there is also a need for fault-tolerant gateways that govern inter-protocol communications. Kumar et al. [20] proposed what they described as a “penta controller gateway” design that includes four dedicated CAN and LIN controllers, which monitor and send the health status of corresponding buses to

the single main controller. Experiments examined the low-budget setup of the proposed solution, which demonstrated its applicability.

In addition to the CAN bus, researchers have also developed other protocols aimed at providing fault-tolerant IVNs. Tien and Rhee [21] proposed a novel HSR switching node (SwitchBox) that is capable of forwarding and filtering unicast traffic, thus preventing unnecessary frames from circulating in HSR networks. This approach allows the use of HSR in any topology (e.g., ring, mesh, star) as well as significantly improving traffic performance in an HSR network. In fact, recent work by Kim et al. [22] proposed a latency performance study of an HSR-based IVN that was applied to network structure arrangements based on different real-world vehicular function domains. Simulation results demonstrated the appropriate numbers of ECUs for each domain to meet the latency constraint. Therefore, HSR was demonstrated to be efficient and reliable enough for the strict requirements of IVNs in many Ethernet-based control systems.

Another research work worth noting features the fault tolerance enhancement of the traditional CAN protocol with the aid of additional ZigBee communication functionality as presented by Ro et al. [23]. In the proposed design, each ECU, in addition to being connected via a shared CAN bus as usual, also supports additional data transmission using a ZigBee module. In the event of network faults, disconnected ECUs will take advantage of ZigBee to continue to communicate with other ECUs with no interruption. Experimental results have shown that wireless transmission duration does not affect the actuator maneuvering activity, and thus, the approach’s performance was verified for critical control systems.

However, none of these works have taken into consideration the fact that the automotive industry mainly relies on proven technologies, and the process of using and transitioning from current IVN architectures to advanced ones seems to be taking a large amount of time. Therefore, as mentioned earlier, we suggest adopting a provisional resolution in the interim.

III. BACKGROUND AND PROPOSAL

In this section, we first provide some background knowledge related to the CAN and HSR protocols. Next, our proposed approach is introduced and thoroughly demonstrated.

A. CAN OPERATION

The development of the CAN serial communication bus [24] extends back to as early as 1983; it mainly provides stable communication among an increasing number of electronic devices (e.g., sensors, controllers, and actuators) inside road vehicles. In addition to various vehicle uses [25], CAN has also been adopted in many other industrial areas, including agricultural equipment, elevators, and lighting control systems. As a globally standardized protocol, CAN

SOF (1 bit)	Identifier (11 bits)	RTR (1 bit)	Control (6 bits)	Data (0-64 bits)	CRC (16 bits)	ACK (2 bits)	EOF (7 bits)
(a)							
SOF (1 bit)	Identifier (29 bits)	RTR (1 bit)	Control (6 bits)	Data (0-64 bits)	CRC (16 bits)	ACK (2 bits)	EOF (7 bits)
(b)							

FIGURE 1. Two CAN frame formats, namely (a) standard CAN frame and (b) extended CAN frame.

is responsible for a considerable reduction in the cabling and wiring in car development.

Currently, there are four frame types for CAN operation: data, remote, error, and overload. All of them support two different types of frame formats: a standard and extended version. While the former version supports an 11-bit identifier, the identifier field for the extended one is 29 bits in length; recent CAN controllers are able to send and receive both frame types for different network configurations. The details of the two frame formats are shown in Fig. 1. It should also be noted that in the two formats, the identifier in either length defines the priority of the frame (or its sender node), which is one of the important parts of the CAN basic operating characteristics, namely *arbitration*.

Because nodes in a CAN network are connected to one another via a shared two-wire bus, potential conflicts might happen when there is more than one node trying to transmit frames at the same time. Therefore, CAN has a bit-wise arbitration mechanism that compares the identifier of each sender to allow only one transmitter after the arbitration phase: the lower the identifier of a node, the higher the priority of its frame. In addition, as there is no destination address field in the CAN frame, all the nodes in the shared CAN bus have to listen to the same traffic and select relevant frames instead. While the CAN arbitration scheme is useful in preventing conflicts, a node with high priority might keep sending its frames, consequently causing large delays for lower-priority nodes as they refrain from transmitting. To be able to operate without faults, all nodes in the same network are supposed to be able to sample every single bit on the CAN bus. However, due to noise and other external factors, certain means of synchronization are required.

In addition, the built-in error handling function enables CAN networks to maintain high data integrity because every frame is received and checked by all the nodes. If any errors are detected within a frame, it will not be accepted, and a corresponding error frame is generated by the receiving node, leading to the frame being resent afterward [26]. However, the CAN operation is unsecured and security implementation is left to higher-level layers.

B. HSR OPERATION

The HSR protocol is standardized by the International Electrotechnical Commission (IEC 62439-3 Clause 5 [12]) as a redundancy protocol for switch Ethernet networks. HSR is particularly suited for time and data critical systems that

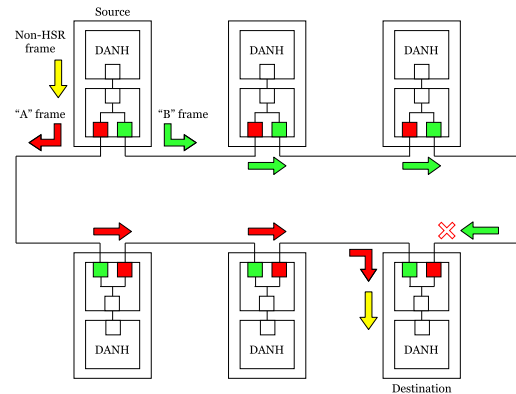


FIGURE 2. HSR example of unicast traffic in a single-ring topology.

demand high availability and zero switchover time in the case of network faults. Therefore, HSR is popular, and its use can be seen in electrical substation automation, transportation, and power inverters. Because HSR operation relies heavily on short frame-processing (i.e., forwarding or discarding) latency, it usually requires a hardware implementation (e.g., FPGA) for switching functions inside HSR devices.

Unlike the parallel redundancy protocol (PRP) [12] that duplicates an entire network in order to create a second physical path from source to destination, HSR only requires a single network to achieve redundancy. In detail, HSR mainly employs a ring topology although it can be applied to other network structures (e.g., connected rings or mesh topologies), provided that there are always two paths between every pair of nodes. An example of unicast traffic in an HSR single-ring topology is shown in Fig. 2.

It can be seen that each node in the network is connected to two nearby nodes, together forming a ring structure. When a “source” node sends a frame, it inserts two copies of it (i.e., the “A” and “B” frames) into the ring via two different Ethernet ports. The “destination” node, in turn, receives the “A” frame and passes it to the upper layer, while simply discards the later-arriving “B” frame. The replica detection method can be realized using the source address and a sequence number inside every HSR frame. These two fields, together with the destination address, HSR tag, payload, and checksum fields form the structure of an HSR frame as illustrated in Fig. 6a. Thanks to this redundancy mechanism, there is virtually no stoppage in network activity even when there is a node or case failure. Further detailed principles and the implementation of HSR can be found in [27] and [28].

Despite the fact that HSR is useful for network systems with low-latency requirements, its excessive redundant traffic might be a major limitation. That is to say, even in a situation where the network is not faulty, duplicated and circulated frames inside the HSR rings might impair the network performance by reducing the actual available bandwidth for applications. Therefore, there are a large number of techniques that work on reducing unnecessary network traffic [29], [30].

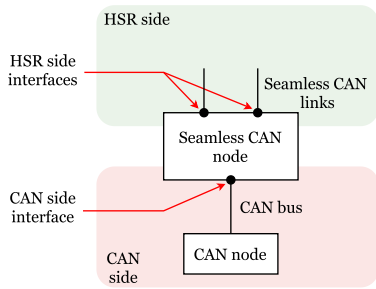


FIGURE 3. Seamless CAN node's interfaces.

C. SEAMLESS CAN

In this paper, we propose a fault-tolerant architecture based on an HSR ring topology for an IVN system. Our architecture is also backward compatible with CAN-based systems because each traditional CAN “bit” from its source node is encapsulated inside an HSR frame (i.e., it becomes the payload of that HSR frame), and its copies are then inserted into the ring, following the HSR concept. It should be noted that because we are replacing the CAN bus with an HSR ring and the CAN controllers inside CAN nodes transmit the bits serially to the bus, what is actually stored in each HSR frame is one single bit (i.e., either a “0” or “1”) instead of an entire CAN frame. To achieve this task, we introduce the concept of a “seamless CAN node” with three different interfaces in order to communicate with the CAN node and its two neighboring “seamless CAN nodes,” as depicted in Fig. 3.

Fig. 4 shows the operational concept of the seamless CAN node. When a bit is received from the CAN side’s interface as shown in Fig. 3, the seamless CAN node will embed this information within the payload portion of an HSR frame and send it off in two directions via two different interfaces on the HSR side. In addition, when a new frame arrives at either interface on the HSR side, the sequence number of that frame will be registered in the node memory, and the bit information will be taken out and delivered to the CAN node via the interface belonging to the CAN side. Similar to HSR, frames received from the HSR side will be dropped if they hold a sequence number that is already in the node’s record. This concept of work provides the following fault tolerance solutions:

- 1) Message fault tolerance when a frame error occurs;
- 2) Link fault tolerance when a link failure occurs.

Fig. 5 shows an example of seamless CAN network topology under two different environments. In Fig. 5a, the network activity under a healthy environment is depicted. It is seen that after the CAN signal is encapsulated inside the seamless CAN frame and sent in two copies, the frame transmission activity is similar to HSR’s operation. In contrast, there is a link failure in the network shown in Fig. 5b. Although only one copy of the frame reaches the destination node, the CAN node still operates without any interruptions as activities from the seamless CAN side are transparent to nodes on the CAN side.

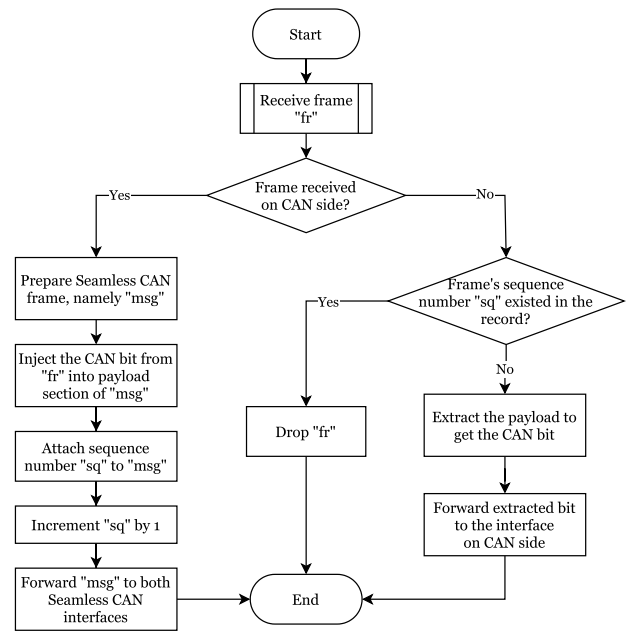


FIGURE 4. Seamless CAN node's concept of operation.

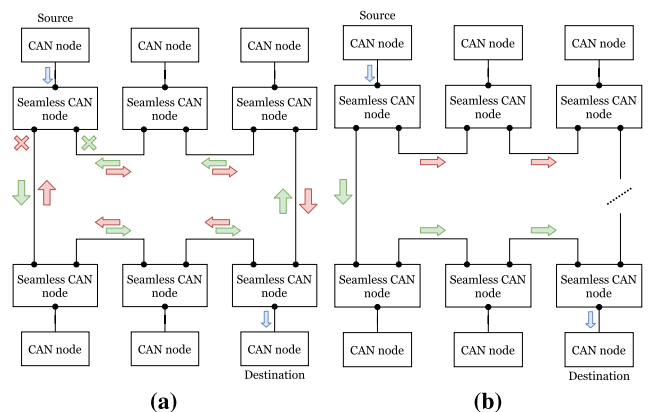


FIGURE 5. Single-ring seamless CAN network in two environments, namely (a) healthy environment case and (b) network failure case.

The differences between CAN, CAN FD [31], and seamless CAN are summarized in Table 1. Briefly speaking, CAN FD is known as CAN Flexible Data-Rate, which is a slightly enhanced version of the classic CAN protocol with the capability to choose and adapt to slower or faster transmission rates depending on message sizes. It is obvious that while all the three protocols are subject to frame collision, seamless CAN frames are less likely to be affected by errors, thanks to the ability to recover the frame from the other path. Meanwhile, nodes in conventional CAN and CAN FD networks will constantly try to resend the frame whenever a corrupted frame is detected. In addition, seamless CAN supports error correction for encapsulated CAN frames.

In terms of the frame size, the standard CAN frame is 44 bits long without the data field. As it is able to carry a payload of up to 64 bits, the CAN frame size ranges from 44 to

TABLE 1. Comparison among CAN, CAN FD, and seamless CAN (herein abbreviated to sCAN for brevity).

	CAN	CAN FD	sCAN
Collision	Yes	Yes	Yes
Frame size (bits)	[44, 108]	Flexible	{176, 184}
Fault tolerance	No	No	Yes
Error correction	No	No	Yes
Topology	Bus	Bus	Ring

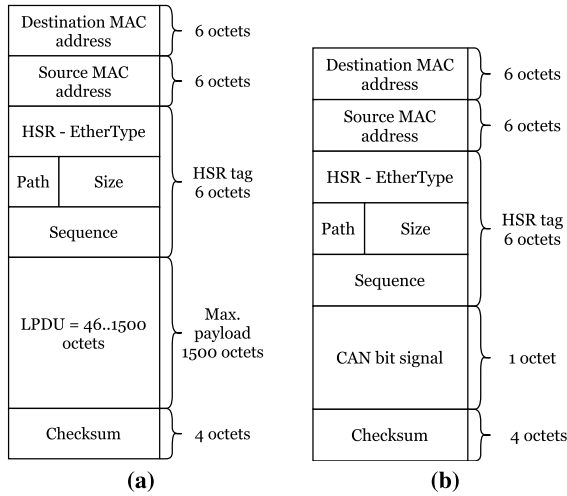


FIGURE 6. Comparison between a (a) standard HSR and (b) seamless CAN frame.

108 bits. In contrast, without the payload field, a standard HSR frame is 176 bits in length. In our proposed work, as shown in Fig. 6, we place the CAN single bit in an octet, which is then the payload of an HSR frame. In short, the seamless CAN frame size will be either 176 bits (i.e., a frame with an empty payload) or 184 bits (i.e., the payload length is an octet, eight bits in length). Although the seamless CAN frame has a larger size, it does not exert any impact on the network performance due to Ethernet’s faster transmission speed compared to the CAN bus.

Finally, it is note that seamless CAN makes use of the well-known HSR protocol which is proven to provide seamless redundancy by its nature of transmitting packets using two separate paths and the common network topology of HSR protocol is the ring network. Therefore, it is also necessary to analyze the time complexity (i.e., the length of time required to successfully deliver a data packet from a source to a destination node) of the proposed work in the case of a ring network. Let’s make an assumption that the ring network consists of n nodes with n being large enough, it is evident that a packet has to travel n nodes to eventually get to the destination node in the worst-case scenario (i.e., the source and destination nodes are adjacent to each other but their direct link is not functional). Therefore, the time complexity of seamless CAN is $\mathcal{O}(n)$.

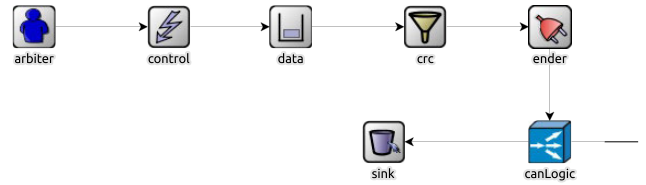


FIGURE 7. CAN node sub-modules.

IV. MODELING

A. SIMULATION MODELING

The object-oriented modular discrete event network (OMNeT++) [32] simulation framework is used to logically simulate

- 1) Standard CAN nodes that are capable of sending and receiving CAN frames;
- 2) An error-inducing mechanism to test the network under multiple scenarios;
- 3) Seamless CAN nodes that are able to encapsulate one CAN signal bit inside a seamless CAN frame and send its replicas via different paths to provide a fault-tolerant connection for CAN nodes.

In order to achieve the configuration for a seamless CAN frame as shown in Fig. 6b, we develop two modules, CAN and seamless CAN. First, the CAN module works on structuring all the CAN frames based on several internal variables determined by the user, with each frame holding a random or predefined payload value. All the whole constructed frames are later sent out bit by bit (i.e., a logical zero for the dominant state and a logical one for the recessive state) as objects within time-based events. The CAN module is also able to receive frames and report whenever an error is detected. In that case, the receiver CAN module will drop the part of the frame received before that error occurrence and issue a resend request to the sender module. Fig. 7 illustrates the OMNeT++ CAN node model and its sub-modules, which are responsible for constructing the corresponding parts of the entire CAN frame and allow any parts of the CAN frame to be altered as needed by the study.

The seamless CAN node (as depicted in Fig. 3) next initially prepares a message object that represents a seamless CAN frame with an empty payload. This module also contains a single sub-module that is capable of encapsulating and decapsulating the received frames. In the case of a CAN bit being received by the CAN side’s interface, it will be put inside the Seamless CAN frame as the payload with the length of an octet. The module later duplicates the complete frame and sends them across the seamless CAN side via two separate interfaces. In contrast, when the module receives a seamless CAN frame from any of the seamless CAN side interfaces, it will decapsulate the frame and forward the extracted payload to the CAN node module via the CAN side interfaces. That is to say, the seamless CAN node module is supposed to be able to read the CAN signal bit, and it does not rely on a store-and-forward scheme for CAN bus activity data

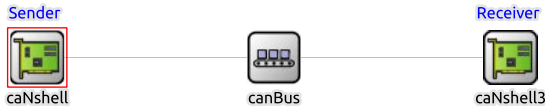


FIGURE 8. CAN bus setup in the simulation scenario.

TABLE 2. Experimental results of the CAN BER values [13].

Environment	Benign	Normal	Aggressive
Bits transmitted	2.02e11	1.98e11	9.79e10
Bit errors	6	609	25239
BER	3.0e-11	3.1e-9	2.6e-7

transfer across the network, as seamless CAN frames will do the job.

Finally, to analyze the network behavior under error-prone environments, we develop another module that is able to induce errors to the frame transmission based on a fixed BER value set at the beginning of the simulation. This error inducer module is located in the link between two adjacent seamless CAN nodes, presuming that there is no error in the data communication between the seamless CAN and CAN node. The goal of this is to demonstrate the superiority of having a fault-tolerance configuration over the conventional CAN bus network.

B. SIMULATION VALIDATION

Before developing simulation scenarios, it is essential to ensure that the CAN simulation model shows certain performance behavior comparable to real-world CAN nodes. Therefore, a testing scenario was set up as illustrated in Fig. 8, where there were two *caNshell* nodes connected directly to a *canBus* node. In actuality, the operation concept is similar to a CAN bus; the sender node will prepare a bitstream resembling a CAN frame, which later will be pushed to the bus bit by bit before finally being delivered to the receiver node. To reproduce the erroneous nature of data transmission processes, the *canBus* node features an error-generating mechanism based on the probability of error occurrence (in terms of BER).

As the simulation starts, it is programmed to halt after the receiver node has received a certain amount of data, and the number of error flags is calculated after the entire simulation. Based on this result, the rate between the number of occurred errors and the total transmitted bits is then compared with the real-world bit error readings obtained from [13] as shown in Table 2. Specifically, Table 2 provides experimental results of the CAN node transmission process under three real-world environments, in which the BER value was calculated based on the total number of bit errors and transmitted bits.

It is obvious that the results from the aggressive environment with the highest BER value would serve as the best candidate for our simulation validation. However, as OMNeT++ does have its sending buffer limitation for a

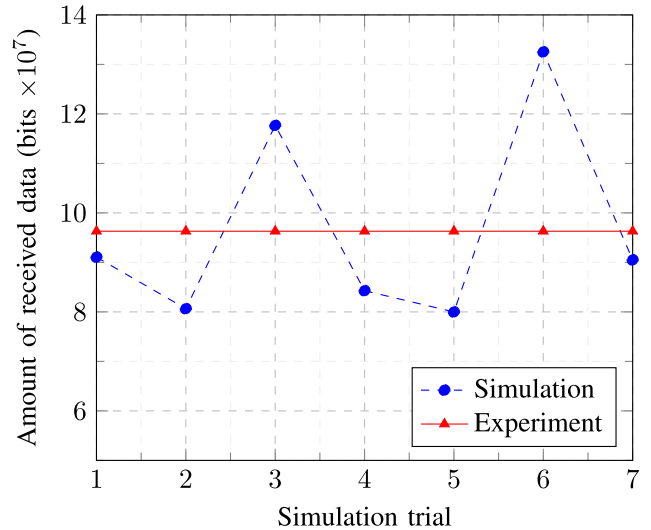


FIGURE 9. Data reception comparison between simulated and experimental results after 25 errors at BER of 2.6e-7.

certain number of message objects, the number of transmitted bits in the third real-world condition had to be reduced before use. In fact, assuming that 9.79e10 transmitted bits lead to 25 239 error bits in the aggressive scenario; 9.79e7 bits result in only 25 error bits. The simulation is then carried out by continually transmitting data from the sender to the receiver with a BER of 2.6e-7 until there are 25 error occurrences; the total number of bits received is recorded. It is evident that the simulated CAN bus is valid only if the number of bits received in the simulation is equal or close to that acquired from practical experiments. For better accuracy, multiple simulation trials were conducted to make relevant comparisons. Fig. 9 shows the simulation validation results, where the x-axis represents the simulation trials and the y-axis measures the amount of data received in comparison with a single experimental result (i.e., 9.79e7). It was apparent that the developed models were reliable enough for further simulation steps since in seven trials, the average number of received bits was approximately 9.67e7 bits, which was acceptable as compared with 9.79e7 bits in real-world applications.

Due to randomness, there is a certain proximity between the simulated and the experimental results. In addition, when an error is detected in the CAN bus during the reception of a frame, the whole frame will be retransmitted regardless of how much information in parts of the frame have already been received before the error occurs. In summary, the total number of received bits depends on

- The size of the message;
- The error position (at which bit the error occurs);
- The value of the BER;
- The number of errors occurring in a single frame.

Therefore, the number of retransmitted bits varies each time an error occurs.

V. PERFORMANCE EVALUATION

After verifying the simulation's validity, we evaluated the performance between the conventional CAN and the proposed seamless CAN scheme in two scenarios:

- Failover capability;
- BER measure.

In each case, relevant models were developed for simulation in OMNeT++ to provide a detailed comparison between seamless CAN and CAN.

A. FAILOVER CAPABILITY

For this metric, a testing scenario was prepared to evaluate the network performance in the case of link failure for both the CAN and seamless CAN protocols. The network arrangement is illustrated in Fig. 8; there is one sender node transmitting data to a receiver node but with a different network link structure according to the characteristic topology of each protocol. At the beginning of the simulation, the environment offers error-free network links between the sender and receiver (i.e., a BER value of zero) with a certain number of frames to be transferred, and each frame is 107 bits long. Later in the simulation, an intentional link failure will occur for a brief period of time, and the network will then return to its normal state. At the end of the simulation, the total number of frames received over time is reported. Detailed simulation scenarios and results are thoroughly discussed as follows.

1) CAN PERFORMANCE

Fig. 10 shows the total number of CAN signal bits as message objects received by a node over the simulation time t measured by "sim. s" (simulation second) unit, including when a link error occurs. In this simulation setup (as shown in Fig. 8), the *caNshell* and *caNshell3* nodes are the sender and receiver, respectively; as there is only one receiver in this case, a data recorder is only required to be fixed inside the receiver node *caNshell3*. For every successfully received bit, this counter increments by one. At $t = 150\,000$ sim. s, the *canBus* node simulates a link error and this lasts until $t = 350\,000$ sim. s. For a healthy condition without faults, there was a linear reception of the bitstream until the link was disrupted for $200\,000$ sim. s from $t = 150\,000$ sim. s. In this faulty situation, there was no message reaching the *caNshell3* node, and its total number of received messages remained the same at around 160 000. In other words, any messages transmitted by the sender node *caNshell* are lost during this time; only after $t = 350\,000$ sim. s, the receiver node began to receive messages normally again. The simulation process finished at $t = 500\,000$ sim. s after approximately 2900 frames were successfully received by the destination node. However, a link failure led to roughly 2000 lost frames, and they would need to be resent in a real-world setting.

2) SEAMLESS CAN PERFORMANCE

Different from CAN, the main topology of seamless CAN is a ring, so even a basic network arrangement of two nodes would include two links between them, as depicted

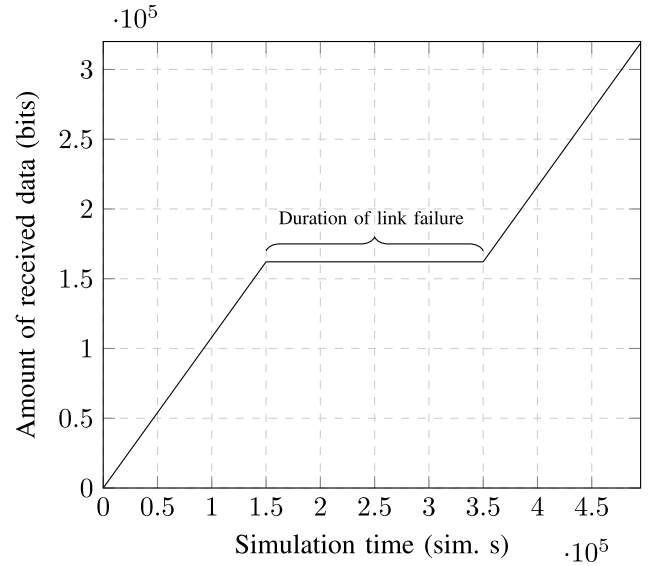


FIGURE 10. CAN frame reception performance in the case of link failure.

in Fig. 11. In fact, two CAN nodes, namely *CAN_1* and *CAN_2*, are connected to two seamless CAN nodes, *SCAN_1* and *SCAN_2*, respectively. These two seamless CAN nodes are connected to each other via two separate links, together forming a ring structure. In this simulation scenario, as there are multiple links in the network, the simulation requires more configuration steps as follows:

- A data recorder is implemented for the *SCAN_2* node in order to monitor all the traffic even when a link failure occurs;
- Each link is maneuvered by an error-generating module (comprised of the *Err_Ind_1* and *Err_Ind_2* nodes) that is capable of causing a link failure at a certain simulation time t . Since there is only one single link failure in this case, the *Err_Ind_1* node will simulate a link disconnection at $t = 150\,000$ sim. s and it lasts until $t = 350\,000$ sim. s;
- The traffic between the *SCAN_1* and *SCAN_2* nodes through *Err_Ind_1* is assumed to have a smaller transmission latency. This means when the two links are both available, the frame copy sent via *Err_Ind_1* node will reach the destination node first and the later arriving frame via *Err_Ind_2* will be discarded.

Fig. 12 shows the total number of seamless CAN frames as message objects received by the *SCAN_2* node via two different ports A and B over the simulation time t . As mentioned previously, duplicated copies arriving at port A will be discarded as their copies have already been received by port B. Therefore, in the first $150\,000$ sim. s, the *SCAN_2* node shows a linear reception of seamless CAN frames completely via port B with no frame coming via port A. However, at $t = 150\,000$ sim. s, a link disruption was caused by node *Err_Ind_1*, and there was an obvious zero frame reception at port B. During this time, frames reaching port A were no longer detected as previously received copies by

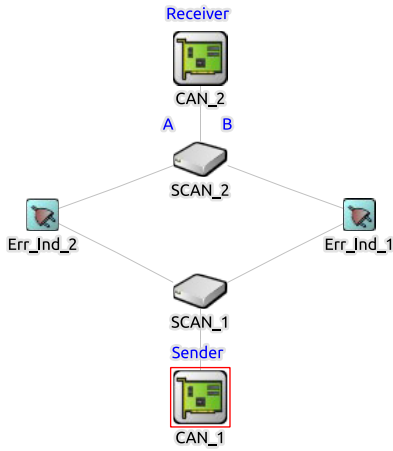


FIGURE 11. Seamless CAN setup in the failover simulation scenario.

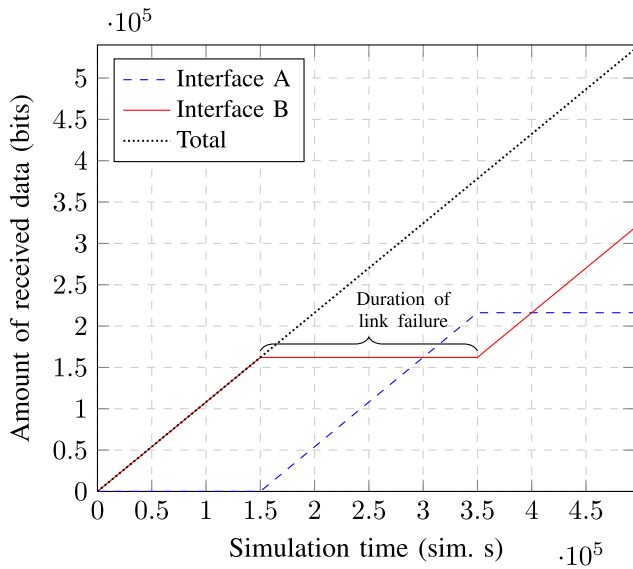


FIGURE 12. Seamless CAN frame reception performance in the case of link failure.

SCAN_2, and thus, they were forwarded to the CAN_2 node in the form of CAN signal bitstreams. The link failure lasted until $t = 350\,000$ sim. s, and the faster link began to deliver frames to the destination node again. In this case, the number of frames received via port B started to increase while the figure for port A remained the same from $t = 350\,000$ sim. s, indicating that SCAN_2 was not accepting frames via port A.

However, in contrast to the CAN simulation results, the number of lost frames in this seamless CAN scenario was zero because all lost frames in one disconnected link were recovered with zero delay by the other functional link. This link fault-tolerance capability makes seamless CAN very suitable for time and data-critical systems.

B. BER MEASURE

We consider a network \mathcal{M} with a set \mathcal{N} of N ($N \in \mathbb{N}$) nodes connected by L undirected links in a set \mathcal{L} . In a typical environment, each link $l_{ij} \in \mathcal{L}$ either directly or indirectly

connecting two nodes i and j ($i \neq j$ and $i, j \in \mathcal{N}$) has its own BER value P_{ij} . It is obvious that in a CAN bus network \mathcal{M}_c (shown in Fig. 8) with $N_c = 2$ and $L_c = 1$, there is only one link l_{12} (the CAN bus) with only one BER value P_{12} . This also holds in the case of more than two nodes being connected to the bus (i.e., $N_c > 2$).

Now let's consider a simple seamless CAN network topology \mathcal{M}_s , as depicted in Fig. 11. Because seamless CAN provides dual paths, for each link l_{ij} , there is another redundant l'_{ij} link that also connects two nodes i and j . Regarding \mathcal{M}_s , there are two links, l_{12} and l'_{12} , between nodes 1 and 2 with their BER values p_{12} and p'_{12} , respectively. As the probability for a bit error occurrence in one link is independent of that in another other link, the overall BER value of all the links between nodes 1 and 2 as P_{12} is given by

$$P_{12} = p_{12} \cdot p'_{12}. \quad (1)$$

In case l_{12} and l'_{12} have the same BER value, $P_{12} = p_{12}^2 = p'^2_{12}$. For example, if $p_{12} = p'_{12} = 10^{-5}$, the overall BER value of link l_{12} is $P_{12} = 10^{-10}$. This shows that a seamless CAN network is less susceptible to errors compared to a standard CAN bus with only one shared link.

In the case of a general seamless CAN ring-type network topology \mathcal{M}_s , a frame is still able to traverse from the source node to some destination node via two paths, but there are more switching nodes in between each path. It is obvious that every pair of adjacent nodes has its own BER value, and the overall BER value between nodes i and j as P_{ij} has to be computed based on all these BER values. Given that $p^l_{ij} = \{p_1, p_2, p_3, \dots, p_{n_l}\}$ is the set of BER values of all consecutive links in the "left" path between nodes i and j , where n_l is the number of links in the left path; the right path also has its BER value set as $p^r_{ij} = \{p_1, p_2, p_3, \dots, p_{n_r}\}$, where n_r denotes the number of links. The accumulated BER values of the left and right path (i.e., P^l_{ij} and P^r_{ij} , respectively) of nodes i and j can be computed by

$$P^l_{ij} = p_1 + \sum_{u=2}^{n_l} \left(p_u \prod_{v=1}^{u-1} (1 - p_v) \right), \quad (2)$$

and

$$P^r_{ij} = p_1 + \sum_{u=2}^{n_r} \left(p_u \prod_{v=1}^{u-1} (1 - p_v) \right). \quad (3)$$

From (1)–(3), the overall BER value P_{ij} between nodes i and j in Seamless CAN network \mathcal{M}_s is calculated using

$$P_{ij} = P^l_{ij} \cdot P^r_{ij}. \quad (4)$$

Now consider two nodes i and j , such that they have the equal number of n links on the left and right paths (i.e., $n^l_{ij} = n^r_{ij} = n$), which leads to $|p^l_{ij}| = |p^r_{ij}| = n$ where $|\bullet|$ denotes the set's length. Without a loss of generality, assume that $p^r_{ij} = p^l_{ij}$ (i.e., the BER value of every link in the two paths is equal to

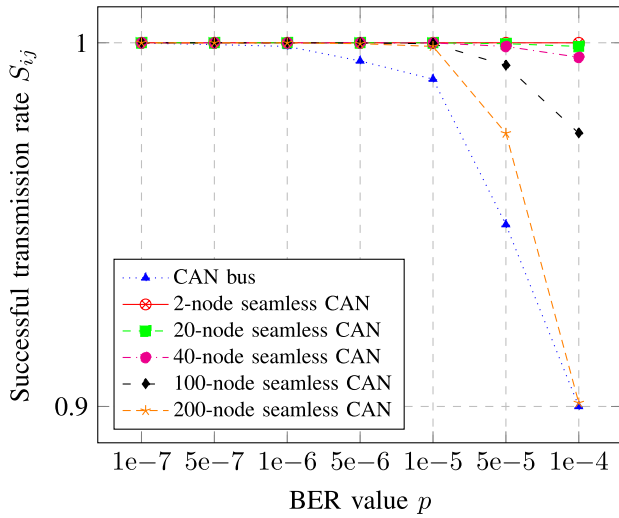


FIGURE 13. Successful transmission rate of CAN and seamless CAN varying in BER and number of network nodes.

a value p). In this case, (2) and (3) can be rewritten as

$$P_{ij}^l = P_{ij}^r = p \sum_{u=0}^{n-1} (1-p)^u, \tag{5}$$

and P_{ij} can still be computed using (4). Using this analysis, Fig. 13 compares the successful transmission rate, denoted by $S_{ij} = 1 - P_{ij}$, between two nodes i and j in the CAN and seamless CAN networks under different environments varying in their total number of network nodes (i.e., $2n = \{2, 20, 40, 100, 200\}$) and BER value (i.e., $p = \{1e-7, 5e-7, 1e-6, 5e-6, 1e-5, 5e-5, 1e-4\}$) for the links in each network. It was apparent that with all link configurations, seamless CAN had a better successful transmission rate than the CAN bus, regardless of the environment's high BER value as $p = \{5e-5, 1e-4\}$. Therefore, seamless CAN is suitable for CAN's current applications.

With the above result, we further analyzed the performance of CAN and seamless CAN using simulation trials as follows. Note that every simulation further in Section V included 90 000 CAN frames or 9 630 000 CAN signal bits (because a frame was 107 bits long) being transmitted from the sender to the receiver, the same set of BER values $\{1e-7, 5e-7, 1e-6, 5e-6, 1e-5, 5e-5, 1e-4\}$, and a record of the number of bit error occurrences at the end.

1) CAN PERFORMANCE

To demonstrate the performance of CAN under different error-prone environments, a network setup was prepared as shown in Fig. 8. An ensemble of six simulations was conducted, in each of which the sender node transmits CAN frames as a sequence of bits to the receiver node with a different BER for the CAN bus, and the results are shown Fig. 14.

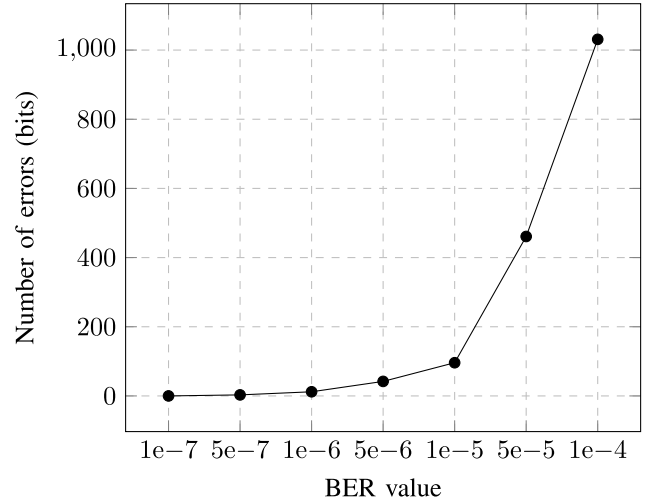


FIGURE 14. CAN network's bit error results under different BER values.

It was found that the lower the BER value, the lower the number of bit errors. In particular, there was a sharp reduction in the number of errors from 1031 to 96 as the BER value decreased by a multiple of 10 from $1e-4$ to $1e-5$. This declining trend continued until the BER value was less than $1e-7$ when there was hardly any error occurrence. Note that because CAN does not have a frame recovery mechanism, the number of error occurrences is equivalent to the number of dropped frames.

2) SEAMLESS CAN PERFORMANCE - DUAL-PATH TWO-NODE

To evaluate the performance of a seamless CAN network under environments with different BER values, a simulation scenario was set up as depicted in Fig. 15. Specifically, there were two connected seamless CAN nodes, namely $SCAN_A$ and $SCAN_B$, which formed a ring, and each of which was also connected to a CAN node. CAN_A and CAN_B were the sender and receiver, respectively. In this setup, as there were two nodes in the network and they had two paths between themselves, it is called a dual-path two-node network. In addition, each of the two paths had an error-inducer node (called Err_Ind_1 and Err_Ind_2), which were responsible for providing random error generation based on the desired BER value set at the beginning of the simulation. Both error inducer nodes have the same BER value. Meanwhile, the seamless CAN node $SCAN_B$ indicated the different data reception values from both interfaces, depending on the number of error occurrences in each path.

Fig. 16 compares the number of bit errors between CAN (obtained from Section V-B1) and seamless CAN. In general, Seamless CAN showed a significant improvement compared to CAN bus, and seamless CAN offered completely error-free transmission with no detected error for any BER values. Regarding CAN bus, in an extremely aggressive environment with a BER equal to $1e-1$, there were more than 1000 errors. As the BER value became $5e-7$ and smaller, CAN bus started

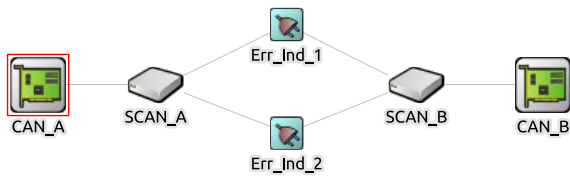


FIGURE 15. Dual-path two-node seamless CAN network structure for simulation scenarios.

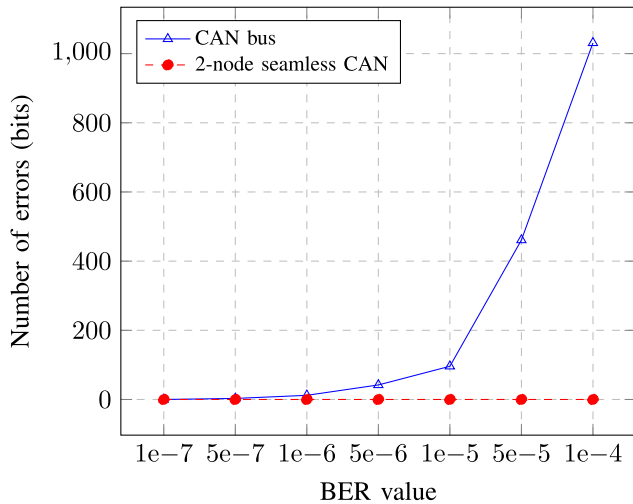


FIGURE 16. Dual-path two-node seamless CAN bit-error results compared with CAN bus under different BER values.

to have a relatively low number of errors. It should also be noted that every time an error occurred in a seamless CAN path, it was automatically recovered from the other path. However, when two copies of the same frame were lost due to error occurrences in both paths, this was called a “critical error”. Because conventional CAN had only one shared bus for transmission, any error occurrence was a critical error.

3) SEAMLESS CAN PERFORMANCE - DUAL-PATH MULTIPLE-NODE

We considered a seamless CAN network with more seamless CAN nodes in a ring topology, indicating that there were actually more CAN nodes being connected to the CAN bus. For example, Fig. 17 depicts an in-vehicle network with 10 CAN nodes, each of which is connected to a seamless CAN node to collectively form a ring topology. For readability, the seamless CAN, as well as the CAN nodes, are not shown. Compared to the dual-path two-node network, there were also two paths between every pair of nodes but more than one node in each path, so this setup was referred to as dual-path multiple-node.

Fig. 18 compares the number of bit errors of a conventional CAN bus and seamless CAN with five different configurations (i.e., 2, 20, 40, 100, and 200 nodes). When the BER was equal to 1e-4, the CAN bus and 200-node seamless CAN had approximately 1000 errors (1031 and 985, respectively). However, when the BER was 1e-5, there were 96 errors reported for the CAN bus while the figure for the 200-node

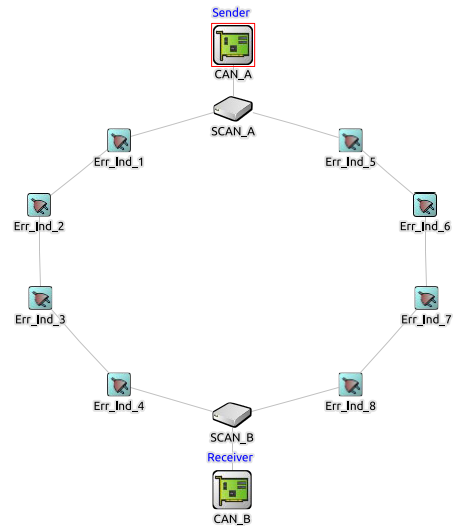


FIGURE 17. Example of a dual-path ten-node seamless CAN ring network topology.

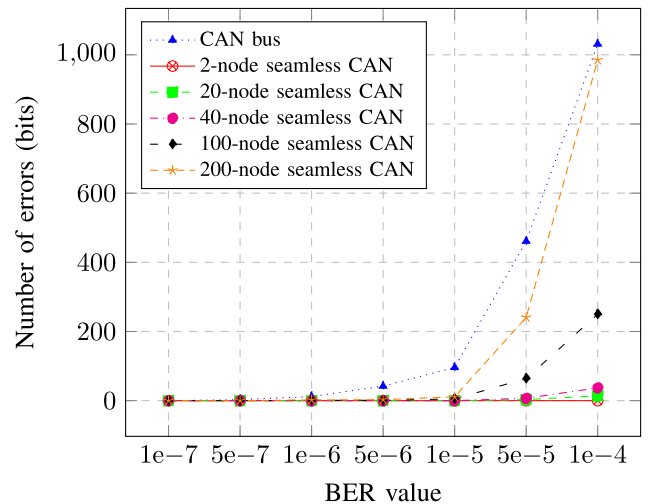


FIGURE 18. Dual-path multiple-node seamless CAN network's bit-error results in comparison with the CAN bus under different BER values and numbers of nodes.

seamless CAN is was 10 times lower (only 11 errors). Moreover, the other seamless CAN network sizes (ranging from two to 100 nodes) all showed fewer errors than the CAN bus for any BER. It was obvious seamless CAN outperformed the CAN bus for all BER regardless of the network size.

In summary, it is worth mentioning that in a modern vehicle there are up to 150 integrated ECUs [33]. With our results of up to 200 seamless CAN nodes in scenarios with acceptable BER values, seamless CAN guarantee consistent performance for current IVN structures with various numbers of ECUs. In addition, because seamless CAN supports a multi-ring topology, it is possible to divide the whole network system into smaller connected rings to ensure seamless CAN's performance.

VI. CONCLUSION AND FUTURE WORK

In this paper, we proposed a novel fault-tolerant algorithm, seamless CAN, for the conventional CAN bus protocol. Our proposed scheme is based on the operating concept of HSR to provide redundant solutions for time-critical in-vehicle network systems. The simulation results showed that seamless CAN had no loss of frames in the case of link failures and significantly better BER performance, indicated by a lower number of bit error occurrences. In addition, the seamless CAN implementation requires no modification to current CAN frames, which makes it a feasible interim solution during the transition period to next-generation IVNs.

However, HSR is known for generating excessively redundant traffic in a network in order to achieve seamless redundancy. Therefore, our future study will focus on applying many traffic reduction techniques [34] to improve seamless CAN's traffic performance. In addition, we believe that implementing seamless CAN with our recently developed HSR switching node *SwitchBox* will enable seamless CAN to be applied to any network topology with reduced unnecessary traffic. Finally, the current seamless CAN frame's payload portion is one octet long, and a seamless CAN node is connected to only a CAN node. Future works may also involve using this payload field to encapsulate multiple signal bits from various CAN nodes to lessen the implementation cost.

ACKNOWLEDGMENT

An earlier version of this paper was presented at the 2021 International Conference on Information and Communication Technology Convergence (ICTC) [DOI: 10.1109/ICTC52510.2021.9621095].

REFERENCES

- J. M. Rhee, I. R. A. Altaha, D. N. M. Hoang, D. H. Kim, J. S. Yang, and S. Y. Park, "Seamless CAN: CAN bus plus high-availability seamless redundancy," in *Proc. Int. Conf. Inf. Commun. Technol. Converg. (ICTC)*, Oct. 2021, pp. 1284–1286.
- G. Pan and M.-S. Alouini, "Flying car transportation system: Advances, techniques, and challenges," *IEEE Access*, vol. 9, pp. 24586–24603, 2021.
- B. Fleming, "Advances in automotive electronics," *IEEE Veh. Technol. Mag.*, vol. 10, no. 4, pp. 4–96, Dec. 2015.
- T. Warren, (Jun. 2021). *Elon Musk Demos Cyberpunk 2077 on New Tesla Infotainment System With 'PS5-Level Performance'*. The Verge. [Online]. Available: <https://bit.ly/37LZ1g5>
- W. Schwarting, J. Alonso-Mora, and D. Rus, "Planning and decision-making for autonomous vehicles," *Annu. Rev. Control, Robot., Auto. Syst.*, vol. 1, pp. 187–210, May 2018.
- L. D'Orazio, F. Visintainer, and M. Darin, "Sensor networks on the car: State of the art and future challenges," in *Proc. Design, Autom. Test Eur.*, Mar. 2011, pp. 1–6.
- T. Ziermann, S. Wildermann, and J. Teich, "CAN+: A new backward-compatible controller area network (CAN) protocol with up to 16× higher data rates," in *Proc. Design, Automat. Test Eur. Conf. Exhib.*, Apr. 2009, pp. 1088–1093.
- S. Kang, S. Han, S. Cho, D. Jang, H. Choi, and J.-W. Choi, "High speed CAN transmission scheme supporting data rate of over 100 Mb/s," *IEEE Commun. Mag.*, vol. 54, no. 6, pp. 128–135, Jun. 2016.
- Y. Lee, "Analysis of unintended acceleration through physical interference of accelerator," *Forensic Sci. Int., Rep.*, vol. 2, Dec. 2020, Art. no. 100079.
- A. S. Sawant and M. D. Joshi, "CAN, FlexRay, MOST versus Ethernet for vehicular networks," *Int. J. Innov. Advancement Comput. Sci.*, vol. 7, pp. 336–339, Apr. 2018.
- Institute of Electrical and Electronics Engineers. *P802.1CB—Frame Replication and Elimination for Reliability*. [Online]. Available: <https://1.ieee802.org/tsn/802-1cb/>
- Industrial Communication Networks—High Availability Automation Networks—Part 3: Parallel Redundancy Protocol (PRP) and High-Availability Seamless Redundancy (HSR)*, Standard IEC 62439-3, International Electrotechnical Commission, Geneva, Switzerland, 2016.
- J. Ferreira, "An experiment to assess bit error rate in CAN," in *Proc. 3rd Int. Workshop Real-Time Netw. (RTN)*, 2004, pp. 15–18.
- J. Rufino, P. Verissimo, G. Arroz, C. Almeida, and L. Rodrigues, "Fault-tolerant broadcasts in CAN," in *Proc. 28th Annu. Int. Symp. Fault-Tolerant Comput.*, Munich, Germany, Jun. 1998, pp. 150–159.
- M. Short and M. J. Pont, "Fault-tolerant time-triggered communication using CAN," *IEEE Trans. Ind. Informat.*, vol. 3, no. 2, pp. 131–142, May 2007.
- J. Ferreira, P. Pedreiras, L. Almeida, and J. Fonseca, "Achieving fault tolerance in FTT-CAN," in *Proc. 4th IEEE Int. Workshop Factory Commun. Syst.*, Aug. 2002, pp. 125–132.
- H. Xiang-Dong, Y. Hui-Mei, and Z. Xiao-Xu, "Design of dual redundancy CAN-bus controller based on FPGA," in *Proc. IEEE 8th Conf. Ind. Electron. Appl. (ICIEA)*, Jun. 2013, pp. 843–847.
- Z. Wang, X. Guo, and C. Yu, "Research of fault-tolerant redundancy and fault diagnosis technology based on can," in *Proc. 2nd Int. Conf. Adv. Comput. Control*, 2010, pp. 287–291.
- J. Rufino, P. Verissimo, and G. Arroz, "A Columbus' egg idea for CAN media redundancy," in *Proc. 29th Annu. Int. Symp. n Fault-Tolerant Comput.*, Madison, WI, USA, Jun. 1999, pp. 286–293.
- N. Kumar, R. Sharma, and B. Ramya, "Design and development of fault tolerant CAN-LIN gateway for in-vehicle communication," *SASTech Tech. J. RUAS*, vol. 9, pp. 17–22, 2020.
- N. Tien and J. Rhee, "Developing a new HSR switching node (SwitchBox) for improving traffic performance in HSR networks," *Energies*, vol. 9, no. 1, p. 36, Jan. 2016.
- D. H. Kim, N. Y. Lee, J. M. Rhee, and S. Y. Park, "Performance analysis for HSR based in-vehicular network (IVN) under end-to-end latency (EEL) constraint with store-and-forward mode," in *Proc. Int. Conf. Inf. Commun. Technol. Converg. (ICTC)*, Jeju, South Korea, Oct. 2020, pp. 1001–1003.
- S. Ro, K.-J. Lee, and H.-S. Ahn, "Design of a reliable in-vehicle network using ZigBee communication," in *Multimedia and Ubiquitous Engineering*, vol. 240. Dordrecht, The Netherlands: Springer, 2013, pp. 777–783.
- CAN specification. Version 2.0*, vol. 300240, Robert Bosch GmbH, Germany, 1991, p. 72.
- K. H. Johansson, M. Törnqvist, and L. Nielsen, "Vehicle applications of controller area network," in *Handbook Networked and Embedded Control Systems*. Boston, MA, USA: Birkhäuser, 2005, pp. 741–765.
- K. C. Emani, K. Kam, M. Zawodniok, Y. R. Zheng, and J. Sarangapani, "Improvement of CAN BUS performance by using error-correction codes," in *Proc. IEEE Region 5 Tech. Conf.*, Apr. 2007, pp. 205–210.
- H. Kirmann, "HSR—High availability seamless redundancy, fault-tolerance in Ethernet networks IEC 62439," Presentation Slides, ABB Switzerland Ltd, Corporate Res., Baden, Switzerland, Tech. Rep., 2010.
- X. Jiang, "High availability seamless ring protocol implementation in FPGA," M.S. thesis, Comput. Eng., Swiss Federal Inst. Technol., Zurich, Switzerland, 2009.
- I. R. Abdulsalam and J. M. Rhee, "Improvement of high-availability seamless redundancy (HSR) unicast traffic performance using port locking," in *Proc. Fourth World Congr. Softw. Eng.*, Dec. 2013, pp. 246–250.
- S. Nsaif and J.-M. Rhee, "DVP: A novel high-availability seamless redundancy (HSR) protocol traffic-reduction algorithm for a substation automation system network," *Energies*, vol. 7, no. 3, pp. 1792–1810, Mar. 2014. [Online]. Available: <https://www.mdpi.com/1996-1073/7/3/1792>
- R. De Andrade, K. N. Hodel, J. F. Justo, A. M. Lagana, M. M. Santos, and Z. Gu, "Analytical and experimental performance evaluations of CAN-FD bus," *IEEE Access*, vol. 6, p. 21287–21295, 2018.
- OMNeT++ Discrete Event Simulator*. Accessed: Jun. 2, 2021. [Online]. Available: <https://omnetpp.org/>
- A. Winning, (May 2019). *Number of Automotive ECUs Continues to Rise*. eeNews Automotive. [Online]. Available: <https://bit.ly/37GUpba>
- N. Tien, S. Nsaif, and J. Rhee, "A comparison of techniques for reducing unicast traffic in HSR networks," *Energies*, vol. 8, no. 10, pp. 12029–12060, Oct. 2015. [Online]. Available: <http://www.mdpi.com/1996-1073/8/10/12029>



JONG MYUNG RHEE (Member, IEEE) received the Ph.D. degree from North Carolina State University, Raleigh, NC, USA, in 1987.

After 20 years at the Agency for Defense Development, South Korea, where he made noteworthy contributions to C4I and military satellite communications, he joined DACOM and Hanaro Telecom, in 1997 and 1999, respectively. At Hanaro Telecom, which was the second-largest local carrier, South Korea, he has served as the Chief Technology Officer (CTO) with a Senior Executive Vice-President position. His main duty at Hanaro Telecom was a combination of management and new technology development for high-speed internet, VoIP, and IPTV. In 2006, he joined Myongji University, where he is currently a Special Mission Professor with the Department of Information and Communications Engineering. His recent research interests include military communications and smart grids, including ad-hoc and fault-tolerant networks.



DONG HWAN KIM received the B.S. and M.S. degrees in electronic engineering from Myongji University, Yongin, Republic of Korea, in 2020 and 2022, respectively. Since 2022, he has been working as an Engineer at LIG Nex1 Corporation, Yongin. His main research interests include low-power digital system design, high-speed video signal processing, and fault-tolerant network design.



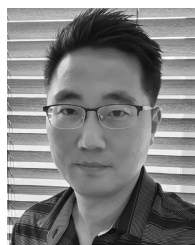
IBRAHEEM R. A. ALTAHA received the B.Sc. and M.Sc. degrees from the Department of Laser and Optoelectronics Engineering, Nahrain University, Baghdad, Iraq, in 2010, and the Ph.D. degree from the Department of Information and Communication, Myongji University, Republic of Korea, in 2017. In 2007, he worked as a Network Engineer and a Network Designer for a private institute for five years. After obtaining his Ph.D. degree, he worked as a Researcher and a Developer for MPEES Company for two years, and worked for six months in the automotive MCU development field at SDCmicro.



JIN SEOK YANG received the B.S. and M.S. degrees in electronic engineering from Myongji University, Yongin, Republic of Korea, in 2020 and 2022, respectively. Since 2022, he has been working as an Engineer at Hanwha Systems Corporation, Yongin. His main research interests include fault-tolerant network design, digital system design, and design methodology in digital front-end design.



DUC N. M. HOANG (Graduate Student Member, IEEE) received the B.E. degree in computer engineering from the Ho Chi Minh City University of Technology (HCMUT), Vietnam, in 2020. He is currently pursuing the M.S. degree with the Department of Electronics Engineering, Myongji University, South Korea. His research interests include ad-hoc networking, wireless networks, fault-tolerant networks, and the Internet of Things.



SANG YOON PARK (Member, IEEE) received the B.S. degree in electrical engineering and the M.S. and Ph.D. degrees in electrical engineering and computer science from Seoul National University, Seoul, Republic of Korea, in 2000, 2002, and 2006, respectively. In 2007, he joined the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, as a Research Fellow. From 2008 to 2014, he was a Research Scientist with the Institute for Infocomm Research, Singapore. Since 2014, he has been with the Department of Electronic Engineering, Myongji University, Yongin, Republic of Korea, where he is currently an Associate Professor. His research interest includes design of dedicated and reconfigurable architectures for low-power and high-performance digital communication systems.

...